UC Berkeley
Earlier Faculty Research

Title
Real-Time Knowledge-Based Integration Of Freeway Surveillance Data

Permalink
https://escholarship.org/uc/item/3j5655n7

Authors
Prosser, Neil A.
Ritchie, Stephen G.

Publication Date
1992-02-01
Real-Time Knowledge-Based Integration Of Freeway Surveillance Data

Neil A. Prosser
Stephen G. Ritchie

February 1992
Working Paper, No. 89
The University of California Transportation Center

The University of California Transportation Center (UCTC) is one of ten regional units mandated by Congress and established in Fall 1988 to support research, education, and training in surface transportation. The UC Center serves federal Region IX and is supported by matching grants from the U.S. Department of Transportation, the California State Department of Transportation (Caltrans), and the University.

Based on the Berkeley Campus, UCTC draws upon existing capabilities and resources of the Institutes of Transportation Studies at Berkeley, Davis, and Irvine; the Institute of Urban and Regional Development at Berkeley; the Graduate School of Architecture and Urban Planning at Los Angeles; and several academic departments at the Berkeley, Davis, Irvine, and Los Angeles campuses. Faculty and students on other University of California campuses may participate in Center activities. Researchers at other universities within the region also have opportunities to collaborate on selected studies. Currently faculty at California State University, Long Beach, and at Arizona State University, Tempe, are active participants.

UCTC's educational and research programs are focused on strategic planning for improving metropolitan accessibility, with emphasis on the special conditions in Region IX. Particular attention is directed to strategies for using transportation as an instrument of economic development, while also accommodating to the region's persistent expansion and while maintaining and enhancing the quality of life there.

The Center distributes reports on its research in working papers, monographs, and in reprints of published articles. For a list of publications in print, write to the address below.

University of California
Transportation Center
108 Naval Architecture Building
Berkeley, California 94720
Tel: 415/643-7378
FAX: 415/643-5456

Authors of papers reporting on UCTC-sponsored research are solely responsible for their content. This research was supported by the U.S. Department of Transportation and the California State Department of Transportation, neither of which assumes liability for its content or use.
Real-Time Knowledge-Based Integration Of Freeway Surveillance Data

Neil A. Prosser
Stephen G. Ritchie

Institute of Transportation Studies
Department of Civil Engineering
University of California at Irvine

Working Paper, No. 89

The University of California Transportation Center
University of California at Berkeley
REAL-TIME KNOWLEDGE-BASED INTEGRATION OF FREEWAY SURVEILLANCE DATA

Neil A. Prosser
Stephen G. Ritchie

Institute of Transportation Studies and
Department of Civil Engineering
University of California, Irvine, CA 92717

July, 1990

ABSTRACT

This paper describes an advanced processing capability that is based on the use of real-time knowledge-based expert system (KBES) technology to integrate diverse types of traffic surveillance data for freeway monitoring and control purposes, particularly as part of future "Smart Roads" projects. One of the major functions of the prototype system developed is the acquisition and processing of input data drawn from sensors and processes in the real world. The real-time nature of these processes, and the associated need for decision-making information and recommendations, places particular importance on the efficient handling of data to avoid unnecessary overloading of the expert system. The relevant types of data include traffic occupancies and volumes from loop detectors in the pavement, information on traffic conditions from closed circuit television cameras, field reports from police officers and other official personnel, and cellular and emergency telephone calls from motorists. The paper emphasizes the way in which the data are acquired, processed and integrated within a prototype KBES framework in order to achieve the objectives of the incident management tasks, and specifically those of incident detection and verification. Examples are given of the implementation of these features.
INTRODUCTION

As part of an on-going research effort investigating the application of artificial intelligence techniques for advanced traffic management, a prototype real-time knowledge-based expert system (KBES) has been developed for managing non-recurring congestion on urban freeways (1). The objective of this system, called FRED (Freeway Real-Time Expert System Demonstration), is to provide decision support in a future Traffic Operations Center (TOC) to traffic control room operators who are responsible for the monitoring and control of so-called Smart Roads, or intelligent vehicle-highway systems (2). The only comparable expert system currently known is one being developed by INRETS, the French transportation agency, for monitoring and control of arterial street intersections (6). The G2 expert system shell used in the INRETS project is the same as the one used in the FRED prototype system.

More specifically, FRED provides decision support to TOC operators in the field of incident management. Incidents are events such as accidents, spilled loads, stalled or disabled vehicles, temporary maintenance and construction activities, signal and detector malfunctions, and special events, that lead to non-recurrent congestion. Incident management involves detecting and responding to incidents in order to alleviate the resultant delay to motorists. In future Smart Roads TOC's, it will become increasingly difficult, if not impossible, for human operators to function effectively without automated assistance because of the increased amount of incoming TOC data and the complexity of both the networks and incident management and response functions (1).

The monitoring aspect of FRED involves synthesizing input data from a number of different sources. The freeway system is monitored, in the first instance, to detect when and where an incident has occurred. Once an incident has been detected more information is required about the state of the system in order to formulate appropriate responses. One of the major
functions of the system is the acquisition and processing of data drawn from sensors and processes in the real world. The real-time nature of these processes, and the associated need for decision-making information and recommendations, places particular importance on the efficient handling of data to avoid unnecessary overloading of the expert system.

Focusing on the initial incident detection and verification phases, which represent the foundation of incident management, this paper describes an advanced processing capability that is based on the use of real-time KBES technology to integrate diverse types of traffic surveillance data for freeway monitoring and control purposes. The relevant types of data include traffic occupancies and volumes from loop detectors in the pavement, information on traffic conditions from closed circuit television (CCTV) cameras, field reports from police officers and other official personnel, and in-vehicle cellular and road-side callbox telephone calls from motorists. The paper emphasizes the way in which the data are acquired, processed and integrated in order to achieve the objectives of the incident management tasks, and specifically those of incident detection and verification. Examples are given of the implementation of these features in FRED.

SYSTEM OVERVIEW

Figure 1 summarizes our view of the different stages of incident management (3). In a companion paper (1), we present and discuss the development of FRED, a component prototype real-time expert system for managing non-recurring congestion on urban freeways, and present the application of FRED to a section of the Riverside Freeway in Southern California as a case study to illustrate the current capabilities of the system. This paper emphasizes the way in which data are acquired and processed in order to achieve the objectives of the incident management tasks. The incident detection stage relies heavily on data acquisition and processing and incorporates most of the features of interest in this paper.
Figure 2 shows the overall FRED system layout (1), which in terms of incident detection and verification is consistent with the current capabilities of the Los Angeles freeway TOC operated by the California Department of Transportation, and also with many other TOC's around the country. Of particular note are the external sources of data in the detection phase. Two streams of data, representing the detection and reporting of an incident, respectively, can be recognized. First, freeway loop detectors transfer 30-second occupancy counts to an incident detection algorithm on a central computer which in turn notifies FRED of the detection of an incident. Second, a number of agencies and sources may provide on-site accounts of an incident and their reports are received by a communications center that filters such reports before sending high priority ones to FRED. Thus the two primary methods of incident detection are by an automated incident detection algorithm and by outside reports.

The incident detection algorithm (5) operates on 30-second occupancy counts (the percentage of time a loop is occupied by vehicles) obtained from sensors placed on the freeway mainline at approximately one mile spacing. Each sensor site has a series of loop detectors, one per lane, arranged to form a counting station. Local controllers are responsible for compiling the counts for the detectors in each lane and then averaging them over the counting station. In the FRED prototype system, simulated occupancy counts are read from a data file. The incident detection algorithm examines the average occupancy counts for each station and by comparing them to those for the station immediately downstream can detect the presence of a significant disruption to traffic flow between the two stations. The sensitivity of the algorithm can be controlled by varying threshold parameters. Once an incident has been detected by the algorithm, a message is sent to FRED indicating the location of the counting station immediately upstream of the detected disruption.

The FRED system has been developed using a real-time expert system shell, G2 (4), running under the Unix operating system and X-windows on a Sun SPARCstation 1 workstation. The so-called "California" incident
detection algorithms (5) and the report processing algorithm were written as separate "C" programs capable of running on a different processor. Communication between the FRED expert system and the external "C" programs is via data files.

In the freeway control environment, an incident management system must operate in "real time" if it is to detect, verify and formulate responses to incidents in a sufficiently short period of time for responses to be implemented effectively. For example, one type of response strategy is the provision of advance information to motorists approaching the site of an incident via changeable message signs or in-vehicle navigation systems. Such information, to be of benefit, must be provided soon after the occurrence of an incident. In this context "real-time" probably implies a maximum interval between incident occurrence and response implementation of the order of several minutes. The real-time operating constraint introduces some important issues related to concurrent processes and data transfer which will be discussed in the next section.

DISTRIBUTION OF PROCESSING AND DATA TRANSFER

The workload on the TOC center at any one time is directly proportional to the number of current incidents. Effective allocation of the expert system's resources is essential to maintain the speed of the overall system. The expert system should be devoted to those tasks for which it is best suited, which is the processing of high-level knowledge rather than simple algorithmic tasks. In the case of FRED, the major high-level task is the formulation of responses to incidents. The detection of further incidents should proceed concurrent and external to the expert system. When another incident is detected the expert system can be interrupted to so as to include the new incident in its set of current incidents. Currently FRED only handles one incident at a time, but the system structure was developed to facilitate the future treatment of multiple incidents.
Given that different tasks are allocated to separate and concurrent processes, the transfer of data between the processes becomes a major issue. An advantage of a knowledge-based approach is that the expert system has the ability to decide what data it requires at any particular stage and is able to send out requests for it. In this way it can select from the vast amount of data available only that which it needs for a specific task. The decision as to whether data needs to be transferred from an external routine to the expert system can also be made by the external routine itself. In this case the expert system is 'waiting' to be interrupted by the occurrence of an event.

These aspects of the operation of a real-time expert system will be discussed in terms of the implementation of FRED. Before doing so the manner in which data and knowledge are represented in the system will be outlined.

**DATA AND KNOWLEDGE REPRESENTATION IN FRED**

The FRED system was developed using a real-time expert system shell called G2 (4), which combines a knowledge-base with sophisticated data interfacing and screen management capabilities. G2, like most sophisticated expert systems, is hybrid in nature, in that knowledge and data are present separately. The manner in which the knowledge is applied to the data determines the behavior of the overall system.

**DATA REPRESENTATION**

All data in FRED are represented as a series of objects. Each object contains a number of pre-determined attributes that represent the data relevant to that object. Objects are all instances of a particular class and the class definition specifies the attributes of all objects within the class. The purpose of classes will become more apparent in the section on inference.
The attributes of objects can receive values from a number of sources. The data can be inferred from rules by the inference engine, determined by an external routine or be simulated by the G2 simulator incorporated into the G2 system. Additionally the system operator can enter values for data interactively.

The two most important object classes in the incident detection and verification phases are the counting stations and incident classes. Tables 1 and 2 show the attributes of each of these object classes along with their type and their source. Variable attributes are either symbolic or quantitative. The counting station attributes are either constant or, in the case of volumes and occupancies, receive their values from an external routine. Incident attributes are either derived from rules and formulae as part of the inference process in the expert system, or explicitly set by the operator or an outside report. Other attributes within these objects, of less importance, are not shown.

An important feature of G2 is the ability to create and delete objects during the running of the system. Such transient objects are particularly suitable for incident objects. When an incident is detected, an instance of the incident object class is created and its attributes set to the parameters of that incident. Once the incident has terminated the object can be deleted. Thus at any one time the number of incident objects equals the number of current incidents. Counting stations, on the other hand, are examples of permanent objects as they represent a fixed element of the freeway system.

KNOWLEDGE REPRESENTATION

Knowledge is encoded into the FRED system as a set of rules forming a knowledge base. These rules consist of antecedent and consequent parts. The antecedents specify conditions that must be satisfied before actions in the consequents can be executed. When this occurs the rule is said to have 'fired'. Essentially, it is the execution of the actions within the consequents of the rules that drives the system.
The most important action command is the conclusion action as it drives the inference process. Conclusions about the state of the system are drawn, or inferred, from the conditions in the antecedent of the rule, and linking conclusions to antecedents achieves a chaining of individual rules via the inference process. The state of the system, as referred to here, is embodied solely within the objects and their attributes. Rules make reference to the attributes of objects in a number of ways. They can refer to the attribute of a particular instance of an object - for example, the following rule refers to the occupancy attribute of the object WB5 which is an instance of the counting station class:

\[
\text{if the Occupancy of WB5} > 30
\]

\[
\text{then} \quad .......
\]

It is more likely however that rules will be generic and will refer to object classes rather than instances of that class. The following rule applies to any instance of the counting station class, and is the rule that initiates the incident management process. Incident\_status refers to the attribute of the counting station class that contains a flag indicating whether an incident has been detected or not.

\[
\text{if the Incident\_status of any count\_station} = 1
\]

\[
\text{then activate the subworkspace of inc-detection}
\]

Generic rules such as this one are well-suited toward the encoding of knowledge.

The chaining of rules can be done in two ways - forwards and backwards. Forward chaining attempts to match facts about the current state of the system to conditions embodied in the antecedents of all rules in the knowledge-base. Rules that are fired may then provide further information,
via conclusions, about the state of the system that can fire more rules and so on.

Backward chaining operates in reverse. The system attempts to infer a hypothesis about the state of the system by firing only the rules that are relevant to that hypothesis. An attempt to fire the relevant rules is made by examining the conditions contained in the antecedent. To satisfy these conditions more rules may become relevant.

Thus forward chaining is a data-driven process, whereas backward chaining is a goal-driven process. FRED is primarily data-driven but uses backward chaining where necessary. For example, the major event that drives the incident management process is the detection of an incident. To respond to a particular incident rules may require further information and backward chaining will occur to furnish this information.

DATA ACQUISITION AND INTEGRATION IN FRED

This section examines the way in which data are transferred from external routines and sensors to the FRED expert system. In the current implementation of FRED there are no sensors in the physical sense but rather a series of external routines that simulate these sensors. Separate programs model the provision of counting station data to the incident detection algorithm and the processing of outside reports. Even in real applications, the expert system would receive data from controller routines that would operate in a manner similar to that used in FRED.

Figure 3 shows the system components from the data interfacing perspective. FRED, and the external routines interfacing with it, are all processes running concurrently and sharing the resources of the Central Processing Unit (CPU). Communication between these processes is via data files. The external routines write any data that needs to be passed to FRED to
the data files which are accessed by a separate G2 Standard Interface (GSI). Data read from these files is passed to objects within FRED which have been designated as having external data sources. Application specific bridge routines must be written within the GSI module to perform the interrogation of data files and returning of data to FRED.

The G2 expert system and the GSI module communicate via object indexes. Each object that has an external routine as its data source is assigned a unique object index by GSI when the system is started. In addition to the system-assigned index, each interface object has attributes, specified by the system developer, that can be used to identify the type of data. In FRED each variable with an external source has a "data type" attribute that is used to determine from which external routine the data originates. For example, the volume variable for each counting station has the same value for the "data type" attribute. In this way the interface bridge routine knows when to access the data file containing the current volume values.

Data can be transferred from the external routines to FRED in two ways - as solicited or unsolicited input.

SOLICITED INPUT

As part of the process of backward chaining, the inference engine may require the value of an attribute that has a data source external to the expert system. From the expert system's perspective the external entity is "polled" for the current value of that variable, and once it returns the value processing continues. At a lower level, the inference engine passes a request for data to GSI specifying the index of the object for which a value is sought and passing the value of the data type attribute. The "get data" bridge routine determines which data file to interrogate, opens it and obtains the required value.

For example, when formulating responses to incidents, 30-second volume counts are often required from the freeway mainline and ramp counting stations. These values are written to a data file every 30 seconds by a
separate simulation routine. If volume counts are required, a request will be sent to GSI and the "get data" bridge routine will be called passing the object index and the data type attribute. The bridge routine will recognize the data type attribute as being the one for volume counts and will open the file containing the current volume counts, select the appropriate value and return them to FRED.

To ensure an efficient interface, requests for data are usually grouped, so that the interface bridge routine is passed an array of object indexes and data type attributes representing requests for a series of objects. In our preceding example, requests for volumes from a series of counting stations will be grouped and sent to GSI at the same time. This ensures that the volume data file will be opened only once.

UNSOLICITED INPUT

Sensors or programs external to the expert system can send input without being directly requested by G2. In this case the external routine is making the decision to transfer data, not the expert system. Such data transfers can act as interrupts to the current operation of the system. An "accept data" bridge routine is called by the interface program every second and this routine interrogates data files that may contain input from external routines. If such data is found it is returned along with the appropriate object index.

In FRED, the detection of an incident by the incident detection algorithm leads to unsolicited data transfer from the algorithm to FRED via the GSI interface. The Incident_status attribute of each counting station (see Table 1) has an external source, namely the incident detection algorithm. When this algorithm detects an incident the identification number of the counting station immediately upstream of the incident is written to a data file. This file is examined by the "accept data" bridge routine in GSI every second. If new data are found, the bridge routine reads the identification
number of the upstream counting station from the file and returns 1 as the value for the *Incident_status* attribute of that counting station.

**PASSING DATA FROM FRED TO EXTERNAL ROUTINES**

The behavior of the external routines can be controlled by FRED. Object attributes with external sources can be "set" by rules in the knowledge base. This is the reverse of data retrieval in that now GSI is passed a value along with an object index. The "set data" bridge routine can then use that value to set some aspect of an external routine. For example, the calibration of the incident detection algorithm can be altered by setting the algorithm number and threshold parameters. Updated values for these parameters are sent to the "set data" bridge routine and then written to a data file by the bridge routine. The incident detection algorithm interrogates this data file at regular intervals, and if new values are found, they are used to recalibrate the model.

**VALIDITY OF DATA**

As mentioned earlier, data representing the state of the external system is continually changing and the inference engine must be able to change its conclusions based on new data. This process is termed non-monotonic reasoning. Conclusions inferred at one stage of the system may become invalid later. Thus data and conclusions have validity intervals associated with them. For example, the counting station volumes are valid for only 30 seconds. Any conclusions inferred from these variables will also be valid for only 30 seconds.

Variables with external data sources will have prescribed validity intervals. Variables whose values are inferred from rules will have their validity intervals supplied by the inference engine as the minimum validity interval of the facts used in the inference. For example, in the following rule if facts A and B expire in 5 and 10 seconds respectively then fact C will have an expiration time 5 seconds from the time at which the rule fired.
if \( A \) is true and \( B \) is true
then conclude that \( C \) is true

SYSTEM ROBUSTNESS

Any system that interfaces with a large number of external processes must be tolerant of malfunctions and breakdowns. A feature built into the inference engine, causes a failed data request to be retried after a specified interval, typically 5 seconds. If such attempts repeatedly fail the variable causing the request is said to have "timed out" and a special rule handles such a situation. For example, the following rule fires when the volume attribute of any counting station times out:

\[
\text{whenever the volume of any count}\_\text{station c1 fails to receive a value}
\text{then inform the operator that "Request for volume value failed"}
\]

Additionally, rules can be written to check for erroneous data, such as negative or excessively high volume counts.

In G2 and FRED the interface program can send error messages and statuses back to the expert system in the case of erroneous communication with an external routine. A separate body of rules within the expert system can be invoked to handle such situations.

SELECTIVE KNOWLEDGE PROCESSING

To maintain a "real-time" expert system consideration must be given to speed of rule processing. In FRED this leads to the need for control over the incident management tasks, particularly when considering multiple incidents. The tasks associated with managing each incident must be prioritized to efficiently use the resources of the inference engine and the TOC operator.
The FRED system was built in such a manner that the rules associated with each incident management task could be invoked when required. Responsibility for invoking these sub knowledge-bases resides with a set of management rules. Rules in FRED are activated and deactivated during the running of the system, thus reducing the load on the inference engine at any one time. For example, none of the incident management rules are active when no incidents have been detected. Once an incident is detected the incident detection rules are activated followed by the incident verification rules and so on.

Currently FRED only works on a single incident at a time, but later expansion will incorporate multiple incidents, with some method of prioritizing the incident management tasks for several incidents. The formulation of incident responses is significantly complicated when considering multiple incidents on the same section of freeway.

SYSTEM PERFORMANCE

The case study corridor used for developing the FRED prototype system (1) is relatively small in terms of the amount of surveillance data received by the FRED system. Future research should involve on-line testing in a larger freeway environment to evaluate the actual real-time performance of the system. However, we are confident that the FRED system approach will meet the processing requirements of incident management. The G2 shell used in the FRED system has recently been successfully employed in a number of industrial and aerospace applications, including the U.S. Space Shuttle, with much more demanding performance criteria. The nature of freeway traffic control is such that the typical response times are of the order of several minutes rather than seconds. The most likely performance limitation in FRED will be its ability to handle several simultaneous incidents. On-going research is addressing this necessary capability of the FRED system.
DATA INTEGRATION IN INCIDENT DETECTION AND VERIFICATION

To illustrate the manner in which data arriving from different sources are integrated within FRED, the basic tasks in incident detection and verification will be considered.

To simplify the example we assume that there are two ways to establish the existence of an incident - automated incident detection with verification by operator controlled CCTV, and on-site incident reports.

Method 1 - Automated Incident Detection and CCTV Verification

The initial event in this process is the detection of an incident by the incident detection algorithm. The detection message is sent as unsolicited input to FRED by setting the Incident_status attribute of the counting station that triggered the detection to 1. This counting station, call it C, is always the counting station immediately upstream of the incident site. A rule is fired whenever the Incident_status attribute of any counting station is set to 1 and leads to the creation of an incident object. Referring to Table 1, the following attributes of the new incident are set at this stage:

- **Status** - set to the symbol "possible" at this stage
- **Upstream_station** - set to C - the name of the counting station that triggered the original detection.
- **Downstream_station** - set to the name of the counting station object immediately downstream of the incident. Obtained from the value of the Downstream_station attribute of C.
Milepost - the approximate milepost position of the incident set at this stage to the average of the Milepost constants of the upstream and downstream counting stations.

Direction - the direction of the carriageway on which the incident was detected, for example, eastbound or westbound, copied from the Direction attribute of the upstream counting station C.

From the operator's perspective the next step is to verify the incident, in our simplified example by consulting a CCTV camera. From visual inspection of the incident site the operator is required to enter information for the following incident attributes:

Type - a selection from a prescribed list of incident types such as overturned truck, 3-car accident, stalled vehicle etc.

Duration - an estimate of the duration of the incident in hours

Lanes_blocked - the number of lanes of the carriageway that are blocked by the incident.

Side_blocked - the side of the carriageway, left or right, blocked by the incident.

No_injuries - the number of confirmed injuries resulting from the incident.

No_fatalities - the number of confirmed fatalities resulting from the incident.

In the case of an overturned truck, or spilled load, the following attributes need to be set as well:
Load\_description - a text string of free format describing the spilled load eg. diesel fuel.

Load\_weight - an estimate of the weight of the spilled load in tons.

Additionally, the operator is able, at the incident confirmation stage, to more precisely specify the location of the incident. This is done in FRED by moving a screen marker on a freeway schematic, drawn approximately to scale, to the position of the incident as seen from the CCTV camera. Once the location is confirmed by the operator the Milepost attribute of the incident is updated using the coordinates of the incident marker. Finally, the Status attribute of the incident is set to "confirmed" to denote a confirmed incident.

Certain attributes of an incident have their values inferred from other attribute values via formulae and rules:

Total\_lanes - the total number of lanes at the incident site - set by examining a table that lists the number of lanes for different milepost locations.

Capacity - the unrestricted capacity of the freeway at the incident site - again obtained from a table ordered on milepost values.

Method 2 - Outside Reports

The incident attributes outlined above are set in a different manner when an incident is detected only from outside reports.

A major part of the acquisition of data from outside reports is performed by the report processing program external to FRED, which allows a communications center operator to enter an outside report. The user is prompted for all the necessary details. The information is then transferred to FRED as unsolicited input and, as in the previous case, an incident object is
created. The following incident attributes, described above, are derived from the report:

- **Milepost**
- **Direction**
- **Type**
- **Duration**
- **Lanes_blocked**
- **Side_blocked**
- **No_injuries**
- **No_fatalities**
- **Load_description**
- **Load_weight**

The **Milepost** attribute is used to infer the names of the upstream and downstream counting stations. The remaining attributes are inferred as in the previous case.

Once an incident has been confirmed, the data contained within the attributes are used to formulate responses. For example, information describing the nature of the incident is used to compute the expected capacity of the freeway at the incident site and FRED then recommends whether to close entrance ramps upstream of the incident.

**CONCLUSIONS**

The operating requirements of a real-time expert system to aid traffic control operators in incident management place particular emphasis on the efficient acquisition and integration of data from a number of external sensors. The manner in which the FRED system achieves this objective was outlined, with particular reference to the data interfacing between several concurrent processes in the incident detection and verification phases of incident management. The system structure developed serves as a good foundation for the development of a sophisticated and robust real-time...
system for management of multiple incidents that commonly occur on large freeway systems. In this respect, the refinement, expansion and evaluation of the FRED system is on-going.

ACKNOWLEDGEMENTS

This research was supported by a U.S. DOT/University of California Transportation Center Award for "Real-Time Decision Support for Freeway Surveillance and Control," and by the National Science Foundation under Grant No. MSM-8657501.

REFERENCES


LIST OF FIGURES

Figure 1 Incident Management Tasks
Figure 2 FRED External System Overview
Figure 3 FRED Data Interface Structure
Figure 1 Incident Management Tasks

1. Incident Detection
2. Incident Verification
3. Identification and Evaluation of Alternative Responses
4. Implementation of Selected Response(s)
5. Monitoring Recovery
Figure 3 FRED Data Interface Structure
LIST OF TABLES

Table 1 Incident Object Attributes
Table 2 Counting Station Object Attributes
**Table 1** Incident Object Attributes

<table>
<thead>
<tr>
<th>ATTRIBUTE NAME</th>
<th>TYPE</th>
<th>SOURCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Symbol</td>
<td>Inference Process</td>
</tr>
<tr>
<td>Upstream_station</td>
<td>Symbol</td>
<td>Inference Process</td>
</tr>
<tr>
<td>Downstream_station</td>
<td>Symbol</td>
<td>Inference Process</td>
</tr>
<tr>
<td>Milepost</td>
<td>Quantity</td>
<td>Operator/Report</td>
</tr>
<tr>
<td>Direction</td>
<td>Symbol</td>
<td>Inference Process</td>
</tr>
<tr>
<td>Type</td>
<td>Symbol</td>
<td>Operator/Report</td>
</tr>
<tr>
<td>Duration</td>
<td>Quantity</td>
<td>Operator/Report</td>
</tr>
<tr>
<td>Lanes_blocked</td>
<td>Quantity</td>
<td>Operator/Report</td>
</tr>
<tr>
<td>Side_blocked</td>
<td>Symbol</td>
<td>Operator/Report</td>
</tr>
<tr>
<td>No_injuries</td>
<td>Quantity</td>
<td>Operator/Report</td>
</tr>
<tr>
<td>No_fatalities</td>
<td>Quantity</td>
<td>Operator/Report</td>
</tr>
<tr>
<td>Load_description</td>
<td>Symbol</td>
<td>Operator/Report</td>
</tr>
<tr>
<td>Load_weight</td>
<td>Quantity</td>
<td>Operator/Report</td>
</tr>
<tr>
<td>Total_lanes</td>
<td>Quantity</td>
<td>Inference Process</td>
</tr>
<tr>
<td>Capacity</td>
<td>Quantity</td>
<td>Inference Process</td>
</tr>
</tbody>
</table>
Table 2 Counting Station Object Attributes

<table>
<thead>
<tr>
<th>ATTRIBUTE NAME</th>
<th>TYPE</th>
<th>SOURCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident_status</td>
<td>Quantity</td>
<td>External</td>
</tr>
<tr>
<td>Occupancy</td>
<td>Quantity</td>
<td>External</td>
</tr>
<tr>
<td>Volume</td>
<td>Quantity</td>
<td>External</td>
</tr>
<tr>
<td>Downstream_station</td>
<td>Symbol</td>
<td>Constant</td>
</tr>
<tr>
<td>Milepost</td>
<td>Quantity</td>
<td>Constant</td>
</tr>
<tr>
<td>Direction</td>
<td>Symbol</td>
<td>Constant</td>
</tr>
</tbody>
</table>