Technical Report Documentation Page

1. Report No. FHWA/TX-03/4023-2	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle REQUIREMENTS SPECIFICATION FOR DMS MESSAGE OPTIMIZATION SOFTWARE TOOL (MOST)		5. Report Date September 2002
		6. Performing Organization Code
7. Author(s) Paul Davis, Sangita Sunkari, Conrad Dudek, and Kevin Balke		8. Performing Organization Report No. Report 4023-2
9. Performing Organization Name and Address Texas Transportation Institute The Texas A&M University System College Station, Texas 77843-3135		10. Work Unit No. (TRAIS)
		11. Contract or Grant No. Project No. 0-4023
12. Sponsoring Agency Name and Address Texas Department of Transportation Research and Technology Implementation Office P.O. Box 5080 Austin, Texas 78763-5080		13. Type of Report and Period Covered Research: January 2002 – August 2002
		14. Sponsoring Agency Code

15. Supplementary Notes
Research performed in cooperation with the Texas Department of Transportation and the U.S. Department of Transportation, Federal Highway Administration.
Research Project Title: Automated Dynamic Message Sign (DMS) Message Design and Display

16. Abstract
Composing a message for a dynamic message sign requires managers and supervisors at TxDOT Traffic Management Centers to consider numerous factors. For example, they must consider the content and length of the message as well as memory load for motorists. Following documented guidelines about formatting and phrasing of messages, the requirements for a software system called the DMS Message Optimization Software Tool, or MOST, are discussed. The system is designed to accept input data through a graphical user interface, to allow selection of terms, and to produce a message suitable for display in a DMS. The application automatically applies principles of good message design and allows users to customize their messages. The design of the system follows work done previously in TxDOT Project 0-4023.

17. Key Words Dynamic Message Signs, Advanced Traveler Information Systems, Intelligent Transportation Systems	18. Distribution Statement No restrictions. This document is available to the public through NTIS: National Technical Information Service 5285 Port Royal Road Springfield, Virginia 22161

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 110	22. Price

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

REQUIREMENTS SPECIFICATION FOR DMS MESSAGE OPTIMIZATION SOFTWARE TOOL (MOST)

by

Paul Davis
Systems Analyst II
TransLink® Research Center
Texas Transportation Institute

Sangita Sunkari
Programmer/Analyst I
TransLink® Research Center
Texas Transportation Institute

Conrad Dudek, Ph.D., P.E.
Research Engineer
Texas Transportation Institute

and

Kevin Balke, Ph.D., P.E.
Center Director
TransLink® Research Center
Texas Transportation Institute

Report 4023-2
Project Number 0-4023
Research Project Title: Automated Dynamic Message Sign (DMS) Message Design and Display

Sponsored by the
Texas Department of Transportation
In Cooperation with the
U.S. Department of Transportation
Federal Highway Administration

September 2002

TEXAS TRANSPORTATION INSTITUTE
The Texas A&M University System
College Station, Texas 77843-3135

DISCLAIMER

ACKNOWLEDGMENTS

This project was conducted in cooperation with TxDOT and FHWA. The authors extend thanks to Melisa D. Finley, Texas Transportation Institute (TTI), for developing decision flow charts and models based upon the work of Dr. Conrad Dudek, Research Engineer at TTI and Professor of Civil Engineering at Department of Civil Engineering, Texas A&M University.

The authors thank the following TxDOT employees who provided guidance and expertise throughout the course of this study: Richard Reeves, project director, and Al Kosik, program coordinator. The authors also wish to thank project advisory committee members Rick Cortez, Wallace Ewell, Patrick Irwin, Bubba Needham, Terry Sams, and Sally Wegmann.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Page

CHAPTER 1: INTRODUCTION

1.1. BACKGROUND

A dynamic message sign (DMS) is a device used to convey information to motorists about events that might affect their travel experience and safety. Dudek (*1, 2*) developed processes for TxDOT that will formulate messages. A Basic DMS Message is the sum total of all information that motorists need in order to make a fully informed, rational driving decision (e.g., whether to take an alternative route). The message elements that make up a Basic DMS Message for events are as follows:

· incident description,
· incident location,
· lanes affected,
· effect on travel,
· audience for action,
· action suggested, and
· one good reason for taking action.

Content, length, load, and format under traffic and environmental conditions affect how motorists comprehend messages. Dudek and Huchingson (*2*) established the *unit of information* (UOI) as the criterion for the length of a part of a DMS message. One UOI would represent a message element. Dudek (*1*) recommended that a DMS message not exceed four UOIs when operating speeds on a highway are 35 mph or higher, and he proposed that the message not exceed five units when speeds are less than that at lower speeds. The ability of a motorist to read and comprehend a message is affected by the number of units of information.

In most cases, the Basic Message constructed will exceed the maximum number of UOIs that should be displayed. Therefore, the designer must reduce the length and content to allow motorists to read, understand, and react to the message.

Dudek (*1*) developed and documented a step-by-step DMS message design process, and he also generated decision flowcharts about how to make well-designed messages. Finley *et al.* (*3*) progressed with the work on the flowcharts. The modified decision flowcharts form the basis for the logic of the software system described in this document.

Once a Basic Message is generated, the following five guidelines serve to organize the UOIs:

· no more than two frames should be used,
· each frame must be understood by itself,
· compatible units of information should be displayed on the same frame,
· a message line should not contain portions of two different UOIs, and
· no more than three UOIs should be displayed on a single frame.

Concepts and principles developed through years of human factors and traffic engineering research form the foundation for effective DMS message design. As knowledge about good design principles grows, it becomes challenging for managers, supervisors, and message designers to stay abreast of issues. Consequently, efforts to simplify the DMS message design process for users through technology offer significant potential. This document contains the requirements of a software system called the DMS Message Optimization Software Tool, or MOST. Its purpose is to expedite the design of effective DMS messages for authorized TxDOT personnel.

1.2. PURPOSE

This section describes the purpose of the software system.

1.2.1. Definition of Purpose

The purpose of the MOST system is to guide TxDOT personnel in the construction of well-designed messages for DMSs by using a selection of UOIs, or terms, and encoded rules or suggestions. Objectives and requirements of the MOST system are described in this document.

1.2.2. Audience

Chapter 1 is intended for managers and administrators who evaluate software systems that expedite, simplify, and standardize as much as possible the design of messages for DMSs. Chapter 2 is for project leaders who wish to understand the objectives of the system as well as integration of parts. Chapter 3 is for developers, to guide them in their implementation of the software.

1.3. SCOPE

This section contains names, limitations, and applications of the software system.

1.3.1. Software Product by Name

The name of the software system will be DMS Message Optimization Software Tool (MOST).

1.3.2. What the Software Will and Will Not Do

When using the application, a user will enter data and select phrases, called “terms.” Each term represents a UOI. The software puts these terms together, and the resulting concatenation of terms is the Basic DMS Message. The tool then produces a variation of the Basic Message called the Suggested Message.

The Basic Message gives motorists information about incidents, roadwork, and other events. The application formats this message for display on a DMS. If connected to a TxDOT database, the application can give the user the names of streets, highways, and other traffic-related information. These names appear in the UOIs. The application allows the user to retrieve and edit data and messages from previous sessions and automatically substitutes abbreviations for certain words to shorten the length of the message. In this document, a session is an event that starts with the creation of a message using MOST that culminates with the long-term storage of that message.

The application automatically applies rules to the Basic DMS Message for making well-designed messages. The resulting message is called a Suggested Message. To complete a session, a user saves the Suggested Message. To allow flexibility of deployment, the application requires third-party software to transfer the message to a DMS.

1.3.3. Application of the Software

This section contains an explanation of the benefits as well as definition of the objectives and goals for the MOST system.

1.3.3.1. Benefits

When messages are built manually, the message designers must estimate such factors as the length of the message, the correct wording, and the comprehensibility of the message given traffic speeds and weather conditions. They must also decide if the message will fit within the frames of a DMS. For long messages, they must decide how to distribute the message across frames.

The system described herein provides messages that fit within the physical dimensions of a three-line DMS with up to two message frames. Users of the software enter data about events, such as incidents or roadwork, and then the application automatically composes the Suggested Message.

The program guides users in their progress as they work. When building a message, the user is directed along a task path. Using encoded rules, the program modifies, removes, or re-orders phrases in the Basic DMS Message. The program determines how much information the message can contain based on prevailing traffic and environmental conditions. The user can compare the Basic Message to the Suggested Message. If the work session is saved, the application saves the message as well as all data entered. By searching for previously saved messages, the user can reload the session data and the Suggested Message.

1.3.3.2. Objectives

The objective of the MOST system is to guide a user to compose an understandable and informative message from a collection of phrases for a DMS having a maximum of three lines.

1.3.3.3. Goals

The goals of the system are to:

- provide an easy-to-use graphical user interface, or GUI;
- collect data about an event;
- provide a set of phrases, called terms, with which a message appropriate to traffic conditions can be built;
- automate certain design tasks as related to choosing, eliminating, or changing terms;
- guide the user in composition of a message; and
- provide a method to save, store, and recall messages and data.

1.4. DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

1.4.1. Definitions

Table 1 contains definitions for words used within this document.

Table 1. Definitions.

Terminology	Definition
Application	The product described in this document.
Basic DMS Message	A message that the user constructs. It is the sum total of all the information a motorist needs to make a fully informed and rational decision in response to events.
Category of term	Terms are categorized into the following seven groups: 1. Incident description 2. Incident location 3. Lanes affected 4. Effect on travel 5. Audience for action 6. Action suggested 7. One good reason
Customer	The Texas Department of Transportation (TxDOT).
Encoded knowledge	A body of knowledge, based on human experience and expertise within a field, represented in computer code in such a way that the knowledge can be applied to solve certain kinds of problems.
Placeholder variables	The application can substitute a value for a variable within a term. These variables, called "placeholders," are identified as italicized terms within square brackets. For example, "NEAR *[street]*" is a term or UOI containing the placeholder variable *[street]*. If the user selects "Magnolia Street" from the list of streets, "Magnolia Street" replaces the placeholder variable, i.e. *[street]*, in the message.
"the software"	The MOST software system.
Suggested Message	A message that the application builds based on the message originally designed by the user. The application applies encoded knowledge to construct the Suggested Message.
Term	A single unit of information, or UOI. The phrase "AT *[highway name]*" is an example. TxDOT will supply these terms.
Unit of Information	An indivisible unit consisting of words or phrases constituting one block of information that the motorist can understand. It is an answer to a question that a motorist would have about an incident or roadwork event. One term represents a UOI.
User	A person who uses the application to generate messages.

1.4.2. Acronyms

Commonly used acronyms used in this report are given in Table 2.

Table 2. Acronyms.

Acronym	Terminology	Definition
API	Application Program Interface	A set of protocols, tools, and interfaces designed to allow software and system professionals to build software. An API gives developers a common foundation and repository of functionalities that allow flexibility in how a software system is implemented.
ASCII	American Standard Code for Information Interchange	A code system used to represent English characters as integers from 0 to 127.
DMS	Dynamic Message Sign	Either permanent or temporarily placed roadway equipment used to display information to vehicle operators by means of various technologies, such as flip panels, light-emitting diodes, or multiple incandescent lights. The messages impart information about events that affect the flow of traffic.
LED	Light-emitting diode	An electronic device that usually displays a red light and is capable of operating for an extended period of time.
MOST	DMS Message Optimization Software Tool	The software system described in this document.
ODBC	Open Database Connectivity	An API for database access, ODBC allows programs to use SQL commands to send and receive data from relational databases.
RDBMS	Relational Database Management System	A software system that allows the administration and use of relational databases. Examples include products from vendors like Oracle® and Microsoft®.
SQL	Structured Query Language	A standardized query language used to interact with databases. Typical actions a SQL statement might invoke are insertions, updates, or deletions of data.
SRS	Software Requirements Specification	A document that specifies the requirements for software. The "IEEE Recommended Practice for Software Requirements Specifications," IEEE Std 830-1998 describes the structure and purpose of SRS documents.
TMC	Traffic Management Center	A centralized location where data and images from vehicular traffic are processed to assist management of traffic.
TxDOT	Texas Department of Transportation	The Texas Department of Transportation.
XML	Extensible Mark-up Language	A universal format developed through the World Wide Web Consortium (W3C) to structure data and documents for exchange over the Web. It is a modular, license-free, and platform-independent set of rules to format, delineate, and identify data within text files in such as way as to allow exchange and presentation to data among heterogeneous computer programs, such as Web browsers and databases.
UOI	Unit of Information	See Unit of Information in Table 1, Definitions.

1.5. OVERVIEW

The contents of the rest of the document are summarized.

1.5.1. Contents of the Rest of the Document

The rest of this document contains an overview of system architectures, databases, and interfaces. Chapter 2 contains a description of each part of the system as well as functions of those parts. It also presents a description of functionalities in the application with which the user interacts. This chapter describes the system at a level of detail suitable for project leaders during planning phases. Chapter 3 contains specifics about parts of the system that developers could use during the implementation phase.

1.5.2. Organization

The organization of this SRS follows recommendations of the IEEE 830-1998, "IEEE Recommended Practice for Software Requirements Specifications."(*4*) This document is organized into chapters and sections. From sections to subsections, descriptions and requirements cascade down from broad, high-level explanations to more detailed descriptions.

CHAPTER 2: OVERALL SOFTWARE DESCRIPTION

2.1. INTRODUCTION

This chapter describes the architecture and interfaces for user, hardware, software, and communications systems. It gives requirements for the operation of the system as well as descriptions for exception handling. Site adaptations, product functions, and user characteristics follow. The chapter includes descriptions of constraints on the system and assumptions in product design. It concludes with suggestions for apportioning requirements during development.

2.2. PRODUCT PERSPECTIVE

An explanation is provided about the interactions of the software with other products as might be associated with a TMC.

2.2.1. System Architecture

The MOST software system is designed to function as a stand-alone application, but it can also exchange data within larger systems. The architecture of the system is shown in Figure 1. A description of each part follows.

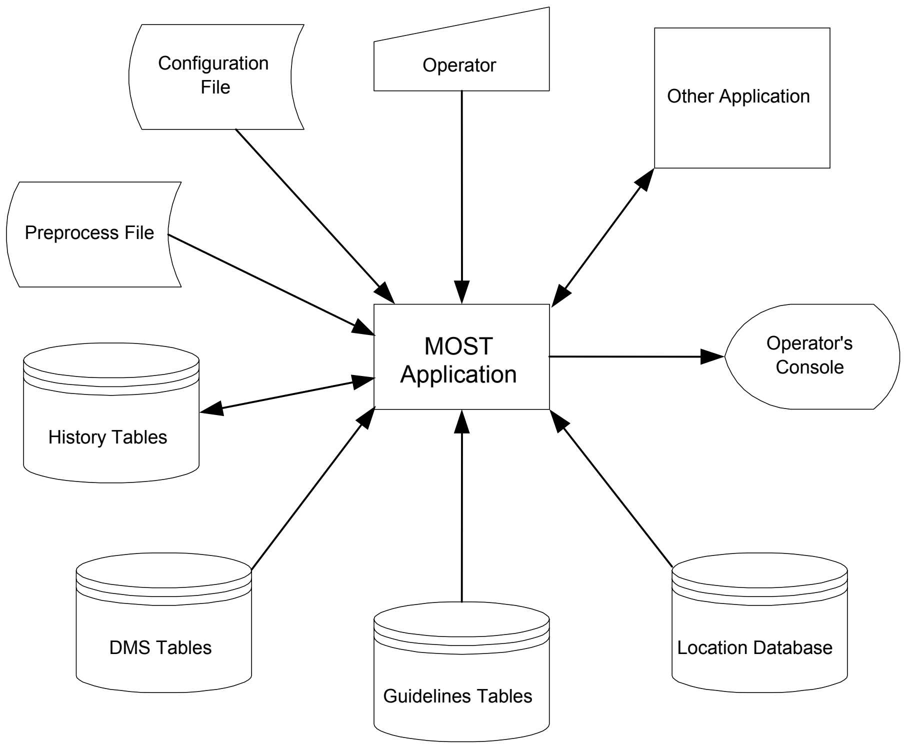


Figure 1. High-Level System Architecture of MOST.

2.2.1.1. MOST Application

This application accepts, processes, and displays data in interaction with a user. The output of the application is a message suitable for display on a DMS. It is designed to be either a stand-alone application or one that can leverage external networked resources.

2.2.1.2. Preprocess File

The application guides the user along a task path, the goal of which is construction of a Suggested Message. Each step of a task path uses a category of terms that are uniquely associated with that step. The preprocess file contains all possible paths as well as categories of terms associated with each step. The preprocess file, as shown in Appendix A, is used in conjunction with DMS tables to identify terms, or UOIs, by category. The application uses the term identified to populate lists. The user picks terms from the lists to construct the Basic Message. The preprocess file is written as an XML file.

2.2.1.3. Configuration File

The configuration file gives the MOST application a flexible way to connect to and query from ODBC-compliant relational databases, such as those available from vendors like Oracle, Sybase®, or Microsoft. The software can be configured to obtain information about roads, highways, cross streets, and so forth from such databases, called "Location" databases in this report. The configuration file has three sections—a connection string, a wild-card character, and SQL commands. The application uses the connection string to establish communication with a relational database system. Because databases differ in the characters used to perform broad, or "wild-card," searches, the file contains the ASCII character specific to the vendor's product for these purposes. Finally, the file contains a list of SQL commands.

An example is given. Assume an incident occurred on IH-35 in Austin. A person at the TxDOT-Austin TMC starts the MOST application. The user enters data about the incident into Screen 1 (see Section 2.2.3.1) and works through to Screen 3 (see Section 2.2.3.3). Before Screen 3 opens, the application checks to see if it should show a list of highways in Austin. To do this, it refers to the Highway SQL Command column in the configuration file. This is shown in Table 3, Section 3.4.2. For this example, assume the following command is taken from the SQL command column.

```
SELECT roadway_number, roadway_description as Highway
FROM roadway
```

For this row in the configuration file, the Highway label identifies this command. This label is also the name of a variable used within the code of the application to represent the names of highways. (The actual name used in the operational version of the program might differ from that used in this example.)

The application submits the query to the Location database. The database returns a list of highways, and the application populates the highway list in the GUI with the results. One of the highways shown in the list is "IH 0035." If the user selects it, the application looks up the entry from the configuration file for cross streets and executes a SQL command like the following:

```
SELECT street.street_number, street.street_description as
CrossStreet
FROM street, cross_street, roadway
WHERE street.street_number = cross_street.street_number
AND cross_street.roadway_number = roadway.roadway_number
AND roadway.roadway_description LIKE '%" & Highway & "%'
```

The application populates the cross street list in the GUI with the results returned from the database. The user now sees the following list and can choose a cross street.

6th Street
12th Street
19th / MLK Blvd
Airport Blvd
35th Street
290

It is possible to prevent queries to the Location database. To do so, the configuration file has the word NULL as its SQL command or as the ODBC command. The application will leave the lists in the GUI for highways, streets, ramps, and so forth vacant. Finally, the commands in the configuration file should query the database and never be set to add, delete, or modify it.

2.2.1.4. Location Database

The Location database contains location-specific urban information, such as street names, highway numbers, and cross street names. It might also contain the cross street locations of DMSs. The software relies on the configuration file to provide commands that allow it to query the Location database. The use of a Location database with the MOST system is optional.

Information contained in the configuration file configures the communications and commands that the application needs to communicate with a Location database. The example in Section 2.2.1.3 illustrates this. In the example, the MOST system queries a Location database for highways and cross streets.

The MOST application reads from this database but never writes to, deletes from, or updates it. If the names of tables in the database change or if a different database product is installed, the commands in the configuration file must be changed.

2.2.1.5. History Tables

History tables contain data entered and messages saved from previous sessions. The tables allow a user to reuse messages and data. As shown in Figure 1 of Section 2.2.1, the application has an external interface with other applications. MOST allows third-party applications to query its History tables to retrieve the Suggested Messages by the date and sign for which it was created. A third-party application would take the message from the table and display it in a DMS.

One of the History tables is called TypeOfDMS, and it contains information about DMS signs. Using the dimensions and placement of a sign in relation to the highway, the user can create a message to fit within its display capabilities. The table can store data such as mounting height and vertical curve design speed of the road. This information might come from the vendor of the sign or from TxDOT engineering offices. The user can update information about a sign or create a new entry for a sign. This information is useful in identifying the number of UOIs the message can contain. However, not having this data does not prevent the application from building useful messages.

2.2.1.6. DMS Tables

DMS tables contain all the terms used in messages. Terms are grouped into categories, and each category is associated with a step along a task path that leads to building a message (see Section 3.8.4. for more about task paths). At each step, the user picks one or more of these terms from the GUI. Internally, the application identifies terms with a session by storing links to terms within the History tables.

The DMS tables are used to calculate the maximum and adjusted maximum UOIs. A maximum UOI is determined using traffic speeds, type of DMS, and time of day. A message can have no more than seven UOIs. Other factors, such as environmental conditions and geometric design of the highway, reduce the number of UOIs, called the adjusted maximum UOI number. This latter number constrains how many terms a message can have.

2.2.1.7. Guidelines Tables

The Guidelines tables contain online information about how to construct a message as well as rules that the application uses to make the Suggested Message. The user can query online help for more information about topics, such as recommended good practices in making messages. The tables also include a Rules table. The application uses this table to replace terms with others suggested by subject area experts, such as TxDOT traffic engineers or managers. Finally, the Order of Information Priority for Incidents table (see Appendix I) is also a Guidelines table. The application uses it when reducing terms (see Section 3.8.5.1)1.

2.2.2. User Interfaces

This section of the report contains descriptions of the application’s user interface.

2.2.2.1. General User Interface Elements

The MOST application displays a series of screens containing interface elements. Combo boxes allow selection of one item from a list (see Figure 2). A combo box also allows the user to enter text. A text box accepts text typed in from a keyboard, but it does not offer to drop-down any lists. Radio buttons force a choice among options. A group box encloses other controls, such as radio buttons, to define a group related by functionality or purpose, but it is not possible for a user to interact with the group box itself.

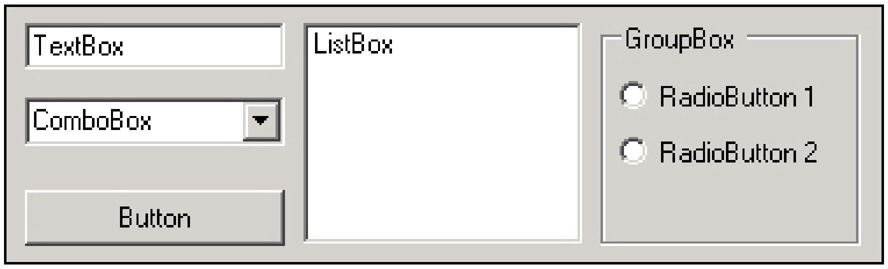


Figure 2. Controls Used in the Application.

Each screen of the application has a resolution of no less than 800 × 600 pixels. A user starts at the top left and works down and across the screen from the upper left to bottom right.

Each screen has several buttons. The Next button closes the current screen and opens the next. The Back button returns the user to the screen most recently visited. The Clear button removes all data entered or selected. The Exit button closes the application. Before it does so, the application presents a message window that asks the user to confirm the exit. In this version of the application, closing the application without saving the Suggested Message results in the loss of all data entered in that session.

2.2.2.2. Online Help

Online help is available on each screen by selecting a Help button. A window opens and displays information about each control on the screen. Selecting the Guidelines button on the last screen gives the user information about the characteristics of effective DMS messages. For example, it shows examples of poorly designed and well-designed messages.

2.2.3. User Interface Prototype

This section of the document describes a prototype of the graphical user interface to explain the functionalities of the system. The appearance and placement of controls as well as the use of screens in an implementation of the MOST GUI does not necessarily match those shown in this document. The document separates functions into numbered screens to to simplify the explanation of functionalities. The features described serve as examples of desirable functionalities, except for those in sections labeled mandatory.

2.2.3.1. Screen 1, Characteristics of DMS, Environment, and Traffic at Event

Figure 3 has a picture of Screen 1, which is the first screen of the application. It has three group boxes: DMS Information, Lighting Conditions, and Traffic Characteristics. It also has a list box containing several categories of events, two of which are incidents and roadwork. Screen 1 has buttons labeled Save DMS Information, Saved Sessions, and Go To Message. Like all screens except the last, it has Next, Clear, Exit, and Help buttons.

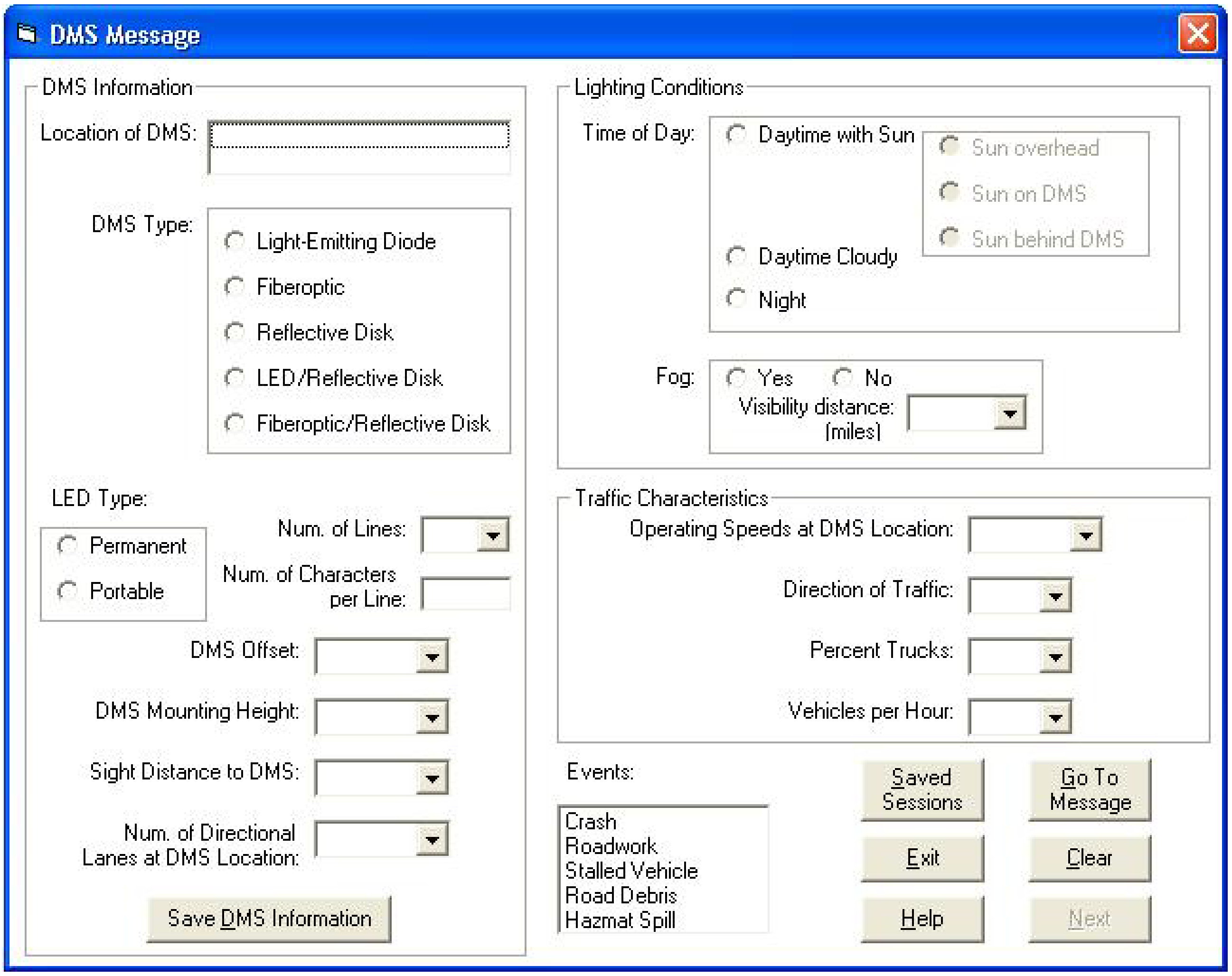


Figure 3. Screen 1, Form for Data about DMS Characteristics, Environmental Conditions, and Traffic Characteristics.

The user must know if a DMS is near an incident or roadwork. In the DMS Information group box, the user can manually enter data about the location and type of DMS sign. By default, the application retrieves data about dynamic message signs from the History tables. The Location of DMS combo box displays a name, location, or other identification for a sign. The software fills in data about the sign from the TypeOfDMS table.

2.2.3.1.1. Name of a DMS appears only once

The History tables contain data about past sessions, but they also contain information about signs. The application gathers this information through Screen 1. By default, the name of the DMS is its location, but the user can change the name by typing a new name over the old one in the Location of DMS combo box. The MOST application will replace the old name with the new one in the History tables. This applies whether overwriting existing information about a sign or entering data for a new entry.

By default, the combo box will show one listing, which is a blank space. A generic dynamic message sign has three lines, two frames, and 18 characters per line. If the user enters a name into the blank space, the application saves the data for the sign under that new name. However, the application will always show the blank line in the combo box.

The software checks for and, if necessary, regenerates the blank line in the list of the combo box. There are three conditions when it will do this:

1. the user saves the session,
2. the user selects the Clear button, or
3. the user revisits the screen from another screen.

The application cannot overwrite the data associated with the generic sign in the History tables. Instead, the application will create a new record under the new name. If the user enters information but not a name, the application prompts the user to give the DMS a new name. The software will not allow the user to save session data unless the DMS name is entered or selected.

The application has a mechanism to prevent the name of a DMS from appearing twice in the Location of DMS combo box. Immediately after a user enters the name and leaves the combo box, the application searches for that name in the History tables and Location database. If the name is not found, that name is stored in the TypeOfDMS table. The name and data are saved when the user selects the Save DMS Information button or saves the session at the last screen. Thereafter, the application will retrieve data about a sign from the History tables using that stored name. There can only be one entry for a DMS sign under a given name in the History tables.

Once entered, if the application might finds that the name exists in the History tables or Location database, the application displays a message to the user explaining that that name already exists. If the system contains these data in the History tables, the application asks the user if old data are to be overwritten. If the name is found in the Location database, the application informs the user that, by saving under that pre-existing name, they will not change data in the Location database under that DMS name. The software does not change data in the Location database. The MOST application will save the data in the TypeOfDMS table. This means that changes to data in the Location database are not automatically propagated to the MOST system. It is important that users are aware of this.

The user may elect to keep the name or cancel the operation. If cancelled, the message window is closed and the name shown in the Location of DMS combo box is changed to a blank line. If accepted, the name in the combo box is unchanged. However, the application does not save data under that name unless the Save DMS Information button is selected or the session itself is saved.

2.2.3.1.2. Mandatory features and adding to the list of events

The other three parts to the screen are Lighting Conditions, Traffic Characteristics, and Events. Items in the Lighting Conditions group box are optional. Choosing from the Events list is mandatory. The user must identify the type of event. The prototype shows several types incidents — crash, stalled vehicle, road debris, and HAZMAT spill. With regard to roadwork, there is only one entry. Designers and developers can add more types of events to the list box. They need to be sure that the screens shown thereafter are correct for the event and will collect all data appropriate for the event. This document does not discuss this expansion of functionality. A module for each type of event should be made to keep the functionalities coded for different kinds of events separate.

2.2.3.1.3. Searching for saved sessions

The MOST application can retrieve data from previously saved sessions. If a user selects the Saved Sessions button, a Saved Sessions window will open. This window will contain dates, times, DMS locations, and types of events in a list box. The application shows these data as rows by session. The window also has OK and Cancel buttons. The user selects a row and the OK button to close the Saved Sessions window and populate the application with data from that saved session. The Cancel button will close this window, but data will not fill the application.

The user can specify values to use in a search. For example, if the MOST application has no information in Screen 1, selecting the Saved Sessions button will display all messages. But if the user entered data into Screen 1 and then selected the Saved Sessions button, the application searches for and displays data about sessions containing those data. Selecting a saved session and choosing the OK button in the Saved Sessions window will close this window and populate the application with data from that saved session.

For example, if fog is present but no other information is entered, the Saved Sessions window displays data for sessions with that environmental condition. To narrow the search further, if the user selects a DMS type of LED and environmental condition of fog, the search will list sessions when both conditions are true. The search applies a Boolean AND condition to data in Screen 1 during these searches. Progressively more refined searches are possible with more selection entered.

2.2.3.1.4. Using a previously saved message immediately

Because all sessions culminate with a message, saving a session will store the message and data for later use. When the Saved Sessions button on Screen 1 is chosen, data from the previously saved session will populate the DMS message session data structure (see Section 3.5). This makes the data available to all screens. Moreover, using a previously saved session will enable the Go To Message button. If the user selects the Go To Message button, Screen 1 will close and Screen 8 (see Section 2.2.3.8 in the example), which is the last screen, will open. The application does not process the message but simply shows the message from the session. At this point, if saved, it stores the message and all data to the History tables under the new date and time and for the DMS chosen. At that time, the message is available to third-party applications for immediate display to the public.

2.2.3.1.5. Showing screens according to event

After selecting the Next button, the application closes Screen 1 and opens Screen 2 (see Section 2.2.3.2) if the event was an incident. If the event is roadwork, Screen 5 (see Section 2.2.3.5) opens. The titles of Screen 2 or Screen 5 say what the event was. The screens that follow Screen

2 are Screens 3 and 4; for Screen 5 they are Screens 6 and 7. The final screen for either sequence is Screen 8.

2.2.3.2. Screen 2, Incident Data Collection

The title of this screen contains the name of the event selected from the Event list in Screen 1. For example, in Figure 3, the event chosen was Crash, and so "Crash" appears as the title for Screen 2, as shown in Figure 4. Note that the group box that contains all GUI controls in this screen also bears the name of event.

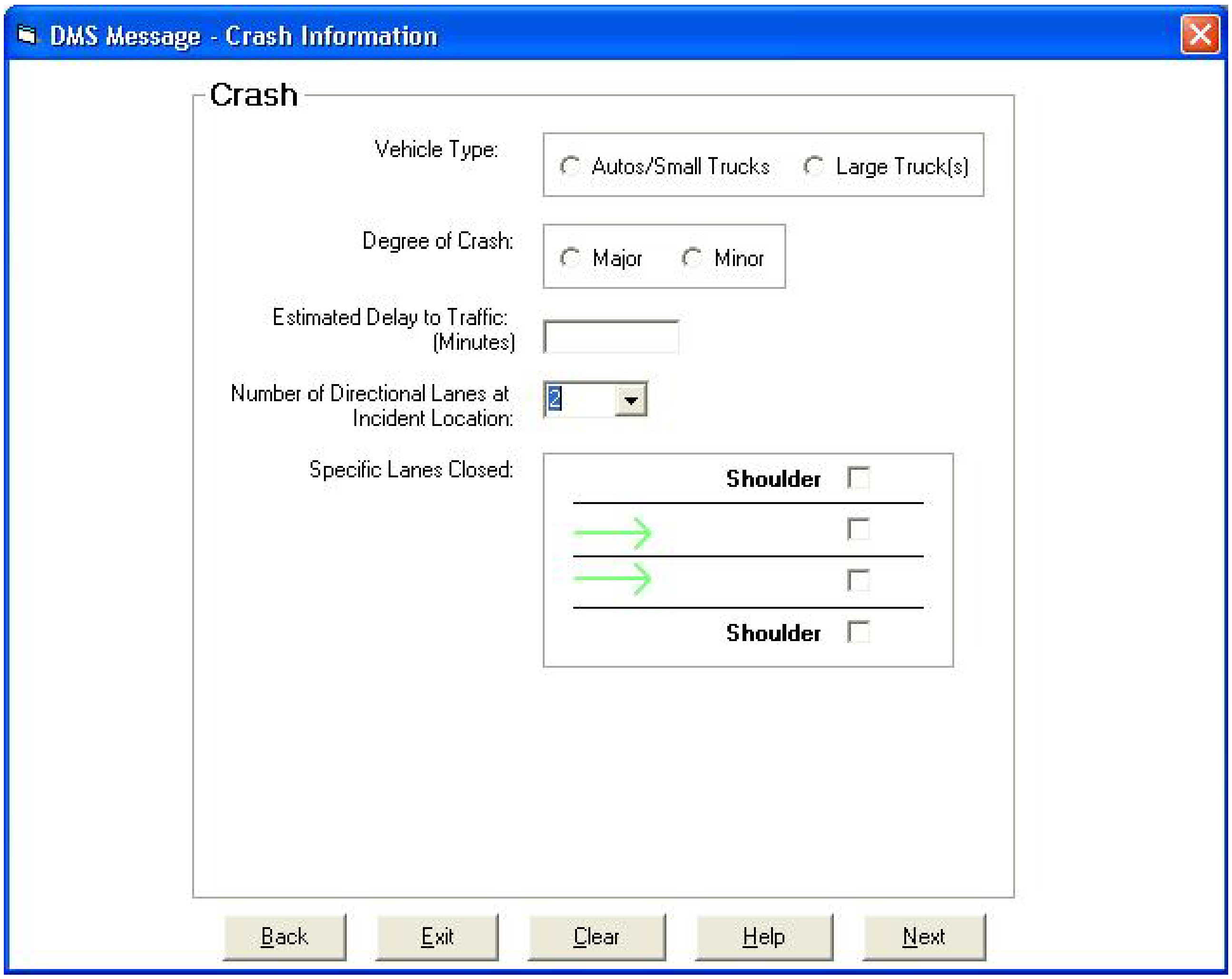


Figure 4. Screen 2, Form to Specify Type and Degree of Incident and Lane Closures.

In Screen 2, the user has the option of choosing the kind of vehicle involved and the severity of the event. The user can enter how long an event is expected to delay traffic. For non-crash events, such as a stalled vehicle, road debris, or HAZMAT spill, the Type and Degree of Crash group boxes are not shown. Only the Number of Directional Lanes at Incident Location combo box and Specific Lanes Closed graphic are shown.

2.2.3.2.1. Mandatory selections

A mandatory selection in this screen is identification of lanes affected by the event. On this screen, the Specific Lanes Closed graphic shows the number of lanes in one direction at the incident. The default number is two unidirectional lanes. If the user chooses more than two lanes through the Number of Directional Lanes at Incident Location control, the graphic changes to show that number of lanes. For example, if the user selects 5, the graphic shows five lanes in one direction.

There is a checkbox in each lane or shoulder. The user selects a check box to indicate if a lane and/or shoulder are closed to traffic. More than one check box can be selected. An unselected check box means that the lane is open.

2.2.3.2.2. Assumption about choices

The program assumes that all lanes are open by default. It will not show a message box asking the user to confirm if lanes or shoulders of the road are open or closed. From this screen, the Next button closes Screen 2 and opens Screen 3.

2.2.3.3. Screen 3, Data Collection about Incident Location

The title of Screen 3 indicates the type of event and number of lanes closed. This screen collects information about the location where the incident occurred.

2.2.3.3.1. Mandatory selections

In this screen, there is one question the user must answer. Failure to answer the question when the Next button is selected results in an error message. The program will inform the user that the question must be answered.

The mandatory selection in Screen 3 is from a group box called DMS Proximity to Event. The user must indicate where the DMS is located in relation to the incident. Internally, the program will select terms to display to the user depending on where the DMS is in relation to the incident. Depending on the needs of the customer, another selection might become mandatory. If required, the customer can require a selection from "Have police/traffic control arrived?" Selecting the Next button closes Screen 3 and opens Screen 4. If no selection is required from "Have police/traffic control arrived?" then the software will use the default value of `"False"` unless the customer states otherwise.

2.2.3.3.2. Showing incident and lane closure status in the title

As shown in the title of Screen 3, Figure 5, the incident was a crash and traffic control personnel have closed or will close all lanes. Incident and lane closures are key information to the choice of task path taken and, consequently, terms shown in Screen 4.

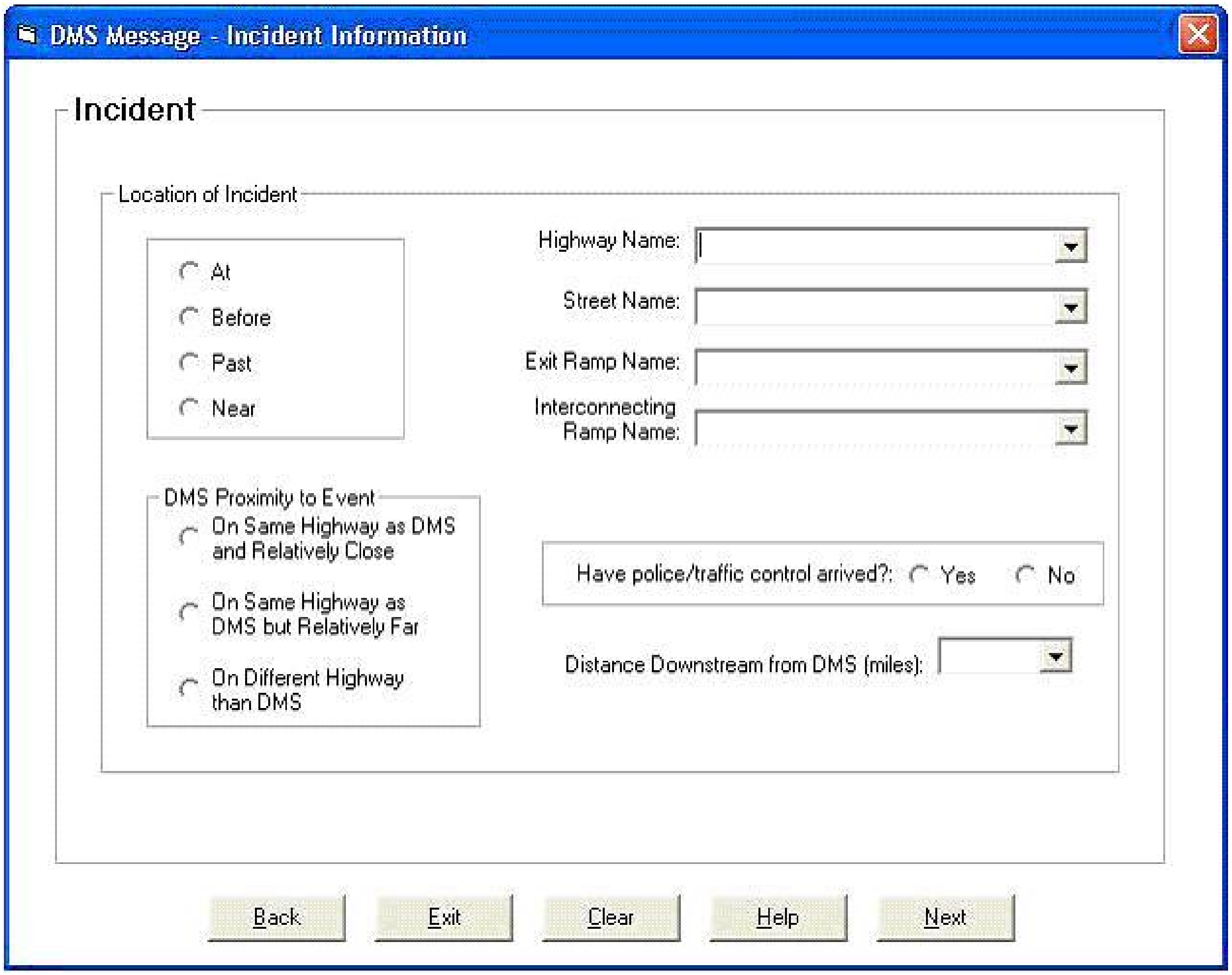

DMS Message - Incident Information
Incident
Location of Incident
At
Before
Past
Near
Highway Name:
Street Name:
Exit Ramp Name:
Interconnecting Ramp Name:
DMS Proximity to Event
On Same Highway as DMS and Relatively Close
On Same Highway as DMS but Relatively Far
On Different Highway than DMS
Have police/traffic control arrived?: Yes No
Distance Downstream from DMS (miles):
Back
Exit
Clear
Help
Next

Figure 5. Screen 3, Form to Specify Incident Relative to DMS.

2.2.3.4. Screen 4, Selecting Terms for the Basic DMS Message for Incidents

It is in Screen 4 that the MOST application allows the user to select terms and build a message. The behavior of this screen is optionally dynamic. This means that when the screen opens, all group boxes except the top leftmost are disabled. The user can only use the enabled controls of the leftmost group box. An enabled control or group box label uses black font. Disabled component labels are in gray font.

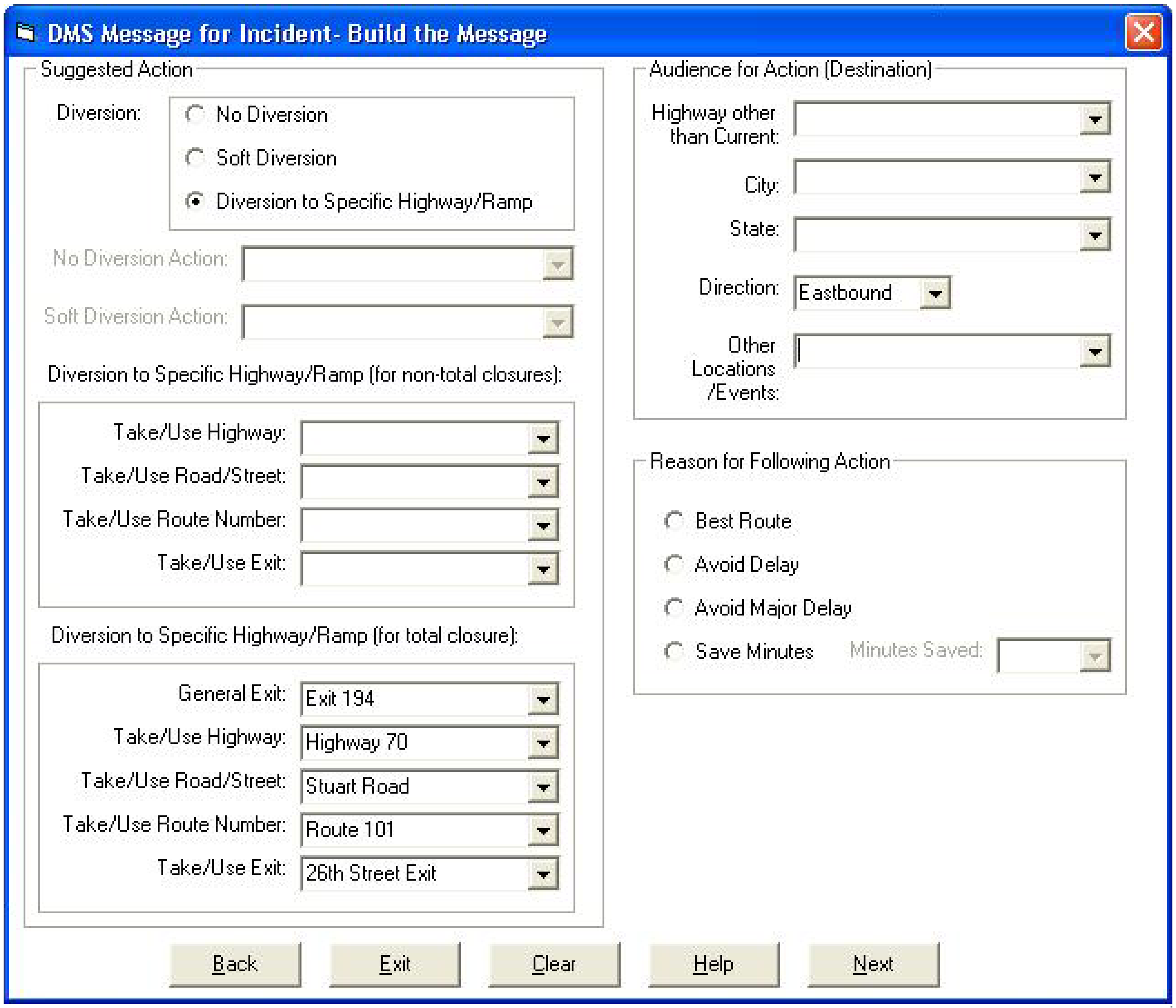
2.2.3.4.1. Mandatory selections

A selection from the Diversion group box within Suggested Action is mandatory. After choosing from this group box, the program guides the user to the next group box or set of controls by enabling them, depending on the user's selection. The user selects from group boxes or sets as the application enables them in a top-to-bottom and left-to-right progression across Screen 4. A task path determines this order of progression. (This report gives more information about task paths in Section 3.8.4.)

In the group box called Reason for Following Action, if Save Minutes is chosen, the user must select from Minutes Saved. Because this group box contains radio buttons, to clear a choice the user must select the Clear button. The Clear button will clear all choices entered and return the progression along the task path to the first activated group box.

2.2.3.4.2. Change to a selection made

If a user changes an entry in a previously visited group box or set of controls, the program disables all controls and inactivates group boxes downstream from that point. The MOST application then positions the progression of the user along the task path at the step where the user made the change. The program associates this step and the one to follow with group boxes or controls, and it can enable or disable them according to the step on the task path. For demonstration purposes only, an example of Screen 4 in Figure 6 shows all group boxes and controls enabled.


DMS Message for Incident- Build the Message

Suggested Action

Diversion: No Diversion [✓ Diversion to Specific Highway/Ramp] Soft Diversion

No Diversion Action:

Soft Diversion Action:

Diversion to Specific Highway/Ramp (for non-total closures):

Take/Use Highway:

Take/Use Road/Street:

Take/Use Route Number:

Take/Use Exit:

Diversion to Specific Highway/Ramp (for total closure):

General Exit:	Exit 194
Take/Use Highway:	Highway 70
Take/Use Road/Street:	Stuart Road
Take/Use Route Number:	Route 101
Take/Use Exit:	26th Street Exit

Audience for Action (Destination)

Highway other than Current:

City:

State:

Direction: Eastbound

Other Locations /Events:

Reason for Following Action

Best Route

Avoid Delay

Avoid Major Delay

Save Minutes

Minutes Saved:

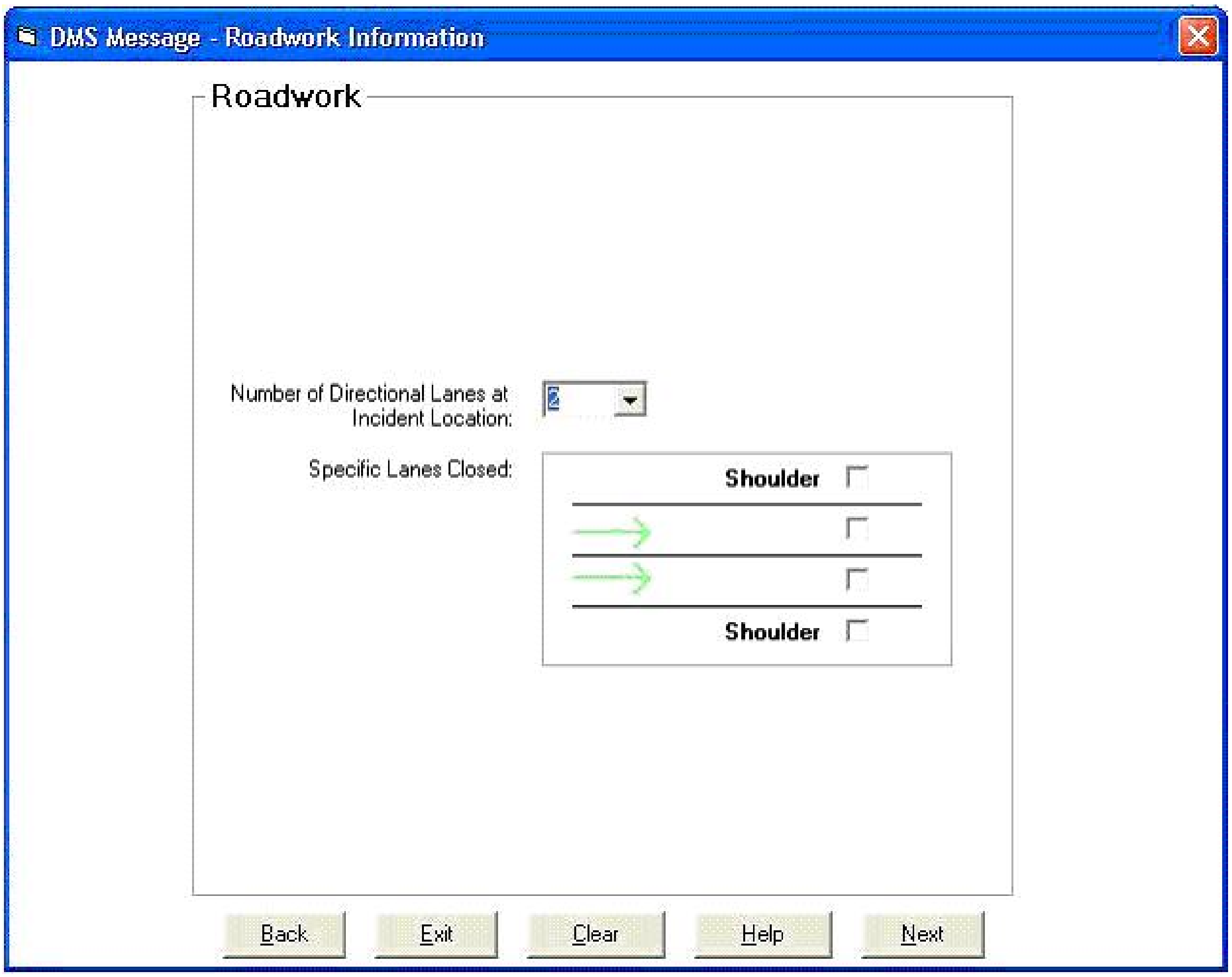
Figure 6. Screen 4, Form to Make DMS Basic Message for Incidents.

2.2.3.4.3. Progression to the last screen

When the user selects the Next button, the application will close Screen 4 and open Screen 8.

2.2.3.5. Screen 5, Lane Closure for Roadwork

Messages for roadwork follow task paths similar to those for incidents. If the user chooses Roadwork from the Events list in Screen 1 (see Figure 3) and then selects the Next button, Screen 1 closes and Screen 5 opens. The program does not display Screens 2, 3, and 4. Starting with Screen 5 (Figure 7), all later screens are about roadwork. Screens 5, 6, and 7 prompt the user to enter data about the roadwork, and then the software builds an appropriate message.



Number of Directional Lanes at Incident Location: 2

Specific Lanes Closed:

Figure 7. Screen 5, Form to Indicate Lane Closures Caused by Roadwork.

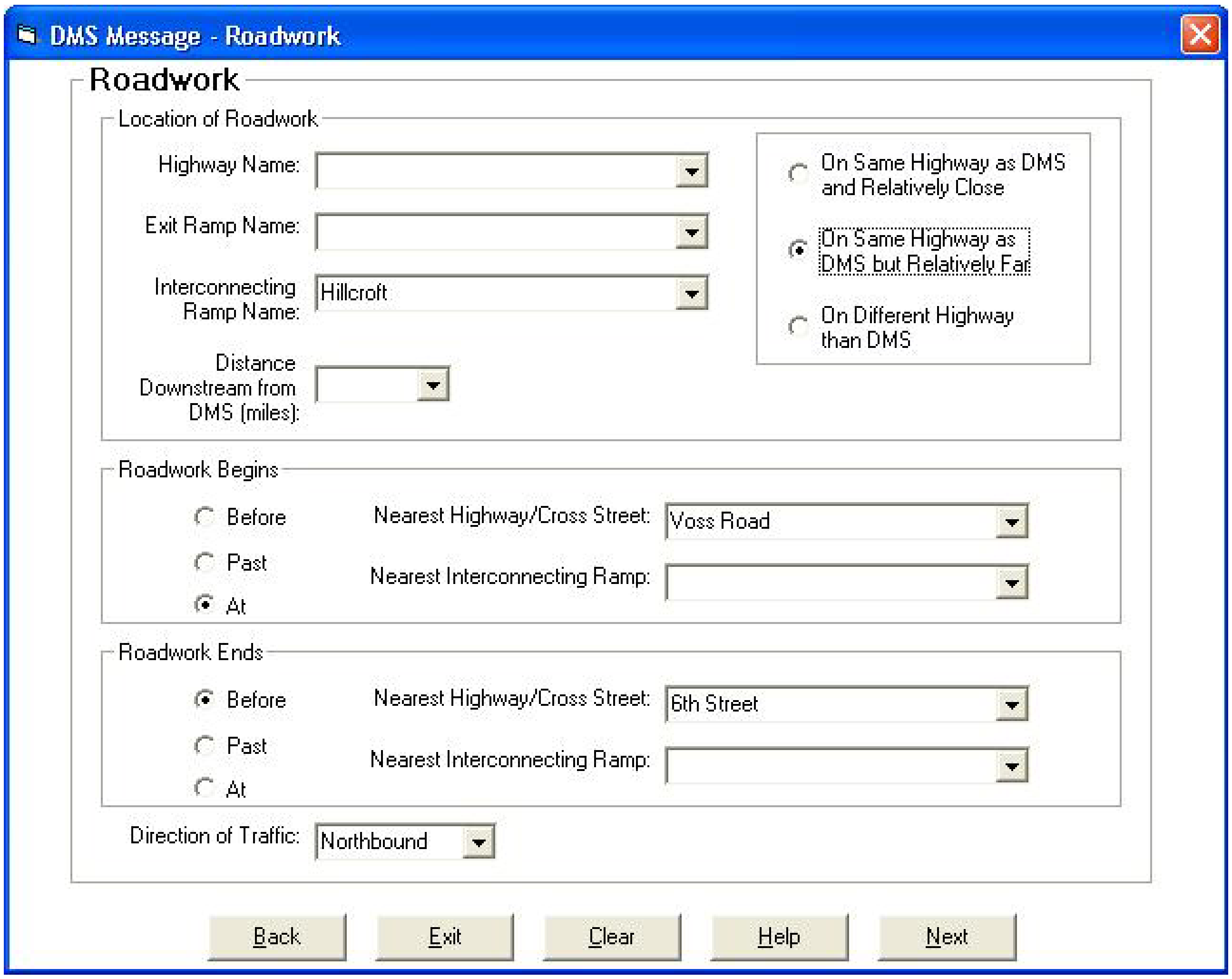
Screen 5 is similar to Screen 2; however, the user can only indicate which lanes are affected by the roadwork. This selection is mandatory. The application does not display a message asking the user if all choices have been made. The Next button closes Screen 5 and opens the next screen, Screen 6.

2.2.3.6. Screen 6, Roadwork Location

All information in Screen 6 (Figure 8) is optional except for the radio button group box contained within the Location of Roadwork group box.

2.2.3.6.1. Mandatory selections

The program must know if the DMS is on the same highway and near the roadwork, on the same highway but far from the roadwork, or on a different highway from the roadwork. All other information selected replaces placeholder variables used in terms of the message.


DMS Message - Roadwork

Roadwork

Location of Roadwork

Highway Name:

Exit Ramp Name:

Interconnecting Ramp Name: Hillcroft

Distance Downstream from DMS (miles):

On Same Highway as DMS and Relatively Close [✓ On Same Highway as DMS but Relatively Far] On Different Highway than DMS

Roadwork Begins

Before Past [✓ At]

Nearest Highway/Cross Street: Voss Road

Nearest Interconnecting Ramp:

Roadwork Ends

[✓ Before] Past At

Nearest Highway/Cross Street: 6th Street

Nearest Interconnecting Ramp:

Direction of Traffic: Northbound

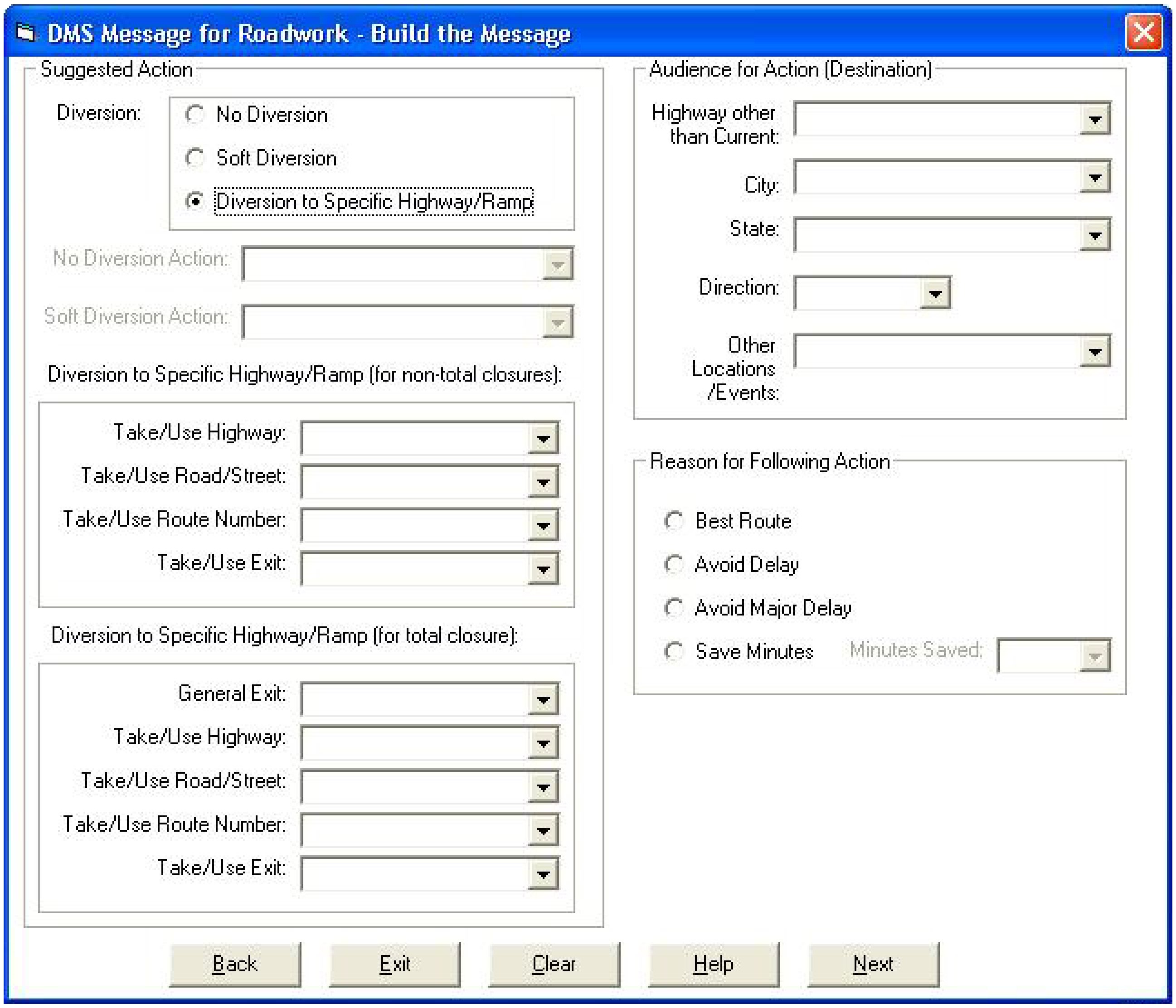
Figure 8. Screen 6, Form to Specify Location of Roadwork.

2.2.3.7. Screen 7, Selecting Terms in the Basic DMS Message for Roadwork

Screen 7 (Figure 9) is similar to Screen 4. By selecting or choosing from controls in activated group boxes, the user selects terms. Like Screen 4, shown in Figure 6, the user chooses in a left-to-right and top-to-bottom progression. Following a task path, a group box or control to be used next is dynamically activated after a selection is made.

2.2.3.7.1. Mandatory selections

The user must select from the Diversion radio button group box in the Action group box. Depending on the task path in effect, the other group boxes or sets of controls may become mandatory. Finley *et al.* (3) contains flowcharts illustrating mandatory steps along different task paths.


DMS Message for Roadwork - Build the Message

Suggested Action

Diversion: ( ) No Diversion ( ) Soft Diversion [● Diversion to Specific Highway/Ramp]

No Diversion Action:

Soft Diversion Action:

Diversion to Specific Highway/Ramp (for non-total closures):

Take/Use Highway:
Take/Use Road/Street:
Take/Use Route Number:
Take/Use Exit:

Diversion to Specific Highway/Ramp (for total closure):

General Exit:
Take/Use Highway:
Take/Use Road/Street:
Take/Use Route Number:
Take/Use Exit:

Audience for Action (Destination)

Highway other than Current:
City:
State:
Direction:
Other Locations /Events:

Reason for Following Action

( ) Best Route
( ) Avoid Delay
( ) Avoid Major Delay
( ) Save Minutes Minutes Saved:

Figure 9. Screen 7, Form to Build Basic DMS Message for Roadwork.

More than one group box can be activated, depending on choices made earlier in the task path.

2.2.3.7.2. Progression to the final screen

For roadwork Screen 7 is followed by the last screen, Screen 8.

2.2.3.8. Screen 8, Editing, Selecting, Saving, and Submitting Messages

This is the last screen. Screen 4 for incidents and Screen 7 for roadwork both lead to Screen 8 (Figure 10).

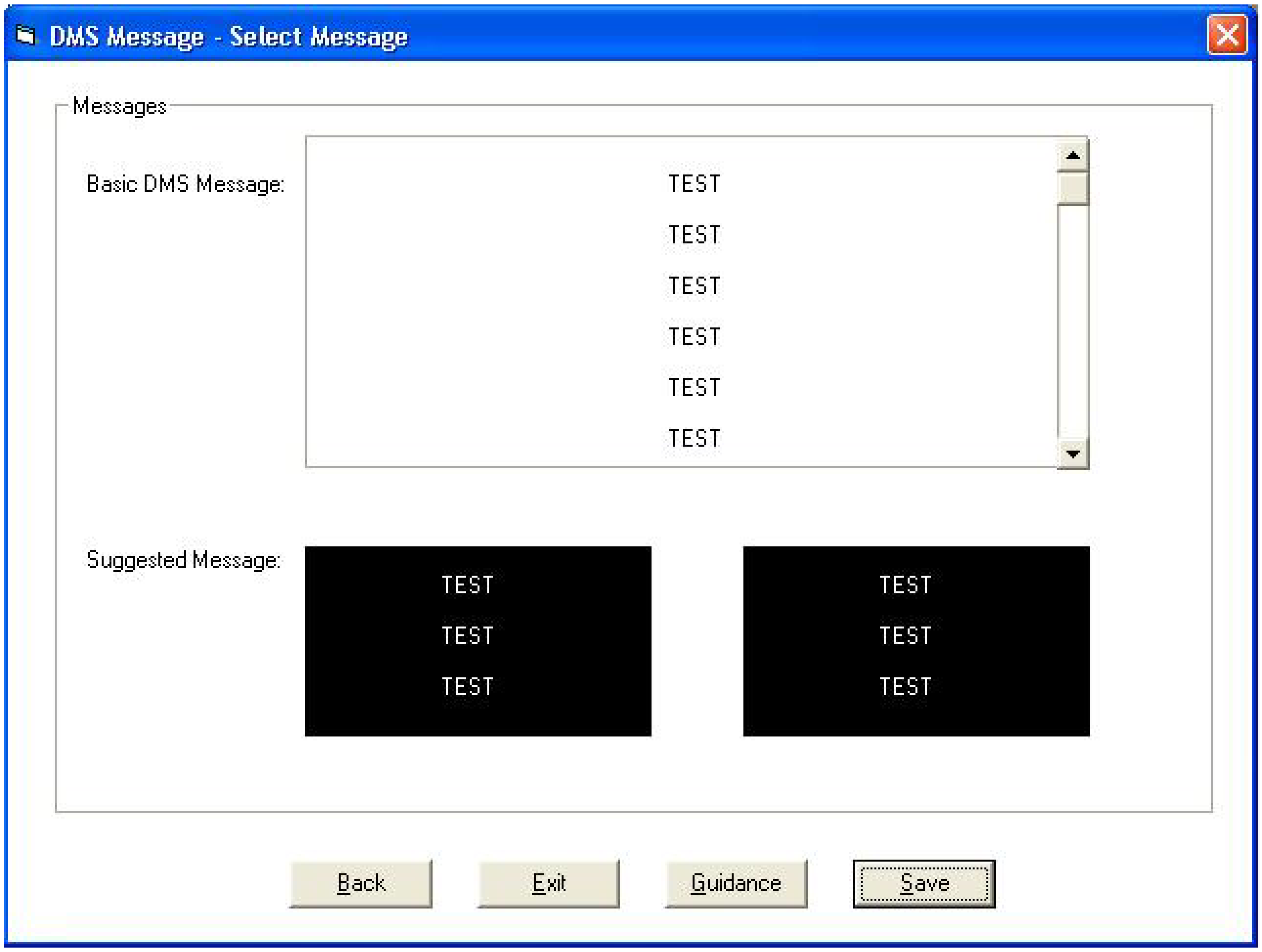


Figure 10. Screen 8, Form to Evaluate and Save Suggested Message.

In this prototype, the MOST application shows the Basic DMS Message in the upper half of the screen. Below it is the Suggested Message. The user cannot edit either the Basic DMS Message or the Suggested Message. If the user selects the Save button, the program saves the message to the InputData table of the History tables and the contents of the DMS Messages Session data structure to the History tables under a unique identification number. The user can change any selection during a session, and the results will be saved under that unique number. The report provides more information about these data structures in Chapter 3.

If the user is using data from a previously saved session and if the Go To Message button was selected in Screen 1, the saved message is retrieved from the History tables and written to the Suggested Message field on Screen 8.

If the user wants to know more about how to build well-designed messages, the Guidance button will open a dialog window containing information and examples. The information for this window comes from the Guidelines table. It will present several scenarios of events and show messages with good and poor designs.

2.3. HARDWARE INTERFACES

The minimum recommended type of computer for the MOST system is a Pentium 500 MHz IBM PC or compatible. However, some versions of the operating system (see Section 2.4) might require a computer with a faster processor speed.

The computer should have an Ethernet card with a bandwidth of at least 10 Mbps if connections with remote databases are planned. The user may find a 100-Mbps Ethernet card preferable if the capability of the network permits. Installation of the application requires a CD-ROM drive or an Internet connection. The user console display should have a resolution of no less than 800 × 600 pixels with a minimum color depth of 16 bits.

2.4. SOFTWARE INTERFACES

The minimum operating system is a 32-bit version of Microsoft Windows®, such as NT 4.0, 2000 Professional, XP Professional, or a later release. The version of Microsoft Windows used must support loading and reading XML trees.

If it will use a Location database, the application requires connectivity dependent on the database used. There are many databases available, such as Microsoft SQL Server, Sybase®, Oracle, Microsoft Access, and MySQL. ODBC is the database connectivity protocol recommended for any ODBC-compliant database. Oracle databases require both a propriety SQL*NET and ODBC. It is better to use native database connectivity drivers if possible. In general, vendors build native drivers to improve interaction with their database product. By doing so, they can reduce the execution time required for an SQL command. By using ODBC or native drivers and a network, the Location database can be, but does not have to be, on the same computer where the MOST application is installed.

2.5. COMMUNICATIONS INTERFACES

To send and receive data over networks, the application uses TCP/IP as the underlying communications protocol.

2.6. MEMORY

A minimum recommended random access memory (RAM) is 256 MB.

2.7. NORMAL AND SPECIAL OPERATIONS REQUIRED BY THE USER

2.7.1. Modes of Operation

A user starts and uses the application manually. The application does not support batch processing. The application is designed as a stand-alone system, but it can optionally communicate with remote databases.

2.7.2. Back-up and Recovery Operations

TxDOT should make periodic copies of all databases and files used with this product. It is especially important that they back up the History tables regularly. These tables contain all messages and data saved for events. If users operate the application extensively, recreating the History tables by manually reentering data might be difficult at best.

Recovery of files consists of copying the application back onto the computer, checking that the ODBC connections are in place, and confirming that communication with any third-party applications not specified within this document are successful. The client can use updated configuration and preprocess files but should always check that the application is retrieving data and that the data are of a kind expected. Users should update tables and files only when the application is not in use.

2.8. SITE ADAPTATION REQUIREMENTS

There are no extensive adaptations required to integrate the MOST system with systems at a TMC. If the program is to use a Location database, the MOST application will require a database connection to that database. To use the MOST database accompanying the application, an ODBC connection to it is mandatory. Users should test ODBC connections for successful queries and updates before the application is used.

There are adaptations required of the configuration file to a site. A technician familiar with writing SQL commands and, possibly, stored procedures must enter and save commands in this file. To do so, this report recommends that each SQL command contain a NULL value. Once the technician establishes database connectivity, they should write and test SQL commands to retrieve data as labeled in the file, but not by using the MOST application. There are tools like Microsoft Query that can test if the SQL command or stored procedure retrieves the correct data. Once the command is confirmed, the technician should enter that command in the configuration file for the kind of data it retrieves and check that the MOST application correctly displays that data in an appropriate list box on the screen. The data should replace the placeholder variables in the list. All data retrieved from the Location database must be text strings.

2.9. PRODUCT FUNCTIONS

This section contains information about the major functions of the system without giving details about how each function works. Refer to Chapter 3 for detailed explanations about program functions.

The software system “MOST” performs the following major functions:

· allows retrieval and editing of information stored from previous sessions;
· collects and stores data that are useful to creation of a message specific to an event;
· provides terms, or units of information, that the user selects to build a message;
· applies a set of rules to modify the Basic DMS Message into a suggested version designed to be more readable and understandable to motorists;
· can apply expert knowledge to single terms or combinations of terms in a Basic DMS Message to create a Suggested Message;
· can automatically substitute abbreviations for words;
· can move or remove terms in the Basic DMS Message to accommodate physical characteristics of the DMS, such as number of lines and number of characters per line;
· allows the user to save the Suggested Message in a saved session for later use;
· provides online help about the application and about choosing terms to build a message; and
· can use a database of location-specific information, if such a database is available.

2.10. USER CHARACTERISTICS

In this section is information about skills required to use the MOST system.

Users should be comfortable using a mouse, computer keyboard, and computer monitor. Visually impaired users can use third-party text-to-voice to guide text entry and list selections. Because the application does not make sound, hearing-impaired users are not hindered.

2.11. CONSTRAINTS

This subsection contains descriptions of constraints on developers during production of the software system.

Developers are subject to budgetary and time constraints of a funded project with a defined start and end date. At scheduled times, they should present reviews of their progress and work completed to peers and managers during review meetings. The developers must show that the product meets requirements (verification) during development and at completion. During and after development, developers create test cases to confirm the correctness of implementation. Management with oversight over the project should conduct a final review to confirm that the product meets TxDOT expectations (validation).

2.12. ASSUMPTIONS AND DEPENDENCIES

The MOST application requires that technicians who install the application include all files and databases (except the Location database, which is optional). They must test all database connections and name the database connection to MOST databases (not the Location database) as “MOST.” (The configuration file contains the name and connection parameters for the Location database, and they can customize these names and parameters as long as they are not called “MOST”). There are no logins or passwords required to use the MOST database connection; however, the customer may test logins and passwords. If used, there is no assurance that the MOST application will connect successfully to its own database.

CHAPTER 3:
SPECIFIC SOFTWARE REQUIREMENTS

3.1. INTRODUCTION

This section contains the design of the system with sufficient detail to allow developers to build the system. The material in Chapter 3 is written for developers.

All inputs and outputs are identified as well as associated functionalities. Aspects of the system include external interfaces, performance requirements, logical database requirements, design constraints, and software system attributes.

Chapter 3 gives the design of the system through functional requirements, which describe the system on the basis of functionality in terms of inputs and outputs. It accomplishes this through the use of process descriptions, data construct specifications, and pseudocode.

3.2. EXTERNAL INTERFACES

The application interfaces with the following entities:

· users,
· TxDOT TMC databases, and
· third-party applications.

The MOST application has a GUI for interaction with the user. The functionalities required for data exchange with TxDOT TMC Location databases are built in but not essential.

The following use case diagram (Figure 11) shows interfaces between the program and external entities.

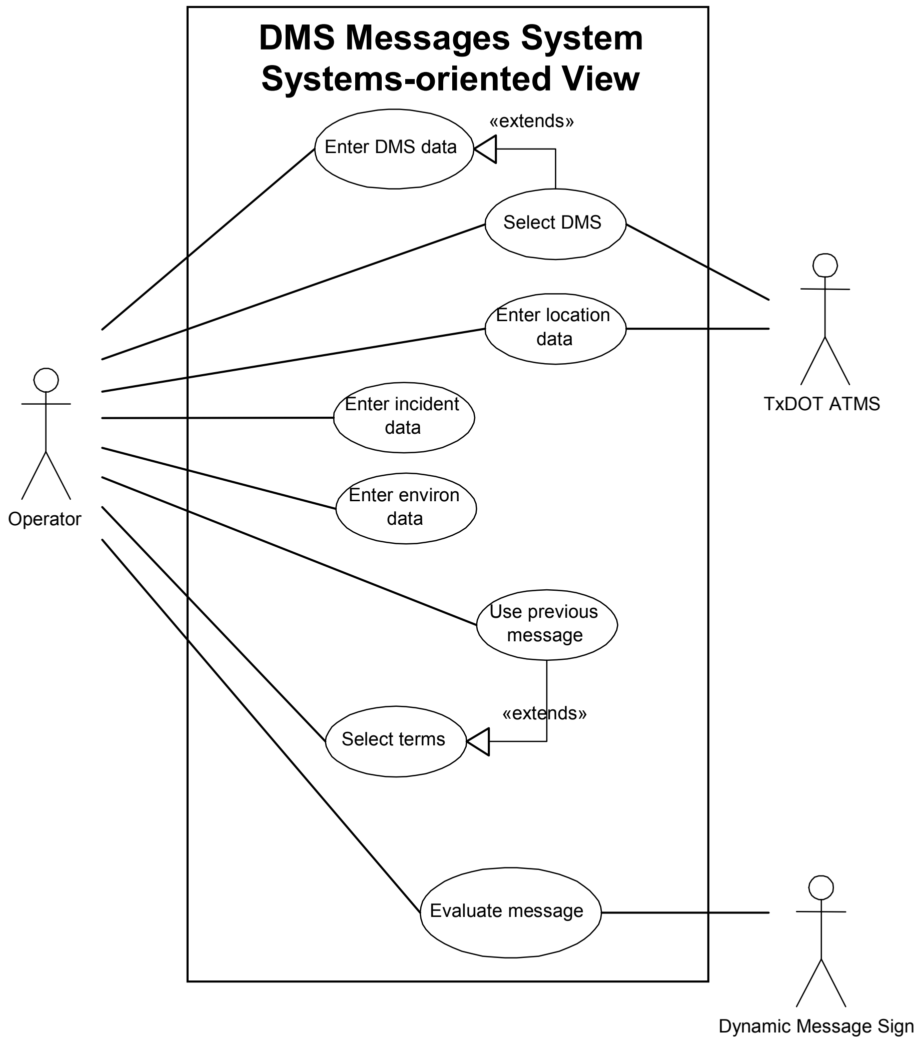


Figure 11. Use Case Diagram of DMS Messages.

As shown in Figure 11, a user can enter or select information about a DMS and location of an event as well as environmental conditions. A user can manually enter information about a DMS or extract data from a Location database. The user builds a message by selecting terms from lists or modifying a message created from a previously saved session.

3.3. PERFORMANCE REQUIREMENTS

With the graphical user interface, users navigate between screens by selecting the Next or Back button. The time between the closing of one screen and opening of the next should not exceed 30 seconds. Time required for database access and queries necessary to fill list boxes should not exceed 30 seconds. When a user starts the MOST application, the time required to display Screen 1 should be no more than 30 seconds.

3.4. DATABASES, TABLES, AND ASSOCIATED FILES

Chapter 2 introduced the tables and files that the system uses. Refer to that chapter for more information about them. This section contains more specific information.

3.4.1. Preprocess File

Once made, users should not edit the preprocess file. Described in Section 2.2.1.2., the MOST application reads this file at program initialization. Appendix A contains a table showing the organization of data. In Appendix A, each row represents a task path. A row begins with a single character followed by a character string and then a sequence of integers. The first character identifies the path. The strings identify variables within the program. The integers in the sequence identify the categories of tasks, and their sequence is the order that application will apply to the categories.

Finley *et al.* (*3*) prepared flowcharts that identified categories of terms. Selections from these categories, when combined, build a Basic DMS Message. The Finley *et al.* report numbers the tables sequentially, and the MOST system uses that numbering to uniquely identify terms. TxDOT will provide the terms.

To implement the categories, a technician must store the UOIs, or terms, to the Terms table of the DMS tables. There, the application references them by category number. The category number corresponds to the table number as used in the Finley *et al.* report. Thus, the number 38 refers to the “No Diversion” category as an Acceptable Action, where the DMS is on the same freeway but relatively close to the incident.

To construct a message, the application queries the DMS tables for a category of terms based on the current step along the task path. Only one step at a time may be taken along the task path; the next step depends on the choices made that led up to it. To accomplish this, the application activates group boxes selectively, one-by-one, or several simultaneously as each step along the path is taken.

After each selection, the application queries the DMS tables for the next category from the preprocess file. Because each row in the preprocess file represents a complete task path, the final message is complete once the application has reached the last category for the path. The execution sequence along a row will not jump across rows as shown in the table in Appendix A. The task paths consist of categories of terms as well as algorithms that determine which category to load. This report provides more information about those categories in Section 3.8.3.

3.4.2. Configuration File

The configuration file contains information in XML that is necessary for the MOST application to communicate with the Location database. If the system uses a Location database, a technician must customize the configuration file for the TMC. The configuration file consists of three parts, a database connection string, a wild-card character, and a table of SQL commands. See Appendix B for a schema and example of the file.

In overview, the first element in the configuration file is the database connection string. A person can write this as either an ODBC string or as a native driver string. The application does not contain hard-coded logins and passwords and does not provide a place in the configuration for them. The name of the database connection from client to server must be called "`MOST.`" If no Location database is available or if the application will operate in stand-alone mode, use "`NULL`" instead of a connection string. Use "`NULL`" for any SQL command that is not possible or will not be implemented.

In Appendix B, another element contains the label "database wildcard." The value is a wild-card character that the RDBMS requires for queries. (Wild-card characters are ASCII symbols used in the "`WHERE`" phrases of a SQL command, and they vary across vendors.) For example, Microsoft SQL Server and Oracle databases use a '`%`' sign, whereas Microsoft Access uses a '*'. A wild-card character identifies all or parts of strings similar to the template in the SQL command applied.

The rest of the configuration file consists of a table of SQL commands, as shown in Table 3.

Table 3. Configuration File Showing SQL Commands.

Label	Attribute Name	SQL Command (Examples)
HighwayName	<Highway_name>	SELECT <Highway name> FROM | NULL
NumberOfLanes	<Number_of_lanes>	SELECT <Number of lanes> FROM | NULL
City	<City>	SELECT <City> FROM … | NULL
RampName	<Ramp_name>	SELECT < Ramp_name> FROM | NULL
RouteNumber	<Route_no>	SELECT <Route_no> FROM | NULL
ExitNumber	<Exit_no>	SELECT <Exit_no> FROM | NULL
CrossStreet	<Cross_street>	SELECT <Highway name> FROM | NULL
Park	<Park>	SELECT <Park> FROM | NULL
Event	<Event>	SELECT <Event> FROM | NULL
Stadium	<Stadium>	SELECT <Stadium> FROM | NULL
DMS	<DMS_location>	SELECT <DMS_location> FROM | NULL

A label is the name of a variable used within the application. It is important that the technician who performs the customization of the file not change the label entry. The label identifies the SQL command as well as results returned by the query. Information about the other two columns of Table 3 follows.

An SQL command is a statement that instructs the RDBMS to perform some action. An attribute name is the name of a column in the Location database. The file provides the attribute name to developers so that they have the name of the column that is the source of data for that label.

The SQL command must return data from its query using that column name. The examples show the notation `SELECT…|NULL`, which means that the file contains either a SQL command or no command (`NULL`).

Appendix B shows a sample configuration file schema and file built using XML. Because the required operating system platform supports the XML language, the application can read files formatted with XML. Therefore, developers must include code to load, search, and extract values from XML-formatted files that have the XML schema.

Here is an example of an SQL command. If the TxDOT Location database uses the attribute name `Highway_Name` in the table Freeways, then the SQL command associated with the label "HighwayName" is as follows:

```
SELECT Highway_Name FROM Freeways
```

The application refers to results returned in this query by the variable called "HighwayName."

Although the configuration file provides space for only one SQL command per label, this does not limit its flexibility. Table 3 shows only simple SQL commands as examples; however, complex statements are possible. If several SQL commands are necessary to extract data for a label, a stored procedure is recommended. Sometimes stored procedures return results faster than SQL commands. In this case, the SQL command for a label is then an invocation of the stored procedure along with parameters sent.

3.4.3. Entity Relationship Diagram

The entity relationship diagram (ERD) refers to tables listed by Finley *et al.* (*3*). The InputData and DMSMessage tables are called History tables. These store session data. The InputData, Input_IncidentDescriptor, IncidentDescriptor, and Terms tables are, as a group, called the Terms tables. (The InputData table serves multiple roles, depending on the context of use.) All tables except DMSMessage and InputData are called the DMS tables. Appendix C shows the attributes of these tables. The flowcharts shown in Figures 2 and 3 of the Finley *et al.* report utilize the database as diagrammed in the ERD below (Figure 12).

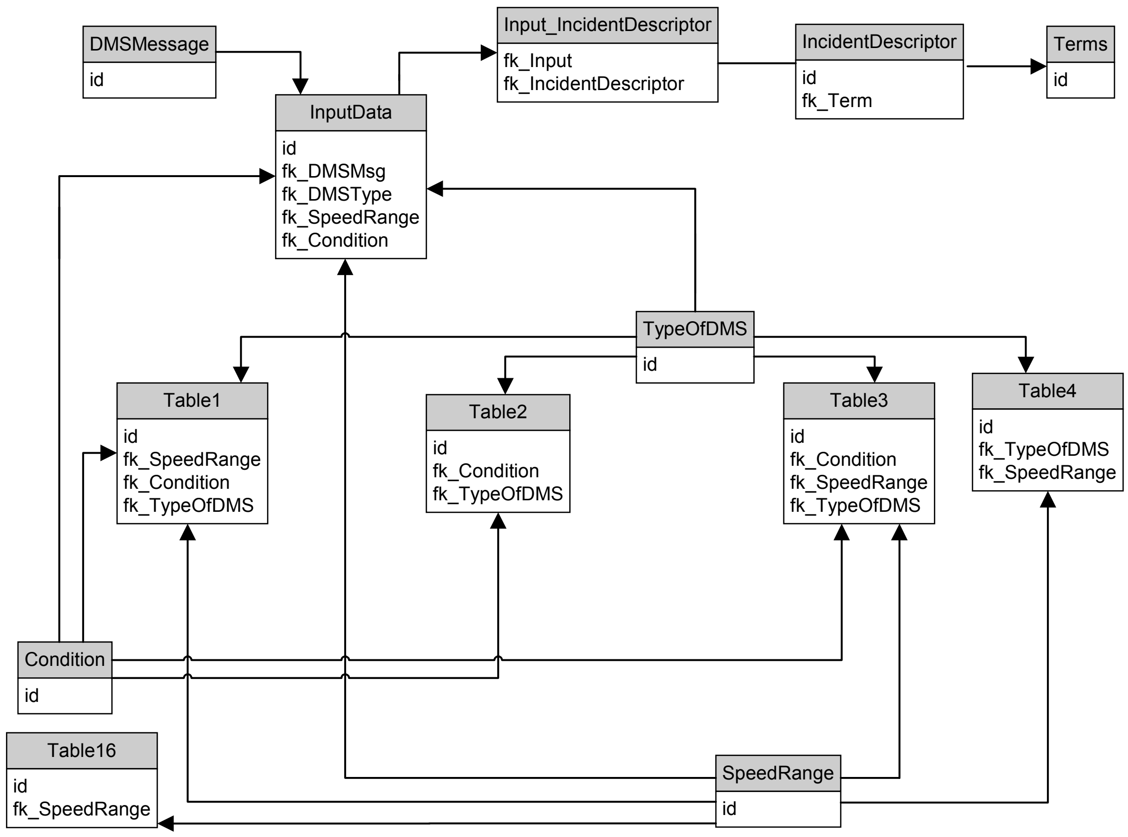


Figure 12. Entity Relationship Diagram for the MOST System.

Subsections below describe each table of the ERD.

3.4.3.1. InputData

The application saves all data that the user enters or selects to this table. The attribute called id is the primary key.

3.4.3.2. Input_IncidentDescriptor

This table contains a key composed of two foreign keys. One foreign key is a copy of the primary key of a record in the table InputData. The other key is a copy of the primary key of a record in the IncidentDescriptor table. The purpose of this table is to allow for a one-to-many relationship.

3.4.3.3. IncidentDescriptor

The records of the IncidentDescriptor table contain category numbers. Category numbers correspond to the numeric titles given to tables of terms by Finley *et al.* (*3*). The IncidentDescriptor table also contains whether the term applies to a large or a portable sign. A foreign key links to the Terms table. If the attribute IsLargeSign is 'T' then IsPortableSign must be 'F,' and vice versa. The copy of the primary key, called id, is used as a foreign key in the table Input_IncidentDescriptor.

3.4.3.4. Terms

This table contains all possible terms. TxDOT will provide these terms. Each record consists of a key and a term, and each record is unique. The primary key is called id.

3.4.3.5. Table1

Note that the name of the table is “Table1” and that there are no spaces within the name. The data for this table comes from Table B1, “Maximum Number of Units of Information in DMS Message (Base Maximum Message Length)” by Finley *et al.* (*3*). The key is the id attribute. The table has links to the tables SpeedRange, Condition, and TypeOfDMS. The primary key is called id.

3.4.3.6. Table2

The source of data for this table comes from Tables B2.1, B2.2, B2.3, and B2.4 from the Finley *et al.* report. The application uses this data to calculate the Adjusted Maximum UOI. The id attribute is the key. The table links to the Condition and TypeOfDMS. Note that the name of the table is “Table2” without any spaces within the name.

3.4.3.7. Table3

Values come from Tables B3.1, B3.2, B3.3, B3.4, B3.5, and B3.6, “Number of Units of Information that Must Be Subtracted from Number Given in Table 1 Due to Horizontal Curve” by Finley *et al.* (*3*). It has links to Condition, SpeedRange, and TypeOfDMS. The data in this table are also used to calculate the adjusted Maximum UOI. The primary key is called id, and the name of the table does not contain spaces.

3.4.3.8. Table4

The id attribute is the key. Data comes from Table B4 whose title is “Number of Units of Information that Must Be Subtracted from Number Given in Table 1 Due to Effects of Fog in Daytime Conditions” by Finley *et al.* (*3*). It has links to TypeOfDMS and SpeedRange. Note that the name of the table is “Table4” and that there are no spaces within the name.

3.4.3.9. TypeOfDMS

This table contains a primary key called id, and this table stores the types of DMSs that the program uses in calculations of the adjusted maximum UOI. For this document, these values are “Light-Emitting Diode,” “Fiber Optic,” “Incandescent Bulb,” and “Reflective Disk.” It also can contain other information entered by a user about physical characteristics of the sign or certain engineering parameters as indicated by the names of its attributes. The user stores values into this table by selecting the Save DMS Information button in Screen 1 (see Figure 3). The application uses the combination of primary key and location of the DMS to identify message signs. It is from this table that the application retrieves specifics about a sign to display in the MOST application. It is also from this table that values used in the calculation of the adjusted maximum UOI are taken, if available.

3.4.3.11. Condition

This simple table has records with a primary key of id and a value for condition, which are “Mid-day,” “Washout,” “Backlight,” or “Nighttime.”

3.4.3.12. SpeedRange

With a primary key called id, this table contains the lower and upper boundaries of speed ranges. The default values are “0–35,” “36–55,” and “56–70.”

3.4.3.13. DMSMessage

DMSMessage has a primary key value that is used as a foreign key in the InputData table. The DMSMessage table stores the message that the user saves at the conclusion of a session.

3.4.5. Guidelines Tables

There are four tables called, in general, Guidelines tables. They are the Guideline table, the Rules table, the Expert table, and the Abbreviations table. The Guideline table (see previous sentence.)?] table consists of the following attributes (see Table 4).

Table 4. Guidelines Table.

Attribute	Purpose
Id	Primary key
Topic	Title of Guideline
Guideline	Name of the document.

If a user selects the Help button on Screen 8, a window opens to show a list of topics. The contents of the list come from the Guideline table. If the user selects a topic, the application opens a document in either the same or another window. The document provides information related to the topic selected. The name of this document is stored in the Guideline table and is associated in the table under that topic. The user can scroll and close the List of Topics and document windows. The user can also print the document.

The Rules table is another of the Guidelines tables. The application uses this table to look up reductions in the working UOI. More information about this reduction is in Section 3.8.5.1.2.

The third Guidelines table is the Expert table. This table contains matches between terms as shown in the Terms table and replacement terms. It allows users to customize terms without changing the standard terms in the Terms table (see Section 3.8.5.1.2.)

The Abbreviations table is a fourth table of the Guidelines tables. It consists of a table with three columns, which contain a primary key, an unabbreviated word, and an abbreviation of that word. For example, the table might contain the following (see Table 5).

Table 5. Abbreviations Table.

Id	Unabbreviated	Abbreviated
1	Boulevard	BLVD

3.4.6. Location Database

Section 2.2.1.4 contains a description of how to use a Location database. The report does not stipulate the design or implementation of the Location database, but it does require it to be a relational implementation using a relational database management system. The specification requires the application to communicate with the database by an ODBC or native database connection driver. The database is likely to be an implementation of a commercial product from vendors like Microsoft, Oracle, or Sybase. All three products have ODBC and native drivers available, bundled as a part of Microsoft Windows or available from the vendor.

The DMS Messages application can query but never update, delete from, or insert into the Location database. The application requires a login and password to the database with read-only privilege. The database administrator for the RDBMS must grant this privilege.

3.5. DATA STRUCTURES

The application will maintain two data structures in dynamic memory. The DMS message session data structure (see Appendix D) can store all data that the user inputs or selects as well as terms selected. The message design data structure (see Appendix H) contains the terms selected as well. The application displays the Built Message using values from the DMS message session structure. It derives the Suggested Message by using the message design data structure. The user can not change the Suggested Message.

3.6. DESIGN CONSTRAINTS

The time required to complete development of the MOST application is constrained by negotiation with TxDOT with regard to time, people, and resources needed.

3.7. SOFTWARE SYSTEM ATTRIBUTES

The system relies on ODBC connections to exchange data between the application and the MOST database that accompanies the application.

3.7.1. Reliability

The number of times that the tool starts and delivers a message to users is a measure of its reliability. If the MOST application cannot establish database connection with its MOST database, the application will state in a message window that it cannot find the database. Once the user selects the OK button in the window, the application will close.

3.7.2. Availability

The application will be available as long as the computer on which it is installed has power and the operating system and environment are working correctly. To start the tool, a user double-clicks an icon on the desktop for the tool or starts it by using the Start button. TxDOT sites must determine access rights to the MOST application. Because the MOST database is made using Microsoft Access, only one person can use the application at a time. This is because Microsoft Access will lock the write permissions on the MOST database while someone is using it.

3.7.3. Security

There is no login to the application. A more secure database connection to the Location database is possible with a login and password combination in the connection string. Because the application creates messages and stores them in the History tables, third-party applications must have read-only login and password privileges to the MOST database before they can acquire the message for display in a DMS.

3.7.4. Maintainability

The organization that develops the product should provide software updates and patches. Distribution of these updates depends upon the wishes of TxDOT sites. Two methods of distribution are CD-ROM and downloads from a Web site. These programs and files contain information that is specific to the installation site. A person at the site should back up the DMS database regularly.

3.7.5. Portability

The MOST system is designed for the Microsoft Windows and Intel x86 computing platform as described in Section 2.4. It is not designed for other operating systems or computer architectures.

3.8. FUNCTIONAL DESCRIPTION

This section of the MOST application requirements document describes the functionality of the application. Figure 2 of TTI Research Report 4023-1, by Finley *et al.* (*3*) provides a series of flowcharts that describe the construction of the Basic DMS Message and the final Suggested Message.

The following algorithms and diagrams are illustrated by the flowcharts in the Finley *et al.* report, which are not reproduced in this report.

3.8.1. Data Structures

The application will create two key data structures in dynamic random access memory. Other data structures are possible.

3.8.1.1. DMS Message Session Data Structure

The application will store data that the user enters or selects into a data structure. This data structure ceases to exist when the user terminates the application. If a user selects the Save button in Screen 8, the application will save all data from the structure into the History tables. The data structure has fields to contain the data as shown in Appendix D, “DMS Messages Data Structure.” The fields accept one or more text strings for terms delimited with semicolons.

3.8.1.2. DMS Message Design Data Structure

This data structure contains the Suggested Message. The terms that the user selects are stored in the Basic DMS Message. The application uses terms from the Built Message to put together [construct?] the Suggested Message. The DMS Message Design data structure is then a work pad for the application.

The data structure contains fields for each category of terms. Terms are stored as text strings, and each field can contain more than one string. Semicolons delimit concatenated strings.

3.8.2. Calculation and Adjustment of Maximum UOI

Use Table1 of the DMS tables to look up the maximum UOI using the type of DMS, speed range, and lighting conditions (mid-day, washout, backlight, or nighttime). Perform a query on Table1 for the attribute value of UOI.

Use the tables shown in Appendix C to query Table2, Table3, and Table4 of the DMS tables with the following parameters.

- Type of DMS,
- LED type,
- Sign type,
- Condition,
- Vertical curve design speed,
- Vertical curve LED mounting height,
- Vertical curve portable LED mounting height,
- Horizontal curve radii,
- Horizontal curve offset,
- Speed range,
- Visibility, and
- Maximum UOI.

The adjusted maximum UOI, or working UOI, is returned.

The following describes the algorithm used to calculate the adjusted MaxUOI, which is called AdjustedUOI. This function or method call is named adjMaxUOI().

```
MaxUOI = GetMaxUOI(TypeOfDMS, SpeedRange, Condition)
VReduction = GetSightRestrictionVerticalCurve()
HReduction = GetSightRestrictionHorizontalCurve()
RainfallReduction = GetSightRestrictionRainfall()
FogReduction = GetSightRestrictionFog()

AdjustedUOI = MaxUOI – (VReduction + HReduction + RainfallReduction +
FogReduction)
```

3.8.3. DMS Messages Data Structure Fields and GUI Controls

Data used to replace placeholder variables and direct execution at decision points come from controls in screens. Appendix D shows the data structure that the application uses to store values from these controls. The screen number and control within that screen are matched to a field in the data structure, as shown in "Matching Controls to DMS Message Data Structure Fields" in Appendix F.

3.8.4. Task Paths

Implementation of a task path begins by applying an algorithm to the sequence of integers from the preprocess file. A step in the algorithm might show or activate one or more lists of terms in the graphical user interface. The user can select, at most, one term from each list. After detecting the selection the terms, the application executes the next step in the algorithm. This process continues until the complete Basic DMS Message is built. Therefore, the result of task paths is the Basic DMS Message. The lists in the GUI appear in Screen 4 (see Figure 6) for incidents and in Screen 7 (see Figure 9) for roadwork.

3.8.4.1. Diagrams of Task Paths

The incidents and roadwork diagrams of task paths come from Finley *et al.* (*3*). The application loads a sequence of numbers for a task path from the preprocess file.

3.8.4.1.1. Incidents.

Figure 13 below shows a flowchart to task paths for incidents. The application uses data entered through Screens 1, 2, and 3 as parameters in decision points.

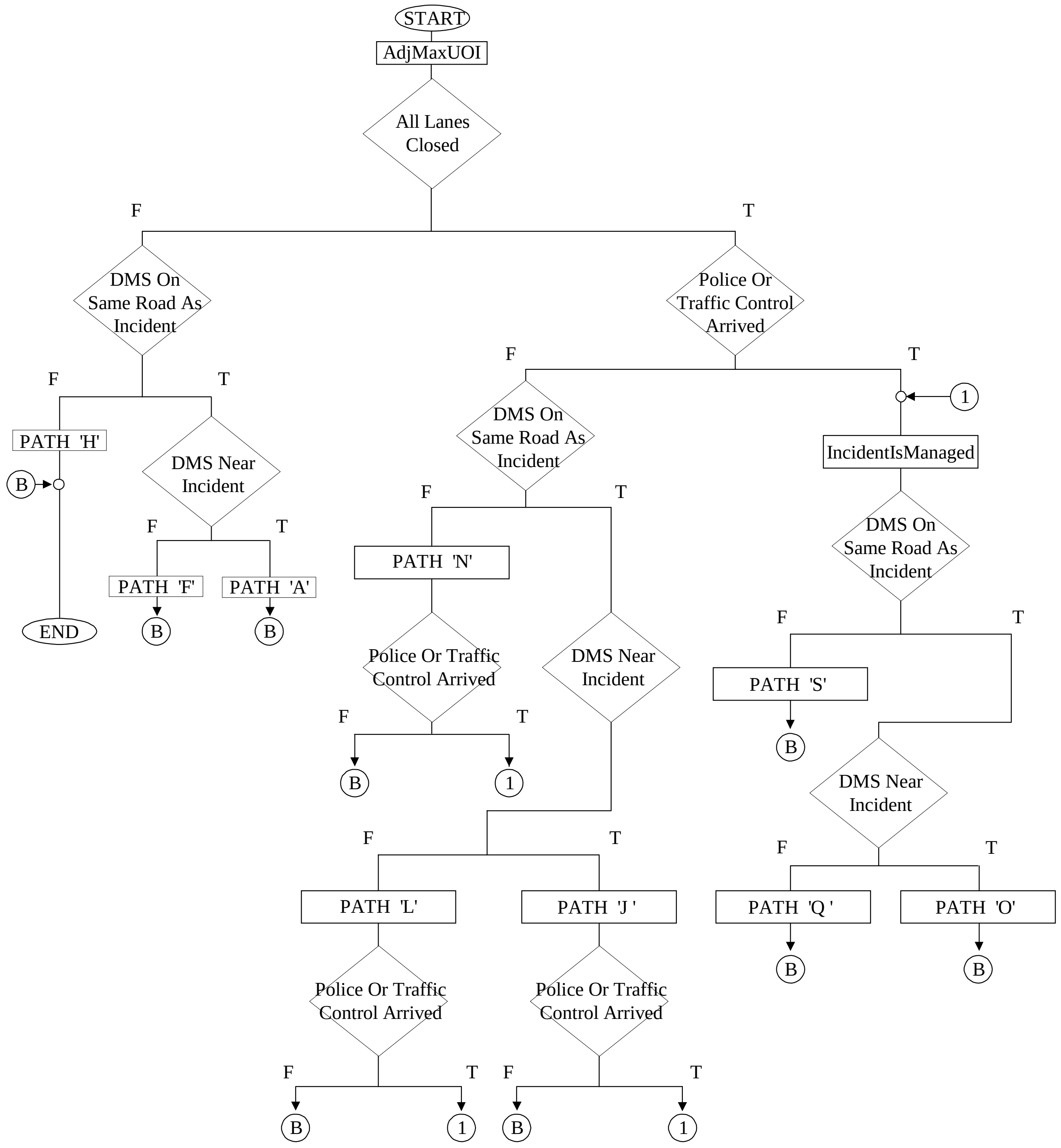


Figure 13. Flowchart of Task Paths for Incidents.

3.8.4.1.2. Roadwork

The flowchart in Figure 14 shows how task paths are chosen when constructing messages for roadwork. The application uses data entered through Screens 1, 5, and 6 as parameters in decision points.

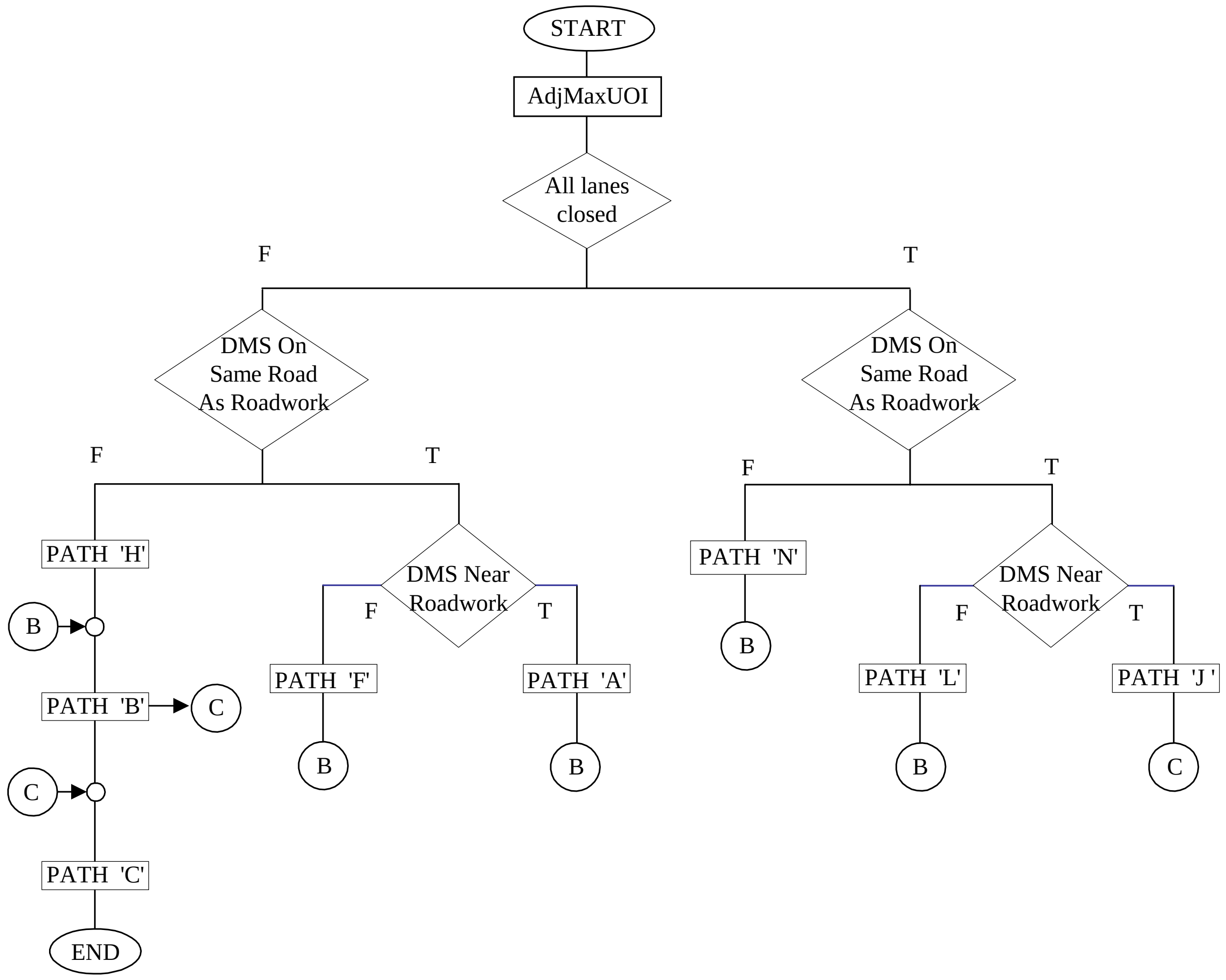


Figure 14. Flowchart of Task Paths for Roadwork.

3.8.4.2. Pseudocode for Task Paths

The pseudocode for task paths for incidents is similar to that of roadwork.

3.9.4.2.1. Incidents

This subsection contains pseudocode that describes flow control of the algorithm shown in Figure 13. It can also be followed in Figure 2 as found in the Finley *et al.* (*3*) report.

3.8.4.2.1.1. Main paths

A main path through Figure 13 proceeds from Start and goes to the on-page connector B. The flow of execution then goes to Path H. Although all paths eventually go Path H, some follow an on-page connector 1. The document does not refer to these as main paths but as IncidentIsManaged paths.

Some lines of the pseudocode below contain comments. The start of a comment begins with double forward slashes, and the comment extends to the end of the statement.

```
Call AdjMaxUOI  // see Section 3.8.2.
If the value for All Lanes Closed is 'F' and
      If Location of DMS to Incident is same road as incident then
            If Location of DMS is near incident then
                  Call PATH 'A'
            Else    // Location of DMS is far from incident
                  Call PATH 'F'
            End if
            Goto on page connector B in PATH 'B'
      Else   // Location of DMS to incident is not on same highway
            Call PATH 'H'
      End if
Else   // All Lanes Closed is 'T'
      If Police or Traffic Control present is 'T'
            Call IncidentIsManaged  // see Section 3.8.4.2.1.2
      Else   // Police or Traffic Control have not arrived
            If DMS is on same highway as incident
                  If Location of DMS is near incident
                        Call PATH 'J'
                        If Police or Traffic Control present is 'T'
                              Call IncidentIsManaged
                        End if
                  Else    // Location of DMS is far from Incident
                        Call PATH 'L'
                        If Police or Traffic Control present is 'T'
                              Call IncidentIsManaged
                        End if
                  End if
            Else    // DMS is not on "Same highway" as incident
                  Call PATH 'N'
                  If Police or Traffic Control present is 'T'
                        Call IncidentIsManaged
                  End if
            End if
      End if
      Goto on page connector B in PATH 'B'
End if
Call ReductionAndFinalization  // see Section 3.8.5.
```

3.8.4.2.1.2. IncidentIsManaged

A branch in paths N, L, and J follows on-page connector 1, which takes execution of the program to paths S, Q, and O. The pseudocode for conditions that direct execution to S, Q, and O is shown in the function or method call named IncidentIsManaged().

```
If DMS is on “Same highway” as incident
      If DMS is near Incident
            Call Subroutine PATH ‘O’
      else
            Call Subroutine PATH ‘Q’
      end if
else  // DMS is not on “Same highway” as incident
      Call Subroutine PATH ‘S’
end if
```

3.8.4.2.2. Roadwork

The following pseudocode is a textual version of the algorithm shown in Figure 14. It can also be followed in Figure 3 of Finley *et al.* (*3*).

Some lines of the pseudocode below contain comments. The start of a comment begins with double forward slashes, and the comment extends to the end of the line.

The pseudocode for Figure 14 follows.

```
Call AdjMaxUOI  // see Section 3.8.2.
If the value for All Lanes Closed is ‘F’ and
      If Location of DMS to roadwork is “Same highway” then
            If Location of DMS is “Near roadwork” then
                  Call PATH ‘A’
                  Goto on page connector B in PATH ‘B’
            Else    // Location of DMS is far from roadwork
                  Call PATH ‘F’
                  Goto on page connector B in PATH ‘B’
            End if
      Else   // Location of DMS to incident is not on same highway
            Call PATH ‘H’
      End if
Else   // All Lanes Closed is ‘T’
      If DMS is on same highway as roadwork
            If Location of DMS is near roadwork
                  Call PATH ‘J’
                  Goto on page connector C in PATH ‘C’
            Else    // Location of DMS is far from roadwork
                  Call PATH ‘L’
                  Goto on page connector B in PATH ‘B’
            End if
      Else    // DMS is not on “Same highway” as roadwork
            Call PATH ‘N’
            Goto on page connector B in PATH ‘B’
      End if
End if
Call ReductionAndFinalization  // see Section 3.8.5.
```

3.8.5. Reduction and Finalization

The reduction and finalization steps are the last two subprocesses in the process of constructing the Built and Suggested Messages. The reduction subprocess ensures that the total number of UOIs in the message is less than or equal to the adjusted maximum number of UOIs calculated in Section 9.1. The finalization step consists of three substeps.

1. Format the message.
2. Adjust the message to fit on three or fewer lines of the DMS.
3. Review and manually edit the message.

Following reduction and finalization, the user can choose to save the message and session.

3.8.5.1. Reduction

The application uses the DMS Message Design data structure in Appendix H to make changes to the Suggested Message. The Built Message is stored in the DMS Message Session data structure and is not altered.

3.8.5.1.1. Unit Reduction by Count of Terms

The following algorithm reduces the number of UOIs.

```
SumOfTerms = Sum of non-NULL terms in Message Design data structure
If AdjMaxUOI < SumOfTerms then
   Query Guidelines table for new terms using the following combination
   in the following order
      1.  Incident Descriptor
      2.  Incident Location
      3.  Lanes Affected
   Replace individually the above terms with results returned from
     Guidelines table
End if

SumOfTerms = Sum of non-NULL terms in Message Design data structure
If AdjMaxUOI < SumOfTerms then
   SumAudienceForAction = Sum of non-NULL entries in Audience For
      Action field of Message Design data structure
   If SumAudienceForAction > 1 then
      Query table as shown in Appendix I with Audience For
         Action to determine priorities of categories
      Delete all Audience For Action entries except highest priority
   End if
End if
processing = true

If Event = Incident Then
   If AdjMaxUOI < SumOfTerms then
      Query DMS Table for priority of terms
      Store query results to IncidentPriorityList
      While processing is true AND IncidentPriorityList is not empty
         SumOfTerms = Sum of non-NULL terms in Message Design data
           structure
         If AdjMaxUOI < SumOfTerms then
```

```
                Delete entry from Message Design data structure matching
                  lowest priority term in IncidentPriorityList as shown in
                  Appendix I
                Delete lowest priority term from IncidentPriorityList
            Else
                processing = false
            End if
        End while
    End if
End if
```

3.8.5.1.2. Replacement of Terms by Application of Expert-Derived Combinations

In some circumstances, engineers or managers might want a certain phrase to replace a certain combination of terms. They could temporarily change the wording of a term in the Terms table, but if the term is not changed back, the original term becomes unavailable for all messages made thereafter. In other words, there is no history of the change from the original. Moreover, the engineers or managers might want users to see a certain phrase if the users choose a certain combination of terms. Changing single terms in the Terms table does not accomplish this. The Terms table does not allow association of terms within the table.

The application applies a very simple form of expert system based on 1:1 substitution. It can reduce the number of UOIs by replacing certain combinations of terms with others as recommended by experts in traffic engineering and management. The listings in Table 6 show categories of terms. These are the same categories found in the preprocess file. (As a reminder, each column in the preprocess file represents a category.)

Table 6. Expert Table.

Unique Identifier Keys	Replacement Strings
Event Descriptor	Event Descriptor
Event Location	Event Location
Lanes Closed	Lanes Closed
Effect On Travel	Effect On Travel
No Diversion Action	No Diversion Action
Closure Location	Closure Location
Soft Diversion	Soft Diversion
Type 1 or 2 Diversion	Type 1 or 2 Diversion
Audience for Action	Audience for Action
Good Reason for Following Action	Good Reason for Following Action
Type 1, 2, 3, or 4 Diversion	Type 1, 2, 3, or 4 Diversion
Type 5 Diversion	Type 5 Diversion

The Expert table contains the Unique Identifier Keys from Table 6. They appear in the same row of the Expert table as they appear, from top to bottom in sequence, in Table 6. In the same row following these keys, the items from the Replacement Strings column appear in the order, from top to bottom, shown in Table 6. The pairings of categories across columns and along the same row must be maintained.

Within a row, from left to right, a user enters the unique identifiers of terms by category. They then enter a phrase in one or more the last 12 columns by category of term. It is the combination of these id keys and their sequence in the first 12 attributes that represents a combination of replacement terms in the last 12 attributes.

Each combination of unique ids must be unique in the Expert table. The unique combination refers to one or more phrases to replace the standard terms. To accomplish this, the application compares the id values for terms in the Suggested Message to values stored in the Expert table. If there is a match, the application replaces terms, category-by-category, with those found in the replacement strings attributes. The application searches the Expert table automatically after the Suggested Message is built but before the message is displayed in the software.

It is possible that no replacement of terms is necessary. Experts can provide terms for which no substitutions are required.

3.8.5.2. Finalization

In this subsection, the presentation of the Built and Suggested Messages are changed to accommodate the dimensions of the DMS. If the Suggested Message will not fit in the sign, an informational message appears to tell the user to use the Back button, change their selections, and have the program remake the Suggested Message.

3.8.5.2.1. Format the Message

In the MOST application, a message sign consists of two frames. Terms are displayed in a top-to-bottom sequence on separate lines within a frame. Sometimes the message has fewer than three lines. Such a message can fit in one frame. Messages with more than three terms are displayed across two frames.

The MOST application can use the tables shown in Appendix G to control the sequence of display of terms. The terms are ordered whether they contain NULL or non-NULL strings. After they are sequenced, the terms are assigned to frames by using the following algorithm.

```
If Lanes Closed does not replace or merge with Incident Descriptor then
   If specific lanes are closed then
      Sequence the terms using Table G
   Else if all lanes are closed then
      Sequence the terms by Table G
   End if

   If Incident Descriptor <> NULL and Incident Location <> NULL and
     Lanes Closed <> NULL and Action <> NULL
     Audience for Action = NULL then
      Assign terms to frames terms following Priority No. 1 in
      Table G
   Else
      If Incident Descriptor <> NULL and Incident Location <> NULL and
        Audience for Action <> NULL and Action <> NULL
        but Lanes Closed = NULL then
```

```
            Assign terms to frames terms following Priority No. 2 in
            Table G
         Else
            If Incident Descriptor <> NULL and Incident Location <> NULL
              and Lanes Closed <> NULL and Action <> NULL
              and Audience for Action <> NULL then
                Assign terms to frames terms following Priority No. 3 in
                Table G
            Else
               If Incident Descriptor <> NULL and Incident Location <> NULL
               and Audience for Action <> NULL and Action <> NULL
               and Good Reason for following Action <> NULL
               and Lanes Closed = NULL then
                  Assign terms to frames terms following Priority No. 4 in
                  Table G
               End if
            End if
         End if
      End if
   Else if Lanes Closed does replace or merge with Incident Descriptor then
      If specific lanes are closed then
         Sequence the terms using Table G
      Else if all lanes are closed then
         Sequence the terms by Table G
      End if

      If Lanes Closed <> NULL and Lane Closure Location <> NULL and
        Audience for Action <> NULL and Action <> NULL
        and Good Reason for Following Action = NULL then
          Assign terms to frames terms following Priority No. 1 in
          Table G
      Else
         If Lanes Closed <> NULL and Lance Closure Location <> NULL and
            Audience for Action <> NULL and Action <> NULL
              and Good Reason for Following Action <> NULL then
                Assign terms to frames terms following Priority No. 2 in
                  Table G
         Else
            If Freeway Closed <> NULL and Closure Location <> NULL
              and Audience for Action <> NULL and Action <> NULL then
                Assign terms to frames terms following Priority No. 3 in
                Table G
            End if
         End if
      End if
   End if
```

The Built and Suggested Messages are displayed to the user after the above algorithm is applied. It is also possible for experts to provide terms such that no special formatting need be applied.

3.8.5.2.2. Review and Manual Editing of the Message

The user cannot edit the Suggested DMS Message. The following five rules are applied, either by the application or by the user. This information comes from Finley *et al.* (3), which also serves as a source of information for the online help.

Rule No. 1. No more than two frames should be used.

If the number of terms in the Suggested Message exceeds the number of lines in the physical DMS, an error message appears. The program instructs the user to eliminate a term.

Rule No. 2. Each frame must be understood by itself.

If the application detects that a term is displayed across frames, it instructs the user to check if the message is understandable. If it is not, the user should select the Back button and change their selection of terms.

Rule No. 3. Compatible units of information should be displayed on the same frame.

Appendix G gives the sequences of compatible terms in frames. If required, the application will check each time before the user sees the Suggested Message to confirm that compatible terms are still together. If they are not, the application displays a warning message to ask the user to confirm that each frame contains an understandable message. It is possible that terms are provided such that the software will not have to checks for compatible terms.

Rule No. 4. A message line should not contain portions of two different units of information.

The application will check if the application has combined terms on the same line. A warning message to the user will ask if the message is still comprehensible. It is possible that the software will place only one term per line as a rule.

Rule No. 5. No more than three units of information should be displayed on a single frame at high freeway speeds.

Whenever the user edits the message, the application compares the total number of UOIs in a frame to a standard value of three UOIs per frame. If the number of UOIs exceeds three, the program instructs the user to reduce the number of terms. The software can be programmed to display only one term per line and never more than three terms per frame.

3.8.5.2.3. Saving the Message and Session Data

If the user selects the Save button in the screen showing the Suggested Message, the application inserts all data from the DMS Message Session into the History tables. No records in the InputData table are ever updated. Each session is considered unique and is time-stamped.

3.8.6. Coordination of Processes and Data within the Application

This section describes how the application uses the structures and coding in Sections 3.8.1 and 3.8.2 in conjunction with the graphical user interface.

3.8.6.1. Matching Decision Points That Determine Task Paths to GUI Controls

As shown in Figures 13 and 14 and described in Sections 2.1.1, 2.1.2, and 2.2, there are three decision, or branch, points in the algorithms. The decision points described below indicate which control in the application acquires the data and where in the DMS Messages Data Structure the value is stored.

Decision: Are all lanes closed?
GUI Control: Lanes Closed
DMS Messages Data Structure Element: LanesClosed
(The size of the LanesClosed array is the value stored in NumDirectionalLanesAtEvent. If all array cells contain a value other than NULL, then all lanes are closed.)

Decision: Is DMS on same road and relatively near to event?
GUI Control: DMS Proximity To Event
DMS Messages Data Structure Element: HighwayDMSProximity

Decision: Is DMS on same road and relatively far from event?
GUI Control: DMS Proximity To Event
DMS Messages Data Structure Element: HighwayDMSProximity

Decision: Is DMS on different road?
GUI Control: DMS Proximity To Event
DMS Messages Data Structure Element: HighwayDMSProximity

3.8.6.2. Matching Categories of Terms to GUI Controls

When a user selects a radio button, an item in a list box, or a combination of these controls, they indicate which unique identifiers to use in queries to the Terms table of the DMS tables. The application stores the unique identifiers in the DMS Messages Data Structure before making the query. Queries are made to select terms from categories of terms. Controls are therefore associated with categories of terms. Appendix E shows these associations.

3.8.6.3. Association of Task Paths to GUI Controls

Appendix A shows the task paths possible when the user constructs a Basic DMS Message. Each task path consists of several steps, each of which is represented by a category of terms. The application has one or more GUI controls matched to a category of terms.

3.8.6.4. Querying for Terms Using Values from GUI Controls

The preprocess file lists categories of terms found in the Terms table of DMS tables. For example, task path A uses categories 5 to 13. These lists of terms come from TxDOT. The user uses the controls in Screen 4 (see Figure 6) and Screen 6 (see Figure 8) to choose terms from the lists.

REFERENCES

1. Dudek, C.L. Variable Message Sign Operations Manual. Report No. FHWA-NJ-2001-10, New Jersey Department of Transportation, November 2001.

2. Dudek, C.L., and R.D. Huchingson. Manual on Real-Time Motorist Information Displays. Report No. FHWA-IP-86-16, FHWA, U.S. Department of Transportation, August 1986.

3. Finley, M.D., T.J. Gates, and C.L. Dudek. DMS Message Design and Display Procedures. Research Report 4023-1. Texas Department of Transportation, November 2001.

4. IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998. The Institute of Electrical and Electronics Engineers: New York, NY, 1998.

APPENDIX A. PREPROCESS FILE

Appendix A. Preprocess File

Task Path	Lanes Affected	DMS Location	Event Descriptor	Event Location	Lanes Closed	Effect On Travel	No Diversion Action	Closure Location	Soft Diversion	Type 1 or 2 Diversion Action	Audience For Action	Good Reason For Following Action	Type 1,2,3 or 4 Diversion	Type 5 Diversion
A	Specific	Near, same freeway	5	6	7	8	9	X	10	11	12	13	X	X
F	Specific	Far, same freeway	17	18	19	20	21	X	22	23	24	25	X	X
H	Specific	Different road	26	27	28	29	30	X	31	32	33	34	X	X
J	All	Near, same freeway	35	36	37	X	38	X	39	40	41	42	X	X
L	All	Far, same freeway	43	44	45	X	46	X	47	48	49	50	X	X
N	All	Different road	51	52	53	X	54	X	55	56	57	58	X	X
O	All	Near, same freeway	59	60	61	X	X	62	X	X	65	66	63	64
Q	All	Far, same freeway	67	68	69	X	X	70	72	X	75	76	73	74
S	All	Different road	77	78	79	X	81	80	82	83	84	85	X	X

An "X" indicates a NULL value.

APPENDIX B. CONFIGURATION FILE SCHEMA AND EXAMPLE

Appendix B. Configuration File Schema and Example

Configuration Schema:

```
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
   <ElementType name="DMS_Config" content="eltOnly">
      <element type ="ODBC" minOccurs="1" maxOccurs="1"></element>
      <element type ="wildcard" minOccurs="1"
maxOccurs="1"></element>
      <element type ="HighwayName" minOccurs="1"
maxOccurs="1"></element>
      <element type="NumLanes" minOccurs="0"
maxOccurs="1"></element>
      <element type="CrossStreet" minOccurs="0"
maxOccurs="1"></element>
      <element type="RampName" minOccurs="0"
maxOccurs="1"></element>
      <element type="Event" minOccurs="0" maxOccurs="1"></element>
      <element type="DMS" minOccurs="1" maxOccurs="1"></element>
   </ElementType>
   <ElementType name="ODBC" content="eltOnly">
      <element type="ODBC_connection_string" minOccurs="1"
maxOccurs="1"></element>
   </ElementType>
   <ElementType name="ODBC_connection_string" content="textOnly"
dt:type="string">
   </ElementType>
   <ElementType name="HighwayName" content="eltOnly">
      <element type="Highway_name" minOccurs="1"
maxOccurs="1"></element>
      <element type="SQL_command" minOccurs="1"
maxOccurs="1"></element>
   </ElementType>
   <ElementType name="Highway_name" content="textOnly"
dt:type="string">
   </ElementType>
   <ElementType name="NumLanes" content="eltOnly">
      <element type= "Number_of_lanes" minOccurs="1"
maxOccurs="1"></element>
      <element type="SQL_command" minOccurs="1"
maxOccurs="1"></element>
   </ElementType>
   <ElementType name="Number_of_lanes" content="textOnly"
dt:type="string">
   </ElementType>
   <ElementType name="CrossStreet" content="eltOnly">
      <element type= "Cross_street" minOccurs="0"
maxOccurs="1"></element>
```

```
      <element type="SQL_command" minOccurs="0"
maxOccurs="1"></element>
   </ElementType>
   <ElementType name="Cross_street" content="textOnly"
dt:type="string">
   </ElementType>
   <ElementType name="RampName" content="eltOnly">
      <element type= "Ramp_name" minOccurs="0"
maxOccurs="1"></element>
      <element type="SQL_command" minOccurs="0"
maxOccurs="1"></element>
   </ElementType>
   <ElementType name="Ramp_name" content="textOnly"
dt:type="string">
   </ElementType>
   <ElementType name="Event" content="eltOnly">
      <element type= "Event" minOccurs="0"
maxOccurs="1"></element>
      <element type="SQL_command" minOccurs="0"
maxOccurs="1"></element>
   </ElementType>
   <ElementType name="Event" content="textOnly" dt:type="string">

   </ElementType>
   <ElementType name="DMS" content="eltOnly">
      <element type= "DMS_location" minOccurs="1"
maxOccurs="1"></element>
      <element type="SQL_command" minOccurs="1"
maxOccurs="1"></element>
   </ElementType>
   <ElementType name="DMS_location" content="textOnly"
dt:type="string">
   </ElementType>
</Schema>
```

Configuration Example:

```
<DMS_Config>
   <ODBC>
   <ODBC_connection_string>SET tdb = OPENDATABASE
      ("Location.mdb")</ODBC_connection_string>
   </ODBC>
   <HighwayName>
      <Highway_name>IH35</Highway_name>
      <SQL_command>SELECT Highway_name FROM </SQL_command>
   </HighwayName>
   <NumLanes>
      <Number_of_lanes>3</Number_of_lanes>
      <SQL_command>SELECT Highway_name, Number_of_lanes FROM
</SQL_command>
   </NumLanes>
   <CrossStreet>
      <Cross_street></Cross_street>
      <SQL_command>NULL</SQL_command>
```

```
  </CrossStreet>
  <RampName>
    <Ramp_name>42</Ramp_name>
    <SQL_command>SELECT Highway_name, Ramp_name FROM
</SQL_command>
  </RampName>
  <Event>
    <Event>Football game</Event>
    <SQL_command>SELECT Event FROM </SQL_command>
  </Event>
  <DMS>
    <DMS_location>East on IH35</DMS_location>
    <SQL_command>SELECT DMS_identification, DMS_location FROM
</SQL_command>
  </DMS>
</DMS_Config>
```

APPENDIX C. ERD OF DMS MESSAGES

Appendix C. ERD of DMS Messages

InputData

Attribute	Description
Id	Key
SessionDate	Date and time stamp
Message	Variable-length ASCII string with semicolons separating lines of the message
IncidentHighway	Variable-length string for highway where event occurred
IncidentTrafficDirection	Character for direction of traffic affected by event
IncidentCrossStreet	Variable-length string for nearest cross street
IncidentRampName	Variable-length string for nearest ramp name
IsFog	Boolean operator for presence of fog
VisibilityRange	Variable-length string for ranges of visibility
HasControlArrived	Boolean operator for whether police or traffic control have arrived
AllLanesClosed	Boolean operator to answer “Are all or some lanes closed by incident?” as ‘T’ or ‘F’
NumRtLanes	Integer for number of right lanes affected
NumLfLanes	Number for number of left lanes affected
DistanceToIncident	Float for distance to event in feet
DistanceToDMS	Float for distance between event and DMS sign in feet
MinutesDelay	Integer for projected minutes delay caused by event
DestinationHwy	Variable-length string for highway or road where event affects motorists going to this highway. This value might appear as “destination” only in terms from tables for “Good Reason for Following the Action”
DestinationStreet	Variable-length string for cross street affected as destination by incident. This value might appear as “destination” only in terms from tables for “Good Reason for Following the Action”
DestinationCity	Variable-length string for city affected as destination by event
DestinationState	Variable-length string for state affected as destination by event
DestinationEvent	Variable-length string for event affected as destination event
DestinationStadium	Variable-length string for stadium affected as destination by event

DestinationOther	Variable-length string for any other destination affected by event
TakeHwy	Variable-length string for suggested highway to take
TakeStreet	Variable-length string for suggested cross street to take
TakeRampName	Variable-length string for suggested ramp name to take
TakeDirection	Variable-length string for suggested direction to take
MinutesSaved	Integer for projected minutes saved by taking suggested action
DMSLocation	Variable-length string for location of DMS to event, such as on same highway and near, on same highway but far, and on different highway
VPH	Integer for vehicles per hour
MaxUOI	Integer for maximum UOIs that application looks up
UOI	Integer for units of information calculated and to be subtracted from MaxUOI
fk_DMSMessage	Link to record in DMSMessage table
fk_DMSType	Link to record in DMSType table
fk_SpeedRange	Link to record in SpeedRange table
fk_Condition	Link to record in Condition table

Input_IncidentDescriptor

Attribute	Description
fk_Input	Link to record in InputData table
fk_IncidentDescriptor	Link to records in IncidentDescriptor table

IncidentDescriptor

Attribute	Description
id	Index key
Category	Integer for number of table in documentation by Finley *et al.* (*3*). Category is also the category number given in the preprocess file.
IsLargeSign	Either 'T' or 'F'
IsPortableSign	Either 'T' or 'F'
fk_Term	Link to Terms table

Terms

Attribute	Description
id	Index key

Terms	Variable-length string; content from TxDOT

Table1

Attribute	Description
id	Key
UOI	Units of information to be subtracted from maximum UOI
fk_SpeedRange	Link to SpeedRange table
fk_Condition	Link to Condition table
fk_TypeOfDMS	Link to TypeOfDMS table

Table2

Attribute	Description
id	Index key
MountingHeight	Mounting height of DMS
Offset	Offset of DMS from highway
TypeOfLED	Either ‘PERMANENT’ or ‘PORTABLE’
VCSpeed	Vertical curve design speed
fk_Condition	Link to Condition table
fk_TypeOfDMS	Link to TypeOfDMS table

Table3

Attribute	Description
id	Index key
HorizCurveRadii	Integer for curve radii in feet
OffsetSightObstruction	Integer for offset of sight obstruction from edge of travel lanes in feet
UOI	Integer for units of information to subtract from maximum UOI
Offset	Integer for offset of sign
fk_Condition	Key to record in Condition table
fk_SpeedRange	Key to record in SpeedRange table
fk_TypeOfDMS	Key to record in TypeOfDMS table

Table4

Attribute	Description
id	Index key

Visibility	String for visibility range in fog
UOI	Integer for units of information to subtract from maximum UOI
Offset	Integer for offset of sign in feet
fk_TypeOfDMS	Foreign key to a record in TypeOfDMS table
fk_SpeedRange	Foreign key to SpeedRange table

TypeOfDMS

Attribute	Description
Id	Index key
DMSIdentification	Unique identifier given to DMS in Location database
TypeOfDMS	Variable-length string with values such as LED, Fiberoptic, Reflective Disk, LED/Reflective Disk, Fiberoptic/Reflective Disk
DMSOffset	Integer for offset of DMS in feet
DMSMountingHeight	Integer for mounting height of DMS in feet
LEDType	String for type of LED, either “Permanent” or “Portable”
HasHorizontalSightObstruction	Boolean operator, either ‘T’ or ‘F’
HasVertCurveSightObstruction	Boolean operator, either ‘T’ or ‘F’
VertCurveDesignSpeed	Integer, either ‘30,’ ‘40,’ or ‘50’ in feet
VertCurvePermLEDMountHt	Integer, either ‘20’ or ‘25’ in feet
VertCurvePortLEDMountHt	Integer, either ‘7’ or ‘10’ in feet
HorizCurveRadii	Integer, either ‘250,’ ‘500,’ ‘750,’ ‘1000,’ ‘1250,’ ‘1500,’ ‘1750,’ ‘2000,’ ‘2250,’ ‘2500,’ ‘2750,’ ‘3000,’ ‘4000,’ ‘5000,’ ‘7500,’ or ‘10000’ in feet
HorizCurveOffset	Integer, either ‘2’ or ‘10’ in feet
HorizOffsetSightObstruction	Integer, either ‘10,’ ‘20,’ ‘50,’ ‘100,’ ‘150,’ ‘200,’ or ‘250’ in feet

Condition

Attribute	Description
Id	Index key
Condition	Variable-length string

SpeedRange

Attribute	Description
Id	Index key

Minimum	Integer for minimum speed of range
Maximum	Integer for maximum speed of range

DMSMessage

Attribute	Description
Key	Index key
Message	Variable-length string

APPENDIX D. MESSAGE SESSION DATA STRUCTURE

Appendix D. Message Session Data Structure

Field	Notes
Id	Primary key
Event	
LocationProximity	
IncidentHighway	
IncidentCrossStreet	
IncidentRampName	
IncidentInterconnectRampName	
HasControlArrived	
DistanceToIncident	
HighwayDMSProximity	
VehicleType	
DegreeOfCrash	
MinutesDelay	
NumDirectionalLanesAtEvent	Integer. This is the size of the LanesClosed array.
LanesClosed	Dynamic array with first cell value selected for Median and last for Shoulder. Default is NULL, meaning that median, lane, or shoulder is open.
Diversion	
NoDiversion	
SoftDiversion	
DestinationHwy	
DestinationStreet	
DestinationCity	
DestinationState	
DestinationOther	
DestinationDirection	
ReasonForFollowingAction	
MinutesSaved	
TakeHwyNonTotal	
TakeStreetNonToal	
TakeRouteNumberNonTotal	
TakeExitNonTotal	
GeneralExitTotalClosure	
HighwayTotalClosure	
RoadStreetTotalClosure	
ExitTotalClosure	

RoadWorkHwy	
RoadWorkExitRampName	
RoadWorkInterconnectingRampName	
RoadWorkDistanceDownstreamFromDMS	
RoadWorkBegins	
RoadWorkEnds	
RoadWorkNearestHwyStBegin	
RoadWorkNearestRampBegin	
RoadWorkNearestHwyStEnd	
RoadWorkNearestRampEnd	
RoadWorkDirectionTraffic	
VisibilityRange	
DMSType	
DMSLocation	
DMSOffset	
DMSMountingHeight	
LEDType	
NumLines	
NumCharPerLine	
SightDistanceToDMS	
NumDirectionalLanesAtDMSLocation	
TimeOfDay	
DayTimeWithSun	
IncidentTrafficDirection	
IsFog	
OperatingSpeedsAtDMS	
VPH	
PercentTrucks	
MaxUOI	
UOI	
EventDescriptor	Basic DMS Message. Semicolon-separated id values from Terms table
EventLocation	Basic DMS Message. Semicolon-separated id values from Terms table
LanesClosed	Basic DMS Message. Semicolon-separated id values from Terms table
EffectOnTravel	Basic DMS Message. Semicolon-separated id values from Terms table
NoDiversionAction	Basic DMS Message. Semicolon-separated id values from Terms table

ClosureLocation	Basic DMS Message. Semicolon-separated id values from Terms table
AudienceForAction	Basic DMS Message. Semicolon-separated id values from Terms table
GoodReasonForFollowingAction	Basic DMS Message. Semicolon-separated id values from Terms table
fk_DMSMessage	Foreign key to DMSMessage table
fk_DMSType	Foreign key to DMSType table
fk_SpeedRange	Foreign key to SpeedRange table
fk_Condition	Foreign key to Condition table

APPENDIX E. RECOMMENDED ASSOCIATION OF CATEGORIES OF TERMS AND GUI CONTROLS

Appendix E. Recommended Association of Categories of Terms and GUI Controls

Event Descriptor

- Event
- LEDType
- DegreeOfCrash
- VehicleType

Event Location

- LEDType
- DistanceToIncident
- LocationProximity
- IncidentHighway
- IncidentCrossStreet
- IncidentRampName
- IncidentInterconnectRampName

Lanes Closed

- LEDType
- NumDirectionalLanesAtEvent
- LanesClosed

Effect on Travel

- LEDType
- DegreeOfCrash (Determines major delay from simply a delay)
- MinutesDelay
- DestinationHwy
- DestinationCity
- DestinationState
- DestinationOther
- DestinationDirection

No Diversion Action

- LEDType
- TakeHwyNonTotal
- TakeStreetNonTotal
- TakeRouteNumberNonTotal
- TakeExitNonTotal
- IncidentTrafficDirection

Closure Location

- LEDType
- DistanceToIncident

LocationProximity
IncidentHighway
IncidentCrossStreet
IncidentRampName
IncidentInterconnectRampName

Soft Diversion
LEDType

Type 1 or 2 Diversion Action
LEDType
TakeHwyNonTotal
TakeStreetNonTotal
TakeRouteNumberNonTotal
TakeExitNonTotal
IncidentTrafficDirection
GeneralExitTotalClosure
HighwayTotalClosure
RoadStreetTotalClosure
RouteTotalClosure
ExitTotalClosure

Audience for Action
LEDType
DestinationHwy
DestinationStreet
DestinationCity
DestinationState
DestinationOther
DestinationDirection

Good Reason for Following Action
LEDType
ReasonForFollowingAction
MinutesSaved

Type 1, 2, 3, or 4 Diversion
LEDType
TakeHwyNonTotal
TakeStreetNonTotal
TakeRouteNumberNonTotal
TakeExitNonTotal
IncidentTrafficDirection
GeneralExitTotalClosure
HighwayTotalClosure
RoadStreetTotalClosure

RouteTotalClosure
ExitTotalClosure

Type 5 Diversion Action

LEDType
TakeHwyNonTotal
TakeStreetNonTotal
TakeRouteNumberNonTotal
TakeExitNonTotal
IncidentTrafficDirection

Type 6 Diversion Action

LEDType
RoadWorkHwy
RoadWorkExitRampName
RoadWorkInterconnectingRampName
RoadWorkDistanceDownstreamFromDMS
RoadWorkBegins
RoadWorkEnds
RoadWorkNearestHwyStBegin
RoadWorkNearestRampBegin
RoadWorkNearestHwyStEnd
RoadWorkNearestRampEnd
RoadWorkDirectionTraffic

APPENDIX F. MATCHING CONTROLS TO DATA STRUCTURE FIELDS

Appendix F. Matching Controls to Data Structure Fields

Field	Screen	Control
Event	1	Events
LocationProximity	3	LocationOfIncident: Radio button
IncidentHighway	3	LocationOfIncident: Highway name
IncidentCrossStreet	3	LocationOfIncident: Street name
IncidentRampName	3	LocationOfIncident: Exit Ramp name
IncidentInterconnectRampName	3	LocationOfIncident: Interconnecting ramp name
HasControlArrived	3	Have police/traffic control arrived
DistanceToIncident	3	Distance downstream from DMS
HighwayDMSProximity	3 & 6	List of three radio buttons indicating if DMS on same or different highway and near or far from incident
VehicleType	2	Vehicle type
DegreeOfCrash	2	Degree of crash
MinutesDelay	2	Estimated delay to traffic
NumDirectionalLanesAtEvent	2 & 5	Number of directional lanes at event location
LanesClosed	2 & 5	Array of lanes closed with first cell of array starting at shoulder at top and going down to other shoulder
Diversion	4 & 7	Radio button about type of diversion
NoDiversion	4 & 7	Suggested Action: No Diversion Action
SoftDiversion	4 & 7	Suggested Action: Soft Diversion Action
DestinationHwy	4 & 7	Audience for Action: Highway other than current
DestinationCity	4 & 7	Audience for Action: City
DestinationState	4 & 7	Audience for Action: State
DestinationOther	4 & 7	Audience for Action: Other locations/events

DestinationDirection	4 & 7	Audience for Action: Direction
ReasonForFollowingAction	4 & 7	Reason for Following Action: Radio button
MinutesSaved	4 & 7	Reason for Following Action: Minutes saved
TakeHwyNonTotal	4 & 7	Diversion to Specific Highway/Ramp (non-total closure): Action Terms: Highway
TakeStreetNonTotal	4 & 7	Diversion to Specific Highway/Ramp (non-total closure): Action Terms: Street
TakeRouteNumberNonTotal	4 & 7	Diversion to Specific Highway/Ramp (non-total closure): Action Terms: Route number
TakeExitNonTotal	4 & 7	Diversion to Specific Highway/Ramp (non-total closure): Action Terms: Take exit
GeneralExitTotalClosure	4 & 7	Diversion to Specific Highway/Ramp (total closure): General Exit
HighwayTotalClosure	4 & 7	Diversion to Specific Highway/Ramp (total closure): Take/Use Highway
RoadStreetTotalClosure	4 & 7	Diversion to Specific Highway/Ramp (total closure): Road/Street
RouteTotalClosure	4 & 7	Diversion to Specific Highway/Ramp (total closure): Route Number
ExitTotalClosure	4 & 7	Diversion to Specific Highway/Ramp (total closure): Exit
DMSLocation	1	DMSInformation: Location of DMS
DMSType	1	DMS Information: DMS type
LEDType	1	DMSInformation: LED type
NumLines	1	DMSInformation: Num. of lines
NumCharPerLine	1	DMSInformation: Num. of characters per line
DMSOffset	1	DMSInformation: DMS offset
DMSMountingHeight	1	DMSInformation: DMS mounting height

SightDistanceToDMS	1	DMSInformation: Sight distance to DMS
NumDirectionalLanesAtDMSLocation	1	DMSInformation: Num. of directional lanes at DMS location
TimeOfDay	1	Lighting Conditions: Time of day
DayTimeWithSun	1	Lighting Conditions: Daytime sun
IsFog	1	Fog
VisibilityRange	1	Visibility distance
OperatingSpeedsAtDMS	1	TrafficCharacteristics: Operating speeds at DMS location
IncidentTrafficDirection	1	Direction of traffic
VPH	1	TrafficCharacteristics: Vehicles per hour
PercentTrucks	1	TrafficCharacteristics: Trucks
RoadWorkHwy	6	Location of Roadwork: Highway name
RoadWorkExitRampName	6	Location of Roadwork: Exit ramp name
RoadWorkInterconnectingRampName	6	Location of Roadwork: Interconnecting ramp name
RoadWorkDistanceDownstreamFromDMS	6	Location of Roadwork: Distance downstream from DMS
RoadWorkBegins	6	RoadworkBegins: Before, past, or after radio buttons
RoadWorkEnds	6	RoadworkEnds: Before, past, or after radio buttons
RoadWorkNearestHwyStBegin	6	Roadwork Begins:
RoadWorkNearestRampBegin	6	Roadwork Begins:
RoadWorkNearestHwyStEnd	6	Roadwork Ends:
RoadWorkNearestRampEnd	6	Roadwork Ends:
RoadWorkDirectionTraffic	6	Direction of traffic

APPENDIX G. COMPATIBLE AND ORDER OF SEQUENCES OF TERMS

Appendix G. Formatting Order of Compatible Sequences of Terms

Incident Descriptor Used & Specific Lanes Closed

Order	Terms
1	Incident Descriptor
2	Incident Location
3	Lanes Closed
4	Audience for Action
5	Action
6	Good Reason for Following Action

If the term “BEST ROUTE TO” is used for Order No. 6, reverse the order of terms for Order No. 5 and Order No. 6.

Incident Descriptor Used & All Lanes Closed

Order	All Lanes Closed
1	Incident Descriptor
2	Incident Location
3	Lanes Closed
4	Audience for Action
5	Action

If the term “BEST ROUTE TO” is used for No. 6, reverse the order of terms for Order No. 5 and Order No. 6.

Incident Descriptor Used & DMS Has Two Frames

Priority	Frame	Order	Frame No. 1
1	1	1	Incident Descriptor
	1	2	Incident Location
	2	3	Lanes Closed
	2	4	Action
2	1	1	Incident Descriptor
	1	2	Incident Location
	2	3	Audience for Action
	2	4	Action

3	1	1	Incident Descriptor
	1	2	Incident Location
	1	3	Lanes Closed
	2	4	Audience for Action
	2	5	Action
4	1	1	Incident Descriptor
	1	2	Incident Location
	2	3	Audience for Action
	2	4	Action
	2	5	Good Reason for Following Action

If the term "BEST ROUTE TO" is used for No. 6, reverse the order of terms for Order No. 4 and No. 5 for Priority No. 4.

Lanes Closed Replaces or Combines with Incident Descriptor &
Specific Lanes Closed

Order	Terms
1	Lanes Closed
2	Lane Closure Location
3	Audience for Action (if needed)
4	Action
5	Good Reason for Following Action

If the term "BEST ROUTE TO" is used for Order No. 5, reverse the order of terms for Order No. 4 and Order No. 5.

Lanes Closed Replaces or Combines with Incident Descriptor &
All Lanes Closed

Order	All Lanes Closed
1	Freeway Closure
2	Closure Location
3	Audience for Action (if needed)
4	Action
5	Good Reason for Following Action

If the term "BEST ROUTE TO" is used for Order No. 5, reverse the order of terms for Order No. 4 and Order No. 5.

**Lanes Closed Replaces or Combines with Incident Descriptor &
DMS Has Two Frames**

Priority	Frame	Order	Terms
1	1	1	Lanes Closed
	1	2	Lane Closure Location
	2	3	Audience for Action
	2	4	Action
2	1	1	Lanes Closed
	1	2	Lane Closure Location
	2	3	Audience for Action
	2	4	Action
	2	5	Good Reason for Following Action
3	1	1	Freeway Closed
	1	2	Closure Location
	2	3	Audience for Action
	2	4	Action

If the term "BEST ROUTE TO" is used for No. 5, reverse the order of terms for Order No. 4 and No. 5 for Priority No. 2.

APPENDIX H. DMS MESSAGE DESIGN DATA STRUCTURE

Appendix H. DMS Message Design Data Structure

Field
EventDescriptor
EventLocation
LanesClosed
EffectOnTravel
NoDiversionAction
ClosureLocation
SoftDiversion
Type_1_2_Diversion
AudienceForAction
GoodReasonForFollowingAction
Type_1_2_3_4_Diversion
Type_5_Diversion

APPENDIX I. INFORMATION ORDER OF PRIORITY FOR INCIDENTS

Appendix I. Information Order of Priority for Incidents

Message Elements for Lane Closure Incidents	**Message Elements for Highway Closure Incidents**
Lane Closure	Highway Closure
Lane Closure Location	Location of Closure
Diversion Action	Diversion Action
Audience for Action (if needed)	Audience for Action (if needed)

Eliminate UOIs starting with lowest priority (i.e., 4).