

UNIVERSITY OF MINNESOTA



CENTER FOR TRANSPORTATION STUDIES

**INTELLIGENT
TRANSPORTATION
SYSTEMS
INSTITUTE**

Future TMC Operations System Prototype and Testing Facility: Phase II

Final Report

Prepared by
Panos Michalopoulos
John Hourdakis
Prasoon Sinha

Department of Civil Engineering
University of Minnesota

CTS 04-06

Technical Report Documentation Page

1. Report No. CTS 04-06	2.	3. Recipients Accession No.	
4. Title and Subtitle Future TMC Operations System Prototype and Testing Facility: Phase II		5. Report Date March 2005	
		6.	
7. Author(s) Panos Michalopoulos, John Hourdakis, Prasoon Sinha		8. Performing Organization Report No.	
9. Performing Organization Name and Address University of Minnesota Department of Civil Engineering 500 Pillsbury Drive S.E. Minneapolis, MN 55455-0116		10. Project/Task/Work Unit No.	
		11. Contract (C) or Grant (G) No.	
12. Sponsoring Organization Name and Address Intelligent Transportation Systems Institute Center for Transportation Studies University of Minnesota 511 Washington Avenue, SE Suite 200 Minneapolis, MN 55455		13. Type of Report and Period Covered Final Report	
		14. Sponsoring Agency Code	
15. Supplementary Notes http://www.cts.umn.edu/pdf/CTS-04-06.pdf			
16. Abstract (Limit: 200 words) <p>This report looks at how traffic control rooms are behind the curve of available technology. Mn/DOT realized a need to create TRAMLAB, in order to set up an environment which contains a virtual control room where technologies such as; surveillance, traffic control, incident detection, and management can be tested before implementation.</p> <p>The three objectives for this study are as follows:</p> <ol style="list-style-type: none"> 1. Evaluate existing operations in the control room of the Mn/DOT TMC and make recommendations for increasing the efficiency of their operations. 2. Enhance TRAMLAB by enhancing the user interface and integrating it with a microscopic simulator. 3. Enhance the utilities of the simulator by adding features such as an automated initial state creator and a module for presenting the output of simulation. <p>In spite of the completion of the TRAMLAB objectives, there is still room for further expansion and improvements. One example includes incorporation of algorithms for detecting the malfunctioning of field devices such as detectors. Another improvement would be to expand TRAMLAB for urban street actuated and adaptive control (as opposed to pre-timed only) incorporating actuated control interfaces as well as SCOOT and SCATS or other traffic -responsive schemes. Further, a web interface can be developed to provide the non-university users with access to TRAMLAB</p>			
17. Document Analysis/Descriptors Traffic Control Room Simulator Environment		18. Availability Statement No restrictions. Document available from: National Technical Information Services, Springfield, Virginia 22161	
19. Security Class (this report) Unclassified	20. Security Class (this page) Unclassified	21. No. of Pages 108	22. Price

Future TMC Operations System Prototype and Testing Facility: Phase II

Final Report

Prepared by:

Panos Michalopoulos

John Hourdakis

Prasoon Sinha

Department of Civil Engineering
University of Minnesota

March 2005

Intelligent Transportation Systems Institute
University of Minnesota

CTS 04-06

TABLE OF CONTENTS

Chapter 1. Introduction	1
1.1 Problem Statement	1
TRAMLAB Overview	1
1.2 Objectives.....	2
1.2.1 Study of Mn/DOT TMC Traffic Control Room Operations.....	2
1.2.2 Enhancements to TRAMLAB.....	2
1.2.3 Enhancements to simulator	3
Chapter 2. Study of Mn/DOT TMC Traffic control room operations	4
2.1 Introduction.....	4
2.2 Methodology	5
2.2.1 Observations	5
2.2.1.1 Layout of the Traffic Control Room.....	5
2.2.1.2 Information.....	6
2.2.2 Questionnaires.....	8
2.2.2.1 Information expected from the questionnaire	9
2.3 Observations AND Questionnaires: A Discussion	12
2.3.1 Environment and Devices in the Control Room.....	12
2.3.2 Information: More data and representation.....	14
2.3.3 Traffic Control	15
2.3.4 Automated indications	16
2.3.5 Incidents Detection and Management.....	16
2.3.6 Equipment Maintenance	17
2.3.7 Integration of functions	18
2.4 Conclusion	18
Chapter 3. TRAMLAB Design	19
3.1 Virtual TMC Features	19
3.2 Design Issues and Solutions	19
3.3 Architecture.....	19
3.3.1 Development Environment Selection: An expert system, G2	22
3.3.2 Simulator Selection: AIMSUN2	22
3.3.3 Data Availability and Representation.....	23
3.3.3.1 Detector Station.....	23
3.3.3.2 Detector	23
3.3.3.3 Operating Section.....	24
3.3.3.4 Ramp Meter.....	26
3.3.3.5 Camera	26
3.3.3.6 Variable Message Sign.....	27
3.3.3.7 Metering Zone.....	28
3.3.3.8 Incident.....	29
3.3.4 The Laboratory Environment.....	29
3.3.4.1 The Physical Environment	30

3.3.4.2	The Computing Environment	30
<u>3.3.4.2.1</u>	Graphical User Interface	31
3.3.4.3	Knowledge Representation.....	31
3.3.4.2	Data Interface	33
3.3.4.4	Data structures in G2 Gateway for data storage	34
3.3.4.5	G2 Standard Interface (GSI) description	35
3.3.5	Module Description.....	35
3.3.5.1	Traffic monitoring.....	35
3.3.5.2	Traffic control.....	37
3.3.5.3	Incident detection.....	38
3.3.5.4	Incident verification and Response	39
Chapter 4.	Graphical User Interface	41
4.1	Traffic Display Map.....	41
4.2	Section Map	43
4.3	Detector Station Information Dialog.....	44
4.4	Detector information Dialog.....	46
4.5	RAMP METER INFORMATION DIALOG	48
4.7	CAMERA INFORMATION DIALOG	49
4.8	Variable Message Sign Dialog.....	50
4.9	Metering Zone Information Dialog.....	51
4.10	Incident Information Dialog.....	53
4.11	REPORTING OF INCIDENTS	57
4.12	ACTIVE INCIDENTS	58
4.13	USER MODES	59
Chapter 5.	TRAMLAB Implementation.....	62
5.1	Strategy Evaluation Procedure.....	62
5.2	Test Site and Data for Strategy Evaluation Example	64
5.3	Strategy Evaluation Scenario	65
Scenario Execution		66
5.4	Results	68
5.4.1	Travel Time.....	68
5.4.2	Speed.....	70
5.4.3	Delay.....	70
<u> 5.5</u>	OPERATOR TRAINING	70
Chapter 6.	Concluding Remarks	74
REFERENCES		75
<u> Bibliography.....</u>		76
Appendix-A:	RULES AND PROCEDURES (CODE)	A-1

LIST OF FIGURES

Figure 2.1: Plan of the Traffic Control Room.....	6
Figure 2.2: TMC Control Room	7
Figure 3.1: TRAMLAB Architecture.....	21
Figure 3.2: Table and Dialog of Detector Station.....	24
Figure 3.3: Table and Dialog of Detectors.....	25
Figure 3.4: Table and Dialog of Operating Sections.....	25
Figure 3.5: Table and Dialog of Ramp Meter.....	26
Figure 3.6: Table and Dialog of Camera.....	27
Figure 3.7: Table and Dialog of Variable Message Sign.....	28
Figure 3.8: Table and Dialog of Zone.....	28
Figure 3.9: Table and Dialog of Incident.....	29
Figure 3.10: TRAMLAB Physical Environment.....	30
Figure 3.11: TRAMLAB- Simulator Interface.....	34
Figure 3.12: Traffic Monitoring.....	36
Figure 3.13: Traffic Monitoring through graphs.....	36
Figure 3.14: Traffic Control.....	38
Figure 3.15: Incident Detection.....	39
Figure 3.16a: Incident Verification.....	40
Figure 3.16b: Incident Verification.....	40
Figure 4.1: Traffic Display Map.....	42
Figure 4.2: Section Map.....	43
Figure 4.3: Detector Station Information.....	45
Figure 4.4: Detector Information Dialog.....	46
Figure 4.5: Graphs of Detectors.....	47
Figure 4.6: Ramp Meter Information Dialog.....	48
Figure 4.7: Camera Information Dialog.....	50
Figure 4.8: Variable Message Sign Dialog.....	51
Figure 4.9: Metering Zone Information Dialog.....	52
Figure 4.10: Zone and Ramp Dialog.....	53
Figure 4.11: Incident Notification Workspace.....	54
Figure 4.12: Incident Form.....	55
Figure 4.13: Incident Information Dialog.....	56
Figure 4.14: Report Incident and Active Incidents.....	57
Figure 4.15: Remove Incidents.....	58
Figure 4.16: Menu choice in Administrator Mode.....	59
Figure 4.17: Workspace showing ways to get dialog box	61
Figure 4.18: View Dialog Box.....	61
Figure 5.1: Flow Chart for Traffic Management Strategy Evaluation.....	63
Figure 5.2: I-494 Transportation Corridor (ICTM)	64
Figure 5.3: Location of various field devices in the ICTM Corridor	64
Figure 5.4: Incident Location and Traffic Diversion	67
Figure 5.5: Effects of Strategy2	68
Figure 5.5: Training Site (I-35W)	71

LIST OF TABLES

Table 5.1: Interstate 494.....	65
Table 5.2: MOE Comparisons.....	69
Table 5.3: Training Site Characteristics.....	72

Chapter 1. Introduction

1.1 PROBLEM STATEMENT

Existing traffic control room operations are often not taking full advantage of recent advances in Intelligent Transportation systems (ITS) technology. Although various features have been added over the years, they have not been integrated and incorporated into the Traffic Control Room in a fashion consistent with the ITS architecture. Furthermore, the strategies adopted regarding surveillance, incident detection and management, ramp control, etc., are not necessarily using state-of-the-art in technology and may not be the best for a particular Traffic Management Center (TMC). New facilities on the other hand, incorporate technology and adapting to the National ITS Architecture but in terms of control room operations, they have not deviated from current practice.

Realizing that it is timely to integrate the best possible combination of the technology currently available, The Minnesota Department of Transportation (Mn/DOT's) TMC had requested the researchers at the University of Minnesota to evaluate and possibly improve their existing Traffic Control Room operations. This requires a tool for offline testing of desired alternatives, in user specified and controllable fashion, that minimizes time and effort. An offline analysis also reduces the risk of failure involved in the deployment of policies. Furthermore, it provides reasonable confidence that the selected improvements work efficiently when they are incorporated into the Traffic Control Room operations.

TRAMLAB Overview

Realizing this need Mn/DOT sought development of a laboratory for improving their Traffic Control Room operations. Such improvements include integration of various operations such as surveillance, incident detection and incident management, ramp control, and equipment maintenance into a single system rather than individual components. Since, for each of these components, a number of technological alternatives are available, it is desirable to test and evaluate each option and/or a particular combination of technologies prior to implementation.

In order to facilitate efficient development and evaluation of new concepts and schemes for Traffic Management and Control Room operations as well as to evaluate the effectiveness of desirable combinations of technologies, the concept of a Traffic Management Laboratory (TRAMLAB) was proposed. The basic idea behind TRAMLAB is to emulate the TMC Traffic Control Room and evaluate existing or new concepts of its operations, in a *virtual environment* prior to actual implementation. TRAMLAB will effectively combine the power of simulation with the facility to easily gather, access, and manipulate data through a graphical user interface. It will also combine various functions such as surveillance, traffic control, incident detection, and management into one integrated system.

The initial phase of development of TRAMLAB involved the integration of surveillance as well as incident detection and management techniques. The current status of TRAMLAB development includes a fully operational control module, integration of simulation capabilities, and the support of improved user interfaces.

1.2 OBJECTIVES

The objectives of this study are to:

1. Evaluate existing operations in the control room of the Mn/DOT TMC and make recommendations for increasing the efficiency of their operations.
2. Enhance TRAMLAB by enhancing the user interface and integrating it with a microscopic simulator.
3. Enhance the utilities of the simulator by adding features such as an automated initial state creator and a module for presenting the output of simulation.

1.2.1 Study of Mn/DOT TMC Traffic Control Room Operations

The first objective was to study Mn/DOT's existing control room operations in order to understand the intricacies of its functions. The study included manual observation as well as feedback from the staff members. This facilitated a cursory analysis of the control room operations, which was the basis for discussion and critique. An attempt was made to indicate the areas for improvement leading to an increase in efficiency. Furthermore, suggestions regarding the possible introduction of new concepts into the operational functions were made.

1.2.2 Enhancements to TRAMLAB

Enhancement of user interface

The initial phase of TRAMLAB saw the development of functions related to incident detection and surveillance. However these functions had to be enhanced with a better user interface. Based on the results of the study of the control room operations, enhancements to the user interface were made. Specifically the following enhancements were made: flexible control strategies, communication with the microscopic traffic simulator, and operating control devices such as ramp meters and VMS through the TRAMLAB user interface.

Integration with microscopic simulation

The existing TRAMLAB system was integrated with a microscopic simulator AIMSUN so that the simulator would serve as a data source for the devices in TRAMLAB. The real life conditions are better represented with a calibrated simulation model.

1.2.3 Enhancements to simulator

Initial conditions for simulation

In many situations, it is desirable to start the simulation of traffic in a freeway network with conditions existing in the field at a given point of time. This is a better representation than an empty network state. The information required by the simulator, i.e., flow, speed and density (for each section) is calculated by using the loop detector information at that time.

Presentation of Simulation results

The presentation of the output of simulation in a structured manner enables the user to retrieve information based on the applications. It prevents the user from being overloaded with an enormous amount of data, especially in the case of microscopic simulators. The development of a preliminary version of a visualization module that would enable the operations mentioned above was undertaken.

The implementation of the above features would enhance the existing TRAMLAB. TRAMLAB is expected to create a virtual environment enabling the evaluation of existing as well as new concepts related to traffic control room operations.

Chapter 2. Study of Mn/DOT TMC Traffic control room operations

2.1 INTRODUCTION

The Traffic Management Center (TMC) strategically manages traffic in order to ease congestion. TMC uses data from existing information systems to determine traffic circulation management strategies. TMC strives to reduce travel time and frustration for millions of people living, working, or visiting. The goal of the TMC is to aggressively manage the transportation system to minimize congestion and provide the safe and efficient movement of people, goods, services, and information.

It is in this light that there is a need to evaluate the existing operations of the traffic control room that is part of the TMC. It is highly important that the functionality of the control room is at its maximum efficiency in order to achieve the goals that the center targets. This implies that the distribution of functions among the operators has to be optimized and tools that help the operators in their endeavors should be improved.

The primary function of the Mn/DOT TMC is to promote a smooth and delay-free flow of traffic on the freeway system of the metropolitan Minneapolis-St. Paul area. The TMC controls traffic using adaptive ramp metering. It monitors traffic in order to identify areas of congestion and possible incidents. Units in the field, and when necessary, emergency response providers are alerted to such incidents. The Motorist Information System provides the drivers with information about potential congestion and delays.

The heavy reliance on Closed Circuit TVs (CCTVs) and the switching of images between monitors requires significant operator monitoring in support of incident management. Given that the monitoring of the large number of CCTVs well exceeds human factors recommendations, the use of these monitors by operators represents an important human factors issue and deserves additional examination. Methods for data fusion and automatic camera selection need to be examined. Automation enhancements and decision support should be considered for the operators.

Furthermore, the current traffic control room operations are often not taking full advantage of recent advances in the ITS technology. Although various functions have been added over the years, they have not been integrated and incorporated into the TMC in a fashion consistent with the national ITS architecture, which provides a common framework for planning, defining, and integrating intelligent transportation systems. It reflects the contributions of various members of the ITS community like practitioners, engineers, etc. It also defines the functions making up the building of a management center, the physical entities or the devices that make up the framework, and also the information flow between all the components.

The strategies adopted by the TMC regarding various functions like surveillance, incident detection and management, etc., are not necessarily using state-of-the-art technology. The objective of this research was to study the existing practices in the TMC and to critique the operations in order to:

- Understand the human factors perspective in the operations;
- Explore means to update to latest technology;
- Explore means to make operations easy.

The study also aimed at understanding the intricacies of various TMC functions. The process included manual observation as well as feedback from the staff members. This facilitated a cursory analysis of the control room operations, which was the basis for discussion and critique. An attempt was made to point to the areas where there could be improvement so that the overall efficiency of the operations could be improved. Furthermore, suggestions regarding the possible introduction of new concepts into the operational functions were made.

2.2 METHODOLOGY

The following methodology was adopted for studying the TMC operations.

2.2.1 Observations

First, the operations in the control room were observed for three days during the peak period of traffic on the freeway system. This provided an opportunity to witness the peak period operating procedures and to know the intricacies in traffic management. It helped in gaining a better understanding of the facilities, operating procedures, and the responsibilities of the staff.

Through this study, it was intended to get to know the general structure, facilities of the Control room and to establish rapport with the staff working in the room. Information regarding the day-to-day operations involved in highway traffic management was also gathered.

2.2.1.1 Layout of the Traffic Control Room

The layout of the control room is as shown in the Figure 2.1. It consists of stations for North side and South side operators, two information officers, and one traffic radio announcer.

The Twin Cities' freeway network is divided into North side and South side sections. The operators view the freeways in their sections through live video coming in from the field. The video is seen on 19" CCTV monitors. The operators are also equipped with a personal monitor, a computer showing the current ramp meter status in all the zones, and a computer through which they can change the ramp metering rates and send in VMS messages. The operators use the personal monitor to zoom into a particular stretch of the highway. This could be done through the operation of the field PTZ cameras through the video-switching keyboard.

The information officers are also each equipped with personal monitors to have a view of the freeway sections. Information Officer-1 (IO-1) is equipped with a personal computer through which he/she can communicate with the state patrol through text messages. There is also a computer showing the location of the highway helpers with the help of a GPS system. IO-1 also constantly listens to the radio communication from the state patrol and, if necessary, contacts them in order to get more information regarding a situation in hand.

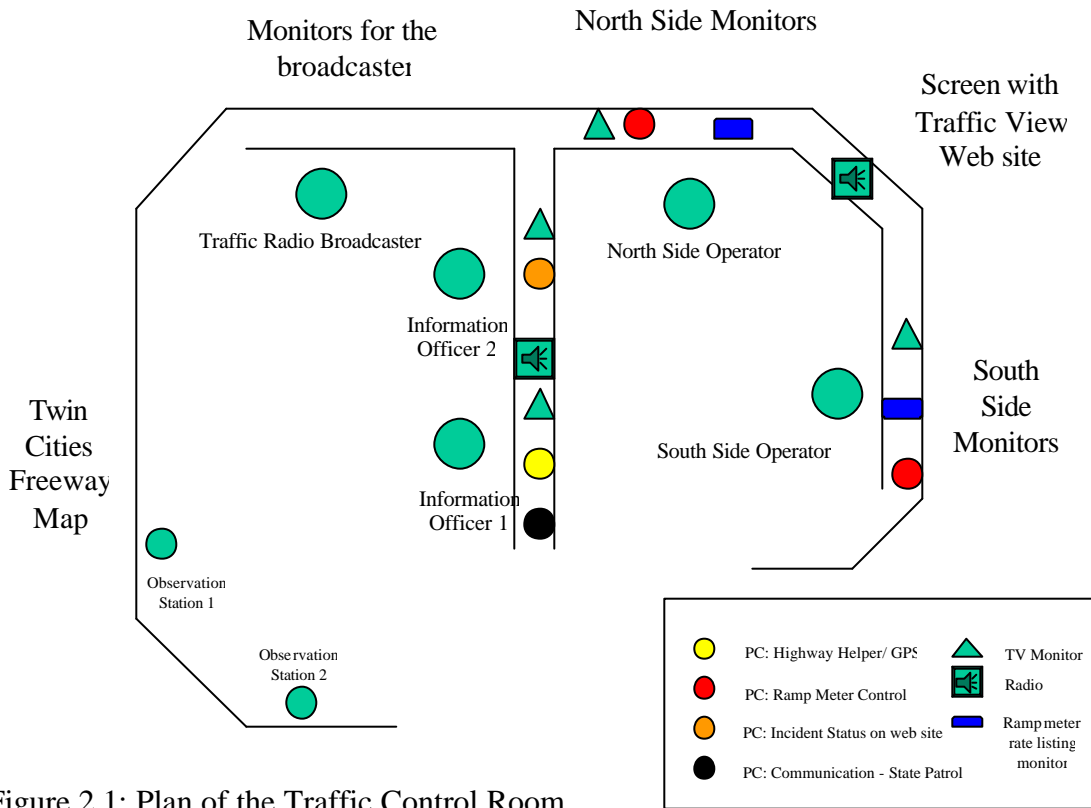


Figure 2.1: Plan of the Traffic Control Room

2.2.1.2 Information

Information Gathering

The primary sources of information regarding the field conditions are:

- Surveillance Cameras

The operators keep a constant watch on the monitors displaying the live feed from the field and look out for recurring and non-recurring congestion. Any discrepancy in the normal functioning of the system will be immediately noted and necessary measures will be taken in order to rectify the situation. The cameras are typically used to detect or verify incidents, inspect the operation of a ramp meter or length of a ramp queue or to check on individual motorists or pedestrians causing unsafe conditions.

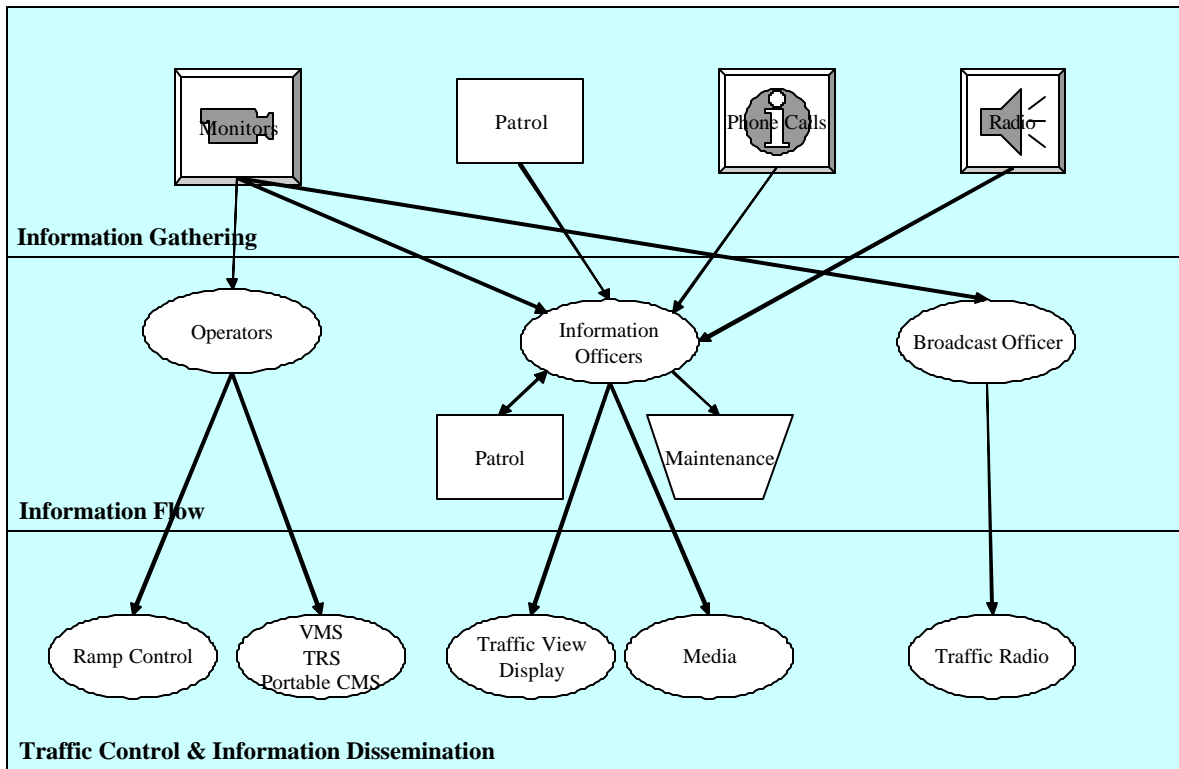


Figure 2.2: Traffic Control Room: Information Receipt, Flow and Dissemination

- **Radio Communication**

The operators and the information officers constantly listen to the radio communication of the state patrol (for monitoring purposes) in order to gain information regarding any discrepancy. Two-way communication is possible between West metro maintenance, East metro maintenance, and ESS/Highway Helpers.

- **Phone Calls**

Apart from the above sources, cellular phone calls from the travelers are another source of information. These calls are routed through the state patrol as the patrol receives them from the travelers and the TMC operators get the information through direct communication from state patrol or through radio messages. The state patrol also directs the calls to the TMC whenever appropriate.

Information Flow and Dissemination

The information flow between the staff members is shown in Figure 2.2. The detection through surveillance cameras of any discrepancy or abnormal situation could be done by any of the staff members. But this information is passed on to the operators as soon as it is known. Information is shared regarding the location of the incident/abnormal traffic

condition and the camera number to tune to. The information officer might communicate with the state patrol in order to check and confirm the situation.

The concerned operator checks the ramp metering rates currently being employed in the zone of interest and takes action to get the situation under control. This might include changing the ramp metering rate or overriding it and sending appropriate VMS messages to disseminate information regarding the situation to the travelers.

In case of an incident, the information officer keeps a log as to when the incident was noticed, when the concerned authorities were informed, what actions were taken to handle the situation, and when it was removed. The IO keeps in touch with the state patrol which is also keeping track of the situation. He informs information officer 2 (IO-2) whenever an incident is confirmed and removed.

The IO-2 in turn updates the traffic view site whenever an incident is confirmed or removed. The IO-2 also informs the media about the current incident. The broadcast officer provides regular traffic radio updates, including information regarding any incident detected, confirmed, or removed.

Example:

Incident: Multi-car accident on I-94 and Highway 252. Six cars are seen on the shoulder.

The sequence of steps noticed were:

1. The incident was detected by the North side operator and informs the rest of the staff about it.
2. IO1 contacts the state patrol and reports the incident.
3. IO1 logs the incident.
4. Information officer 2 broadcasts the message to the media
5. IO2 marks the incident on the traffic view display with the location, time, the message associated with the incident, and the places the closest camera looking at the incident on the cable salvo.
6. The North side operator does not notice any congestion. Therefore, no actions are taken to override the ramp metering or employ a VMS.
7. The broadcast officer makes an announcement about the incident in the traffic update on KBEM 88.5FM

2.2.2 Questionnaires

The second part of the study of control room operations involved the distribution of questionnaires to the staff members. A questionnaire is a useful tool in the study of a Traffic Control Room. With careful definition of terms, questionnaires can be both an efficient and effective means to collect information about operations, even in the absence of a consensus case definition. An effective questionnaire will endeavor to define its terms operationally.

An important part of the process was the preparation of the questionnaires. The objective in preparing the questionnaires was to get to know the personal opinions as well as suggestions of the staff members regardless of the policies, procedures, or actions they follow while performing their duty in the control room. The staff was encouraged to point out what they thought were the good and the bad aspects about the environment as well

as the operating procedures. The staff members were also asked to give suggestions regarding what they thought the control room was lacking.

The questionnaire was designed to include a breadth of topics. Emphasis was placed on the individual preferences and presentation of the data. The following are some of the topics that were included in the questionnaire:

Location and placement of the devices

The questioner attempts to identify staff preferences regarding the location and placement of the devices that were an integral part of the control room operations, like monitors, scanners, etc. It is extremely important to coordinate design and placement of various devices so that they make the whole process viable and also ergonomically efficient.

Data presentation

An important aspect of the control room operations is to ensure maximum utility of the data available to the staff members (operators as well as information officers). For example, currently the occupancy data is displayed throughout the freeway network on a big map. The data helps the operators locate the problem (congested) areas in the network. This gives rise to a discussion on whether it would be more useful to present any other data along with or instead of the current available data. This change might prove to be more beneficial to the operators and help them make useful decisions regarding the control devices based on the state of the network. The questionnaire attempted to clarify such issues based on the opinions of various staff members.

Functions

The Traffic Control Room operations consist of functions such as surveillance, traffic control, incident detection and management, equipment maintenance, and information dissemination. The questionnaire addresses topics such as:

1. The ease of access to the devices on the field such as cameras, ramp meters, and variable message signs.
2. The policies and strategies applied in using the control devices for ensuring smooth flows, safety of the drivers, and minimizing delays.
3. The ease of reporting and correcting any malfunctions in the equipment.

An attempt to find out if the facilities existing in the control room with respect to the access and control of various devices were easy and efficient to use. This culminated in an effort to suggest ways to improve the process in order to ensure optimum results.

2.2.2.1 Information expected from the questionnaire

The design of the questionnaires was done such that the participants were encouraged to articulate their views regarding the existing facilities in the control room as well as any suggestions they might have to make the process better. It is in this regard that the questions explored various options that facilitate the easy functionality of the control room operations. Some of the topics that were intended to be covered in the questionnaire

dealt with personal preferences for devices and their settings, presentation of relevant data, and various functions in the control room operations.

Surveillance is the backbone of efficient operational process in the Traffic Control Room. Traditional surveillance methods include observation of the live video from the cameras located on the highways and sometimes the patterns in the loop detector data over the entire stretch of the freeway network. At this juncture, it is extremely interesting to know more about any non-traditional sources for surveillance. Also, it is of great importance to explore the combinations of data that will indicate the state of the network more accurately than what is already done. Often, the operators would like to maintain control over the monitors they are watching. This is especially true when the operator is facing the situation of multiple incidents on the freeway network. A camera salvo is very useful in such situations. It is necessary to explore the utility of vesting the control of the salvo with the operators or the ease with which that can be done so that the situation at hand could be effectively managed.

Another important aspect of video surveillance and follow up during an incident would be to find out the cause for such situations. This requires the ability to access video footage from the recent past in order to analyze the situations. This implies that video storage plays an important role in the process of evaluation. There would be many issues cropping up regarding video storage, some of which might be related to privacy, to libel suits, etc., as well as technical issues like the method of video storage. It is worthwhile to explore the options that are available in this regard. One of the recent developments in the field of traffic management and control is the use of incident detection algorithms in the management operations. Although there has been widespread skepticism regarding the utility of such methods, it is essential to research individual opinions about the subject.

The questionnaire aimed at exploring the individual preferences of the staff members in order to arrive at what might be possible ergonomic configurations or settings. Questions were formed regarding the placement and location of the devices, the kind of tools the staff wants to use, what data they would want to view, and the format in which that should be presented to them. Furthermore, questions were formed regarding the actions taken by the staff members in certain situations. This gave an idea of the general policies and strategies adopted in the Traffic Control Room as well as individual reactions to various situations at hand. Overall, this line of questioning brought out ideas that aid an ideal environmental setting. This, in turn, facilitated an easy and efficient operating process.

The operator stations are equipped with monitor racks displaying the current conditions in the freeway network. Questions were framed regarding the current placement of the video rack and also the participants were prompted to make suggestions regarding the reconfiguration of this rack for it be more useful to them. Keyboard and mouse are the main input and pointing devices used by the operators. Technological progress brought about a variety of choices with regard to the input devices that could be blended into the TMC operations. Examples of such devices were touch-screen technology, light pens, and voice activated systems. Wide research is being conducted with these devices in order to establish their efficiency and usefulness. The use of such devices in the control room operations was explored through the questionnaire.

Appropriate interpretation of the current traffic conditions is only possible with the presentation of the right data at the right time. In view of this, questions were

designed to obtain information regarding the data the operators would like to see in order to make a good judgment about the situation and react to it. Currently the data presented to the operators comes from different sources and is being displayed separately. For example, the operator has access to the video feed, current ramp metering rates being applied, the messages being displayed in various variable message signs and also the loop detector data showing the state of the network in the form of occupancy counts. These data are not displayed at a single workstation but at different locations. Another aspect of presentation of data is graphical representation. Graphs tend to convey information better than pure numbers. Also, comparison statistics could be produced using graphical tools, leading to better judgment calls in dealing with ramp control or incidents.

The use of automatic alerting mechanisms can prove to be quite useful especially during a very busy peak hour session. Often, incidents and delays could be avoided by taking proactive actions rather than being reactive to the condition of the network. It is very likely that these conditions could be preempted by taking actions at the appropriate time. This is very much facilitated by the use of automatic alerting mechanisms that sense the build up of undesirable conditions in the network.

Efficient operations are also facilitated by easy access to the devices on the field. Currently, ramp meters and variable message signs are accessed through an OS2 terminal and status is updated through textual commands. The questionnaire also solicits opinions on this issue. Questions regarding the development of an integrated interface for accessing and manipulating various devices were posed.

The questionnaire also covers aspects such as the actions taken by the operators in various situations they face. It includes steps taken by them and also the information they seek and disseminate. As mentioned earlier, the primary sources for surveillance are the cameras along the highway showing the existing traffic conditions. But a situation might arise where the operators do not have access to the live video feed from a stretch of the highway they want to survey visually. It is in cases such as this that they have to depend on sources existing in the field or some other information source. This is particularly interesting in the case where the operators suspect that an incident has taken place and they want to verify it. The questionnaire aims to bring out the details about the kind of information the operators look for in situations similar to the above. Also, the actions taken by the operators in abnormal traffic conditions, like the agencies contacted and the immediate steps taken in order to ease the situation, is of very high importance in the study. This is especially significant due to the varying level of experience existing among the staff members. The study aimed at exploring the role of experience in assessing the situation at hand and reacting to it.

The discussion about the handling of various situations by the Traffic Control Room operators immediately gives rise to questions regarding the role of subjective decision-making in the operations. Often, this question is related to the functioning of the ramp metering and handling of incidents. It is worthwhile to know if there are any policies or strategies applied that make the operations operator independent. Also, policies regarding the usage of highway helpers, video taping of incidents, and steps taken to handle malfunctions of ramp meters during peak hours, which are considered operator independent, give rise to a lot of questions that pave the way for interesting research.

In addition to the questionnaires distributed to the staff members and the observations made during peak hours for three days, informal interviews with the staff members were also done. This helped in effective interaction with the members and getting to know the details about the management of the control room operations. It also helped encourage the members to freely discuss the issues at hand and also express their opinions about the functioning of the control room.

2.3 OBSERVATIONS AND QUESTIONNAIRES: A DISCUSSION

The preliminary observation of the facilities at the TMC and the responses from the staff members through the questionnaires raised interesting issues regarding the operations process. The following is an attempt to discuss various issues that ought to be covered in order to lay the foundation for the “next generation” traffic management center.

2.3.1 Environment and Devices in the Control Room

First, the physical environment of the control room had to be analyzed in order to determine if it is conducive for producing efficient results. This led to the question of ergonomics in the control room setting. Ergonomics deals with the applied science of equipment design in the workplace to maximize productivity by reducing operator fatigue and discomfort. It takes into perspective the human factors involved in operating a facility such as a Traffic Control Room.

The current personal workspace for the operators might not be ideal for the kind of work they are undertaking. It is essential for the operator to feel relaxed at the workstation in order to work efficiently. Since the work of the operator is computer intensive, the use of adjustable height workstations might help alleviate strain and will cater to the variety of preferences of the staff. Also the use of keyboard arms is believed to be more convenient than the traditional keyboards. The design of the workbench should be done in such a manner that operators would not have to exert themselves too much in order to access the devices that they use frequently. This should include even aspects such as viewing, apart from physically accessing the devices.

The operators have a personal monitor at their disposal so that they can tune to a particular camera of interest after noticing something irregular on the main monitors. Often the operator would want to look further upstream or downstream from the current location at the same time. A personal monitor does not give this choice to the operator. However, having a set of three to four monitors very close to the work bench and programming them to tune in and around the point of interest might be helpful. This will help the operator get the overall view of the conditions.

Camera control boards are an overhead in the operations of the control room. Operators should avoid using them in order to tune a particular monitor to a camera, as they have to manually punch in the camera number as well as the monitor. Instead, this process should be achieved through an interface in order to reduce the effort. Usually the use of pointing devices such as a mouse or a light pen is much more useful in such endeavors instead of a punch-in keyboard.

Overall, the current workstation lacks a personalized setting. One suggested way of implementing the aspects discussed above is to have a semi-circular workstation setting with a set of personal monitors. Also, the main monitors might be elevated to a

little bit higher level in order to eliminate any extra effort on the part of the operator to view them.

The operator has the option of running preset camera views or salvos over a particular monitor. This is a very useful feature in the TMC operations since the operators do not have to repeatedly keep tuning to the various cameras that they would want to view. However, as mentioned earlier, this feature exists only for preset salvos. If the operator wants to view a set of cameras of his choice, the sequence has to be programmed into what is called a scratch pad tour. But scratch pad tours cannot be saved. They are deleted as soon as the operator shifts to a preset salvo. It might be very useful to find a way to be able to temporarily save the user-defined salvos so that the operator has the flexibility to monitor various locations of the freeway section, possibly different incident sites, without manually entering the camera sequence each time.

During the course of observations of the control room operations, it was noticed that whenever staff members learn about an incident or an abnormal situation through various sources, they inform the remaining staff members of the possible situation. Sometimes, it was noticed that all the staff members try to find the location in question and inform the remaining staff members about the camera number. It is believed that in such cases, a function like transferring the camera view to the other staff members helps in making the whole process efficient.

Graphical User Interfaces (GUI) help to integrate various functions together and also simplify the access of devices. For example, currently the access to various ramp meters and variable message signs is done through an OS2 terminal and the status of each of these devices is changed by textual commands. The process of entering commands through text makes the process tedious. Instead, if the whole operation could be transferred onto a GUI, the operator would be just a click away from accessing the device. Furthermore, more devices could be programmed to be accessible through the interface, for example, cameras, traffic radio signs, and lane control signs, etc., apart from the usual ramp meters and variable message signs. Some of the staff members had expressed the necessity for having the map display of occupancy (approximated speeds) up close to the workstation in a monitor. According to them, the pattern of speed, based on current detector information, helps them understand the present conditions better; the changes could be followed better with the aid of a close-up monitor.

The means of access to the devices and other information is as important as the access itself. The provision of access to the information is nullified if the means to do that is complicated and not user friendly. The current means to access and manipulation of devices is by mouse and textual commands. Current research of user interaction shows that the popular means of access are mouse, light pens, touch screens, and voice activation. Light pens and touch screen technologies are believed to make the interaction between human and computer faster than what is achieved through a mouse. Some of the independent studies show that using light pens or touch screens increases the operation speed by nearly 50%. A light pen also offers more precision than a touch screen and has the advantage of not soiling the computer screen. Another area of research in this field is on voice recognition and voice activated interfaces. According to a study, in a test program augmented with voice input, the task completion time was reduced by 21%. In that study, the control group used keyboard and mouse, but was prohibited from using accelerator keys. The study questioned whether the speedup achieved with voice

(presumably by reducing mouse travel time to menus) could also have been achieved with accelerator keys. It was found that the speedup obtained via voice-plus-mouse (21%) was greater than that of accelerator-keys-plus-mouse, which was 15% for advanced users and 10% for novices. Thus we can see that, in critical control room operations, better speed of action could be achieved through non-traditional input devices. It is suggested that more research into such options, with regard to the everyday operations in the control room, should be undertaken.

2.3.2 Information: More data and representation

Currently, the main sources for information for the operators regarding the conditions prevalent in the field are the live video feed, loop detector data, and the communication from the state patrol and the media. The questionnaire attempted to gather the opinions of the staff members regarding the need and utility of additional information for making the operations work more efficiently. We feel that it is necessary to provide more information in the form of graphical representation of the data given by the loop detectors, graphs comparing the flows, occupancy, and speeds before and after some strategy applied to ease a situation etc.

Graphs are very useful for conveying information regarding the performance of a loop detector or comparing the current situation with the situation existing a time period ago. The malfunctioning of a detector can easily be identified by the use of graphs and plots showing the pattern of historical data, say two to three days, for that detector. The loop detectors giving out erroneous data are hard to distinguish based on the display of information on the map showing current speed, since it takes some time to show the erroneous results. However, with the aid of graphical representation of the historical information of various loop detector data, the operator can easily verify the functioning of a detector. Also, graphical representation can be very helpful in judging the effectiveness of a strategy applied to a situation in hand. By comparing the plots of various measures of effectiveness before and after the application of the strategy, the operators would have a distinct picture of how the traffic is reacting to the actions that they have taken.

As mentioned earlier, most of the operators feel that the map display of speeds for the entire network is very useful in understanding the state of the network. However it is felt that the display of more information apart from the speeds (which is essentially occupancy) would help the operators more in making the right decision when faced with undesirable conditions. Information such as flows, queue lengths, densities, and other such measures could also be displayed along with the current map. This would definitely paint a better picture of the current conditions.

The presentation of the above-mentioned plots and graphs also forms an important aspect in the efficiency of the working of TMC. For example, the operators said data could be presented in various different displays, which might end up cluttering the display monitor of the operator. An effective combination of the data should be arrived at in order to trade off between the amount of space occupied and presenting too much data in a single display.

The live video feed from the freeway network is currently viewed through monitors placed around the workstation of the operator. Another alternative in this regard would be the use of digital video and viewing it on the computer screen. The advantage is the enormous space savings because of the elimination of all the monitors. Also, cameras

could be chosen selectively to display the video of the current conditions and could be placed on the computer monitor to the liking of the operator. Especially with the advent of analog-to-digital converters in the market, the need for rebuilding the infrastructure for converting to such a system is minimum. The advantages of having the digital displays are also that they could be saved by a moving time window in the sense that the video for, say, the 15 minutes could be saved for all the displays. This might be useful in the cases where the operator would want to review the video of the recent past in order to assess the conditions leading to a particular incident or any such information. The video saved could also be utilized for training new operators regarding the handling of such conditions.

2.3.3 Traffic Control

During the process of manual observation of the TMC operations, it was noticed that most of the times when an abnormality in the traffic pattern was noticed or sensed, overrides of the current ramp meter rates were done. However, it was also noticed that overrides were considered and put into effect mostly by staff members with a reasonable number of years of experience in control room operations. Sometimes, it was observed that the overrides were more used as preemptive measure rather than as a reaction to the conditions as specified in the Control Room operations manual. This brings out the question of whether the control room policies and strategies have any relevance when the experience of the operator is taken into account. It also indicates that the guidelines should be revised based on the opinions and general practice of the operators. This would prove to be immensely valuable to the less experienced operators.

As mentioned earlier, a GUI is lacking for the access and manipulation of control devices such as ramp meters and variable message signs. The operational effectiveness of such an interface would be supplemented by the use of a representation schema based on the conditions existing in the network and the control applied. For example, use of color codes for the ramp metering rates for each of the ramp meters in the network enables the operator to analyze the situation better than when it is not provided. This is because a visual representation of the situation is grasped better than a numerical representation. Hence, the information that is currently viewed through numbers, that is, the current rate, the forthcoming rate, the mode of operation, could be replaced by visual means. This change is definitely expected to help the operator more. Furthermore, the process of manipulating the ramp meters, by overriding them, should be made easier by introducing interface mechanisms rather than using the textual commands.

Another issue that has to be considered in the discussion about traffic control is the management of heavy traffic conditions in the network when a critical ramp meter breaks down during peak hour operations. It is very important to take necessary and timely action in such cases in order to prevent cluttering up the highway. Currently, the procedure for handling this particular situation involves informing the Electrical Services Section (ESS) or writing a "yellow slip" for maintenance. Although, the occurrence of such a situation might be rare, it can cause rapid development of undesirable conditions in the network. This forms an interesting research issue in that the actions taken to ease the situation have to result in optimal conditions.

Variable message signs are deployed whenever there is an unwanted situation on the freeway and there is a sign upstream of it. However, research into results of the action

taken, that is, the message displayed, has not been done at a significant level. It is very important to do the above-mentioned study because it gives insight onto the best choice among the options operators have among the messages to be displayed when operators face various situations. The study involves the behavioral patterns of drivers and their reaction to various messages displayed, and so is very complicated.

2.3.4 Automated indications

The control room operators depend on the visual images from the field in order to detect any undesirable conditions in the system. On detection of any such situation, they take appropriate steps in order to gain a control of the situation. This is a reactive kind of action. What could be very useful is to have a proactive approach to the problem. That is, to try and detect the build-up of undesirable conditions, even though the current conditions indicate free flow. This is where the role of automated alarming techniques comes into the picture. The operators usually look for build up of queues both on the mainline and the ramps, occurrence of incidents, etc. If a technique could be developed where in the system gives an indication of a build up of queues or congestion on the mainline, it would immensely aid the operations. Some of the staff members have indicated that often queues on the ramp are hard to monitor. The queue lengths at every ramp can be monitored and based on some threshold values that could be initialized by alarms.

There has been some discussion on what kind of indicators could be used for alarms. The obvious choices were flashing indicators and sound-based indicators. The opinion is split among the staff members regarding the usage of either of them. Some members felt that sound-based indicators were the best while some other thought that visually attractive indicators such as flashing indicators would prove more useful in the control room operations. It is also felt that the indicators should be priority-based. That is, in the case of visual indicators, the flashing should be at a higher frequency for a more critical situation than when it is not. Similarly, if sound-based indicators are utilized, the volume of the indication should be more for critical situations.

2.3.5 Incidents Detection and Management

There has been a widespread skepticism about the utility of incident detection algorithms in the control room operations. Few traffic management centers in the United States which rely on or at least employ incident detection in their daily operations. There are various incident detection algorithms employing various detection techniques ranging from artificial neural networks to stochastic methods. Almost all of the incident detection algorithms utilize the loop detector information to predict the occurrence of non-recurring congestion.

Some of the staff members believe that the algorithms usually are not accurate enough to detect real life situations with as much efficiency as manual surveillance. The opinion that the variability in loop detector information due to malfunctions or any other reason might be interpreted, as non-recurring congestion by the algorithms was prevalent among them. A few of the members however were of the opinion that the latest technology in this area should be used if it really shows the promise of improving the efficiency of the operations in the control room. It is opined by that at least one incident

detection algorithm should be deployed in the control room operations for some period of time on experimental basis, and the performance of the algorithm should be gauged. However, the choice of the algorithm plays an important role in the acceptance or rejection of use of such algorithms.

Assuming that some form of incident detection algorithm is incorporated into the control room operations, it is necessary to supplement such a function with additional features in order to make the whole process efficient. For example, the indication of non-recurring congestion at some location on the freeway network should trigger a visually attractive message to which the operator can react. Furthermore, the operator should be provided with the necessary information related to the situation at hand, such as the nearest cameras to the location indicated (upstream and downstream) and a list of ramp meters in the vicinity, along with their current rates. Utilities such as these will enhance the whole system to function with optimum results.

Currently, the main sources of incident detection in the control room are the 911 calls from the drivers who have witnessed an abnormal situation. Usually, an undesirable situation is reported within 2-3 minutes. However, the information is relayed to the control center by the state patrol. It is also believed by some of the staff members that the information regarding these calls should reach the control center earlier than it already is. The members feel that the 911 caller data should be directly accessible to the verifiers in the control room. This will help cut the reaction time for the response.

2.3.6 Equipment Maintenance

Often the devices existing in the field are subject to breakdown due to various reasons like inclement weather or electrical malfunctions. Loop detectors are the major culprits in this count. Sometimes ramp meters as well as the cameras on the field are subject to such malfunctions. One of the responsibilities of the operator in the control room is to create a list of malfunctions for maintenance. Currently, the list is created manually, that is, the operator has to note down the device identification number and the details associated with it in order to include it in the maintenance list. This process should invariably be automated in order to make the process faster and efficient in the sense that the device should be added to the list electronically through the user interface rather than manually. All the details associated with the device should automatically be adjoined with the entry. Preferably, such entries should also be made into a database so that it would form a common source for entering as well as to retrieve from by the maintenance staff. The process could be further automated by assigning specific tasks to various staff members working for the maintenance crew.

Another aspect in the creation of a maintenance list is the automated entry of a device into the list. If a specific device has been giving no data for a specified period of time, say a loop detector not recording data for a day, it could be assumed that the device is subject to malfunction and added to the maintenance list. This will eliminate the effort involved in finding the faulty devices.

2.3.7 Integration of functions

The TMC consists of various functions that form the core of its operations. Some of the main functions of a control room are surveillance, traffic control, incident detection and management, maintenance and information dissemination. Over the years each of these functions and others have gradually been added to the operations process in almost all of the traffic management centers. However, the addition of such functions has not always been consistent with the ITS architecture, which is the basic framework for interaction of various products and services, in order to pave way for next generation traffic management centers. Often the functions exist as individual modules operating independently. This not only limits the capability of the process but also undermines efficiency of the whole system.

2.4 CONCLUSION

Based on the discussion from observation of the control room operations, it is opined that the need of the hour is the effective integration of all the functions into one comprehensive system. This will not only pave way to a system which would be easy to operate but also make the process more efficient. Automation should also be done in order to achieve fast, effective results rather than the current manual processes. Most of the staff members have a consensus on the need for more graphical representation of the data in order to improve the perception of the existing situation in the network. Easy access and manipulation of data and devices is enabled through effective user interfaces. The control room is lacking such interfaces in its operational process. Making at least these changes would pave way for developing the current TMC into a next generation TMC.

Chapter 3. TRAMLAB Design

This chapter describes the various features that the proposed virtual TMC should have, keeping in mind the TRAMLAB objectives (as discussed in Chapter 1) and the guidelines of the National ITS Architecture. The design issues arising from the need to implement these features and the proposed solutions to those issues are also discussed. Finally, this chapter describes the architecture of TRAMLAB followed by a detailed description of its modules.

3.1 VIRTUAL TMC FEATURES

The features that need to be incorporated in the proposed virtual TMC can be summarized as follows:

- It should be capable of enabling human interactions with systems and functions.
- It should have a training platform that emulates reality.
- It should have an evaluation platform to evaluate various combinations of traffic management schemes and test new technologies.
- It should emulate the TMC operations.

In addition, it should take into account the various deficiencies identified in the Mn/DOT control room study (discussed in Chapter 2). Furthermore, the architecture of TRAMLAB should be consistent with the guidelines set forth in the National ITS Architecture.

3.2 DESIGN ISSUES AND SOLUTIONS

In order to incorporate the aforementioned features, there was a need for emulating an actual TMC environment without access to field hardware. It was also necessary to emulate the responses to traffic management actions in the absence of actual deployment, in a real-time setting. Further, the user interfaces had to be flexible and adaptable so that one could easily make changes in the modules and dialogues and add new devices.

As part of the package of solutions, it was decided to:

1. Select a development environment, which is flexible and user friendly.
2. Use a simulator in the background to replicate reality.
3. Use historical data obtained from the TMC data distribution server in order to replicate demand patterns.

3.3 ARCHITECTURE

Once the solutions to the design issues were identified, an architecture was proposed based on the guidelines of the National ITS Architecture. The proposed architecture provides a common framework for sharing data between the major TMC functions, i.e., Traffic Surveillance, Traffic Control, Incident Detection, Incident Verification, Incident Response, and the field devices. It also enables the users to access, process, and organize the traffic information in an efficient and easy manner.

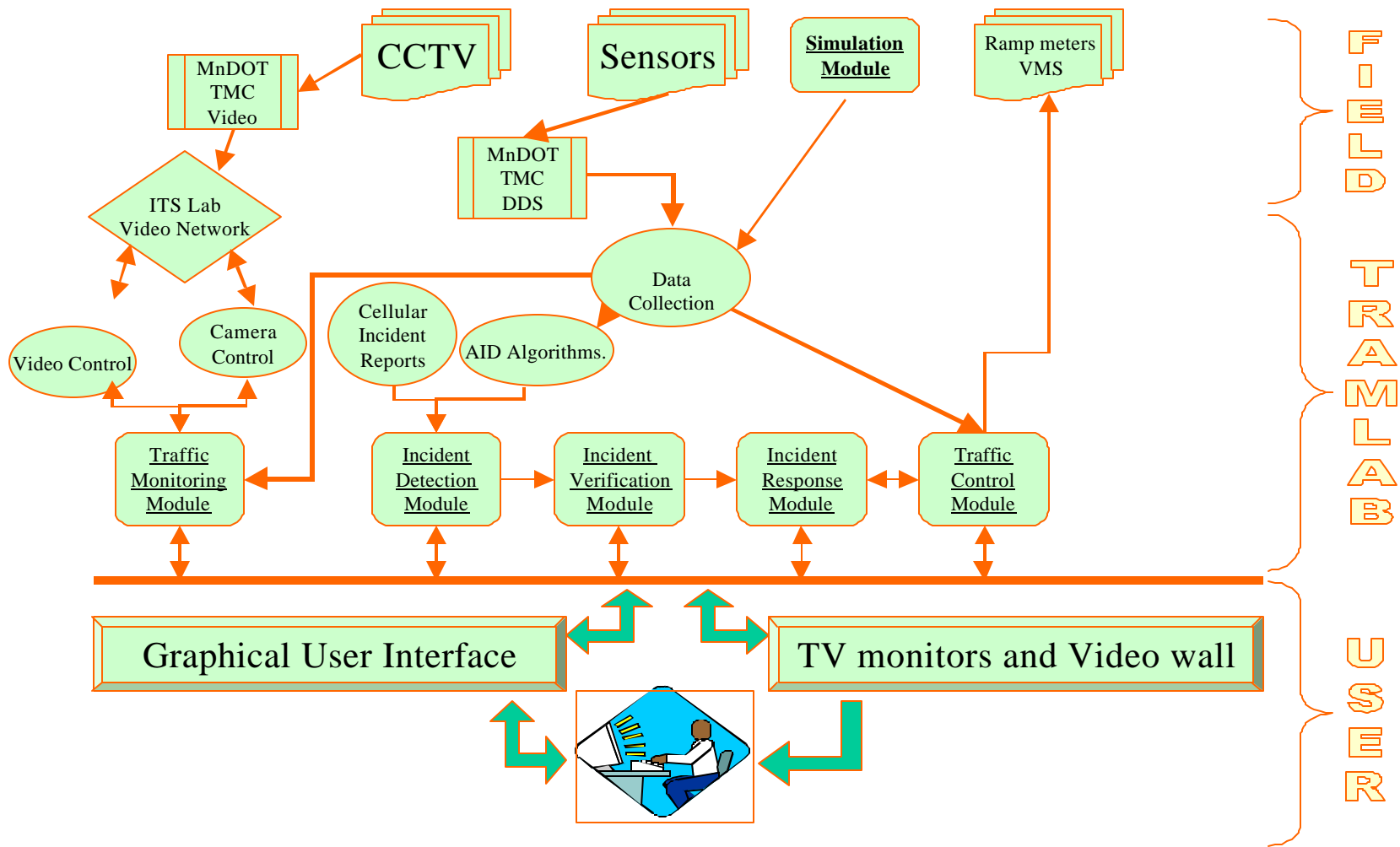
The architecture is divided into three major components as shown in Figure 3.1; these components can be categorized as field, TRAMLAB, and user. In the real-time setting, the field is the actual freeway network and the various installed devices, such as ramp meters, VMS, CCTV cameras, and others. With the exception of CCTV cameras in the laboratory environment or offline setting, the field is represented by a microscopic traffic

simulator. The second part of the architecture, TRAMLAB, is composed of the various TMC functions integrated together in a logical fashion.

For surveillance and data collection purposes, the flow of the information begins from the field objects detectors and CCTV cameras. The CCTV cameras placed at different locations on the freeways captures the traffic images, whereas, detectors record the volume and occupancy of each lane. This process continues throughout the day. The CCTV camera data is continuously sent to Mn/DOT TMC through fiber optic cables. This data is then displayed on the video wall, which is a part of the traffic monitoring module of TRAMLAB and contains 16 TV monitors placed at the ITS laboratory. The part of the graphical user interface of TRAMLAB, which represents the video wall, is directly connected to the video control in such a way that the operator has the option of switching a TV monitor on the video wall to a desired camera directly from the user interface.

The detector data collected by Mn/DOT's TMC is fed to TRAMLAB through the data distribution server. This input data is stored in a historical sensor database, which is later used by the microscopic simulator to simulate the actual traffic conditions. The simulated data is then used by the incident detection (automated incident detection algorithms) traffic monitoring and traffic control module of TRAMLAB. In case of any incident or any abnormal traffic conditions, the incident detection module generates an alarm and notifies the operator. The incident verification module then confirms the incident by monitoring the incident location on CCTV cameras. Once the incident is verified, the incident response module generates a response, which is fed to the traffic control module for implementing traffic management strategies (e.g., overriding the ramp metering rates, displaying VMS messages, etc).

Figure 3.1: TRAMLAB Architecture



3.3.1 Development Environment Selection: An expert system, G2

In developing TRAMLAB, a software development environment was needed. The expert system development environment provided by the GENSYM Corporation called G2 (1) was selected over other alternative development environments because the objective was rapid prototype building and testing rather than product development. This allows users to test new ideas and system components, and develop new user interfaces quickly and efficiently in order to prove concept. For example, the user can change system components such as Automated Incident Detection (AID) algorithms, ramp control strategies, simulators, etc. Furthermore, it provides the capability of running several algorithms in parallel with the same data in order to select the best for the particular case. The other important features that lead to selection of G2 are:

- It provides an object-oriented environment.
- It provides easy integration and communication with external applications, databases and real-time systems (e.g., simulators, control algorithms, etc.).
- It employs a natural language editor that allows the user to rapidly program the graphical user interface.
- It can easily add/remove modules without affecting the system's operation.
- It runs on a variety of hardware platforms.
- It provides multi-user access.
- It provides connection with the Web.

3.3.2 Simulator Selection: AIMSUN2

AIMSUN2 [2] was selected over the other simulators because it is presently one of the best microscopic simulators worldwide and its developers agreed to provide detailed documentation and support. As a microscopic simulator, AIMSUN2 provides the ability to model the behavior of every single vehicle in the network, using several driver behavior models (car following, lane changing). It can deal with various types of traffic networks, such as freeways, ring roads, arterials, and any combination of them. Further, it is capable of modeling different types of traffic control, such as traffic signals, ramp metering, VMS, and others. In addition, AIMSUN2 can model various vehicle types, for example cars, buses, etc., which enables it to simulate any kind of traffic detector's measures: counts, occupancy, speeds, etc. Other important features that lead to AIMSUN2 selection are:

- It has a user-friendly user interface, which makes it easy to build the network geometry.
- It provides animation of vehicle movements.
- It emulates driver decisions based on simulation input (VMS messages).
- It allows communication with external application (Generic Environment for Traffic Analysis and Modeling).
- It allows Dynamic Traffic Assignments.

3.3.3 Data Availability and Representation

Presently, there are about 3,780 loop detectors, 241 CCTV cameras, 64 VMS, and 430 ramp meters installed in the Twin City freeways. TRAMLAB receives data from these field devices through the Data Distribution Server (DDS) of the Mn/DOT TMC. The 30-second detector data received from the DDS is stored in the access databases, which is later used as historical data for creating the input demands in the simulator.

As mentioned earlier, G2 is an object-oriented development environment. Therefore, G2 provides the options for its users to define user-defined objects, for example, detectors, freeway sections, detector stations, VMS, ramp meters, etc. As in any other object-oriented programming language, the definition of any objects is called the class of that object. It is not necessary for the objects to represent just the real-life field devices, such as ramp meters, but can also be used to encapsulate all the information describing an event, i.e., an incident. Each object has two forms of visualization: one is in the form of a table that shows the various attributes of the object. For example, the object detector has attribute's name, ID, volume, occupancy, speed, density, etc. The second form of visualization is the graphical view of the object, which can be anything from an icon to a picture or a map. The various objects used in TRAMLAB and their attributes are described next.

3.3.3.1 Detector Station

The small workspace shown at the top left corner in the Figure 3.2 shows the icon of the class detector station and its object. The table shown in the same figure shows the class-specific attribute of the detector station. The main attributes of detector stations are volume, occupancy, speed, density, and location of the station, operating section name, mile point, type, and status of the detector station. The statement, "Volume initially is given by 30sec-integer-var" means that the attribute volume of the class Detector Station gets its value from the variable 30sec-interger-var. Since the attributes volume, occupancy, speed, and density values need to be updated every 30 seconds, these attributes are kept as an object of 30 second variables (30sec-integre-var, 30sec-integer-var, 30sec-speed-var, and 30sec-density-var). These 30-second variables are also object definitions. The other attributes of the detector station are kept constant. This class has been configured such that, when in operator mode, pressing Ctrl + any mouse button on any detector station shows the detector station information dialog.

This information dialog is used by the operator to view information provided by each detector station, such as the volume, and occupancy values.

3.3.3.2 Detector

Figure 3.3 shows the details of the object detector. The small workspace at the top left corner shows the icons of the class detector and its object. The class-specific attributes of detectors are: name of the detector station to which this detector belongs, name of the highway, direction, milepoint, volume, occupancy, speed, density, status, type, and lane number of the detector. Since the volume, occupancy, speed, and density values need to get updates every 30 seconds, these attributes are kept as an object of 30-second variable.

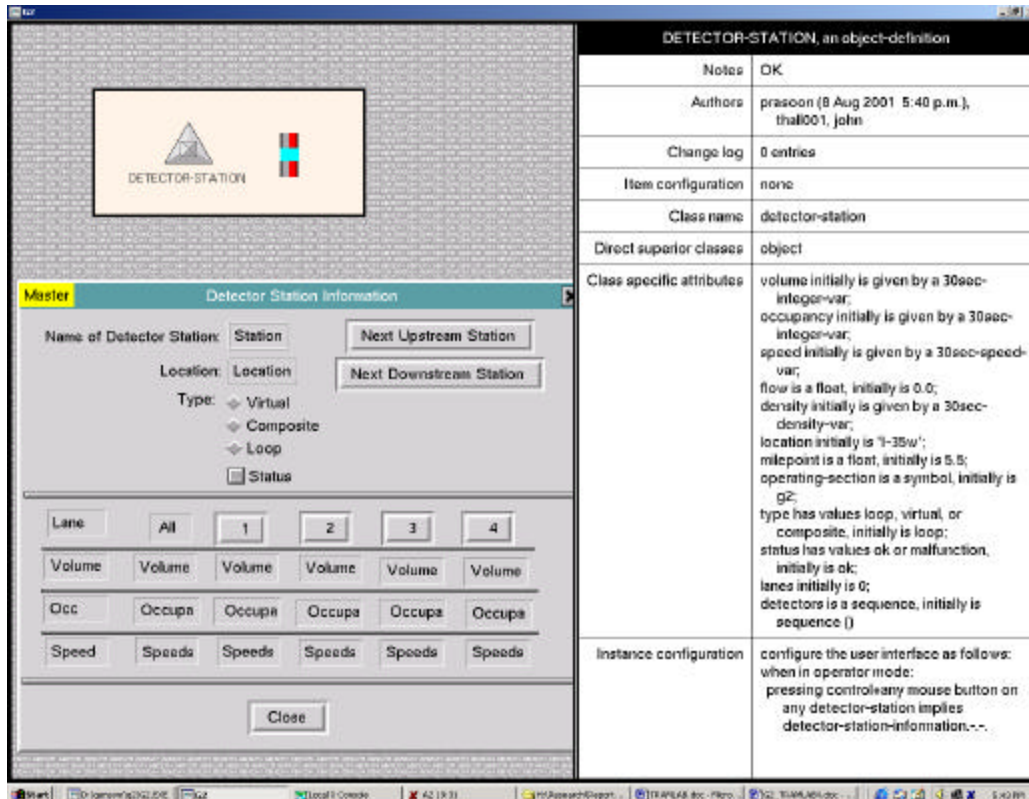


Figure 3.2 Table and dialog of Detector Station

All the other class-specific attributes are constant. The dialog box shown in Figure 3.2 is the detector information dialog.

3.3.3.3 Operating Section

Figure 3.4 gives information about the class operating section and the icon shape of its object. The table in the figure shows the class-specific attributes of the operating section. The operating section class is designed in such a way that the entire freeway stretch that is modeled is covered by the operating sections such that no single section would have more than one object of any of the following: detector station, ramp meter, camera, variable message sign, etc.

Further, the operating sections are connected to each other in a manner such that easy navigation is facilitated between sections having mainline detector stations or cameras. The main class-specific attributes are the names of all the filed objects that are present under the operating section, for example, VMS, camera, detector station, etc. Class-specific attributes of the operating section are *mainline_detectors*, *exit_ramp_station*, *entrance_ramp_detector1*, *entrance_ramp_detector2*, *ramp*, *camera*, *location*, *highway*, *accident*, *vms*, *zone*, *incidents*, *type*, etc. This class has been configured such that, when in the operator mode, pressing Ctrl +Shift + any mouse button on any operating section starts a procedure named *Createincident* (explained in Chapter 4), which artificially creates an incident at the location of that operating section. This property is used for creating incident scenarios.

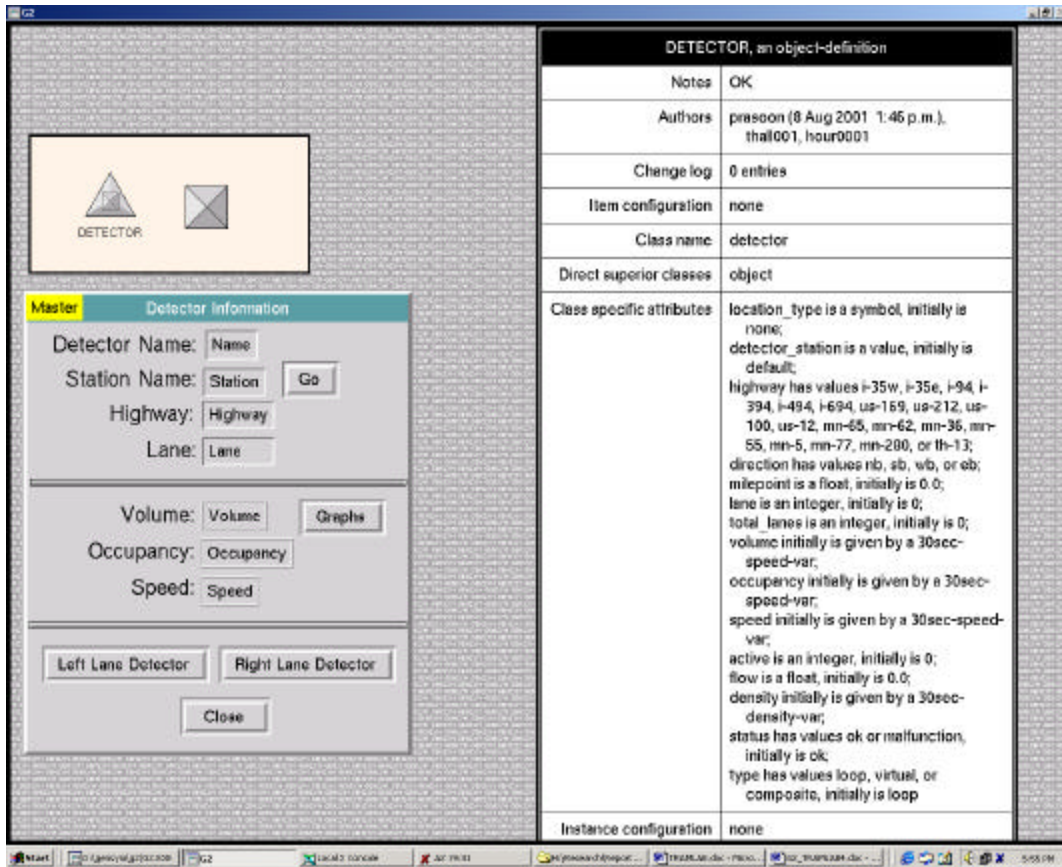


Figure 3.3 Table and dialog of Detectors

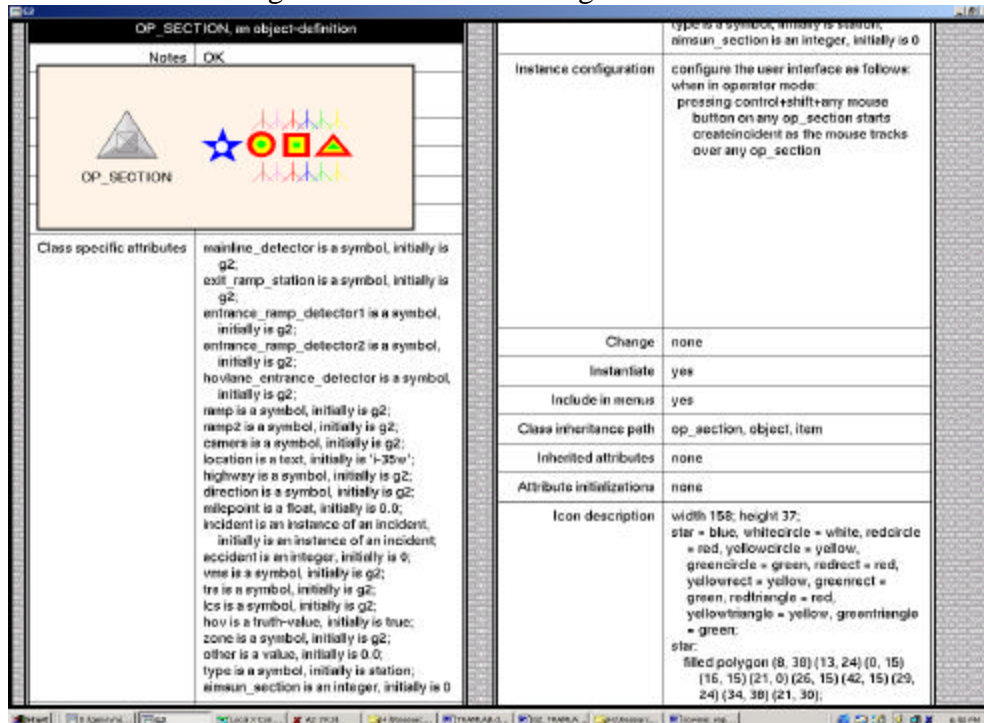


Fig 3.4 Table and dialog of operating section

3.3.3.4 Ramp Meter

Figure 3.5 gives information about the class Ramp Meter. The object of this class is called the ramp meter. The table shown on the right side of the figure shows that the class ramp meter inherits some of its attributes from the class object. The class-specific attributes of this class are *Ramp_id*, *Computed_Meterrate*, *Field_Meterrate*, *Override*, *rate_control*, *Red_time*, *number of cars per green*, *queue delay*, etc.

Rate control is the name of the dominating control, which can be either zone control, occupancy control, or the default control. The *Computed_Meter* rates for every ramp meter is calculated by the ramp control algorithm and passed to each ramp meter object through the GSI-CPI (G2 Standard Interface- Control Plan Interface). The GSI and CPI are discussed in a later part of this chapter.

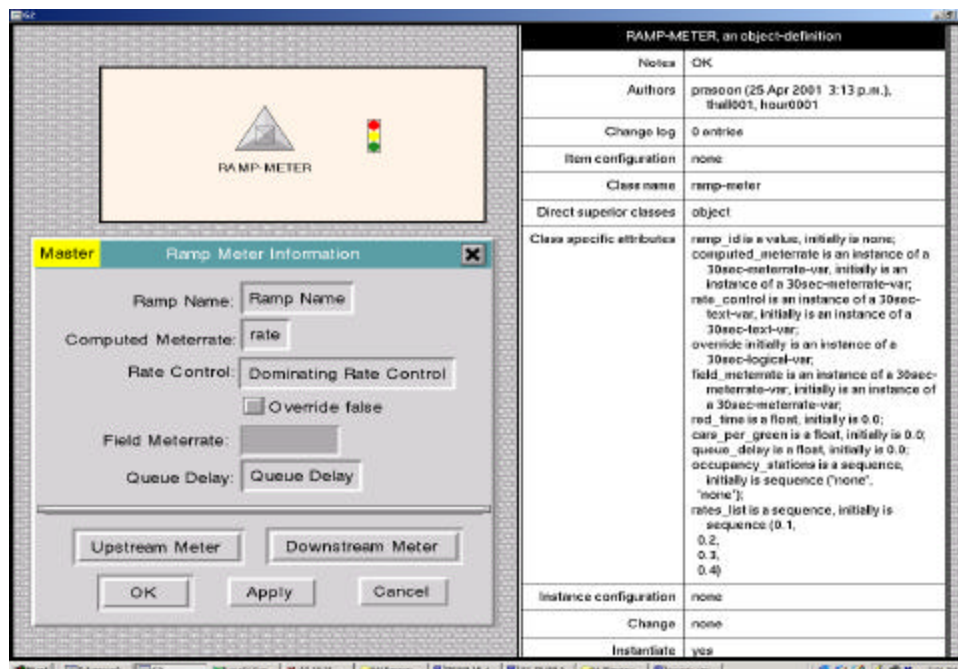


Figure. 3.5 Table and dialog of Ramp Meter

The *Field_meterrate* is same as the *computed_meterrate* if there is no override. However, if the operator overrides the metering rate by entering the overridden rate in the *Field_meterrate* edit box and by clicking on the override button, the field meter rate is fixed to the overridden rate until the override is removed. The overridden rates are removed by entering the rate 0 in the field meter rate box.

3.3.3.5 Camera

Figure 3.6 shows the icon of the class called Camera and its corresponding object beside it. The object of this class is called camera. The table shown on the right side of the figure shows that the class camera inherits some of its attributes from the class object. The class-specific attributes of this class are *location*, *id_number*, *channel_no*, *type*, and

section. All these attributes have a constant value for a camera. These attributes can be changed whenever required. The attribute location explains the location of camera.

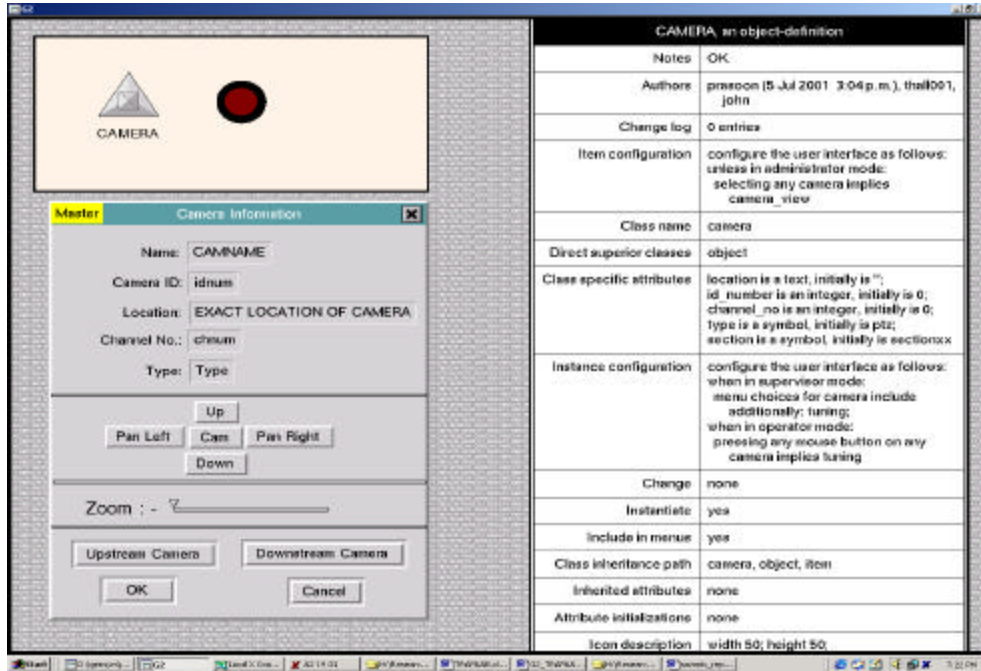


Figure. 3.6 Table and dialog of Camera

Every camera has a different *id_number* but the channel numbers can be the same. The attribute section shows the name of the section on which the camera is located. The class camera has been configured such that when in supervisor mode, the table of the camera shows the options for tuning but when in operator mode just clicking with the mouse on the camera implies tuning.

3.3.3.6 Variable Message Sign

Figure 3.7 shows the icon of the class variable message sign (VMS) and its corresponding object beside it. The object of this class is called variable message sign. The table shown in the left of the figure shows that the class VMS inherits some of its attributes from the class objects.

The class-specific attributes of this class are *location*, *highway*, *direction*, *message*, *action* and *status*. Operators can either select the predefined messages from the scroll area or they can enter new messages by writing in the message edit box of the variable message sign dialog. The attribute “Action” stores the predefined messages of VMS. As soon as the attribute “message” of the VMS receives any value a “whenever rule” invokes a remote procedure call (explained in Chapter 4). This procedure passes the text of the message from G2 to the simulator.

3.3.3.7 Metering Zone

Figure 3.8 gives information about the class zone. The class-specific attribute of zone are *zone_id*, *highway*, *direction*, *the zone capacity*, *smoothing factor*, *bottleneck capacity*, *turn-on-parameters*, *turn-off-parameters*, and *the volume threshold values*. The zone dialog receives the ramp metering rates from the corresponding ramp meters.

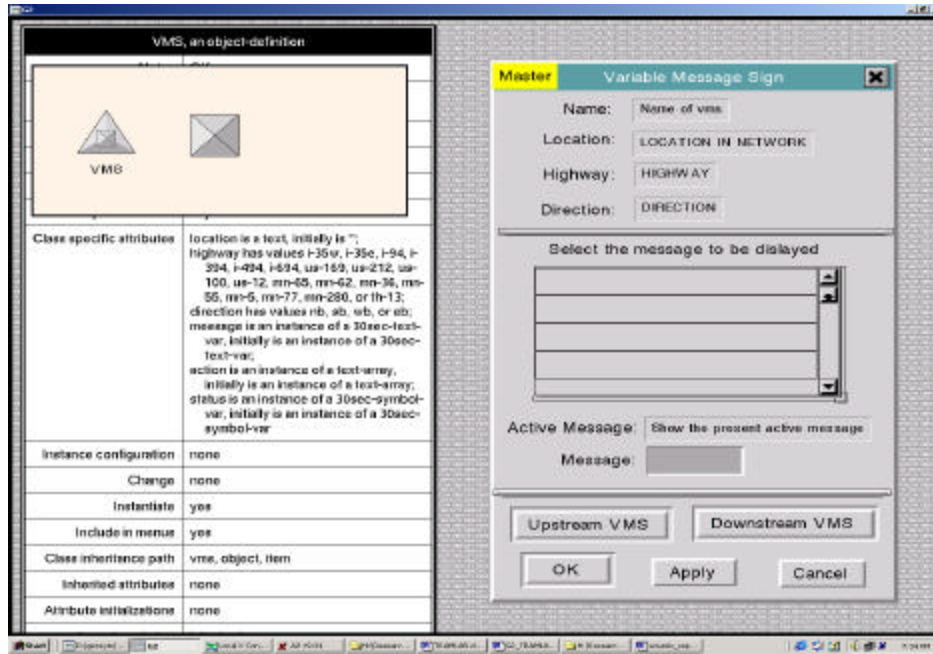


Figure 3.7 Table and dialog of Variable Message Sign

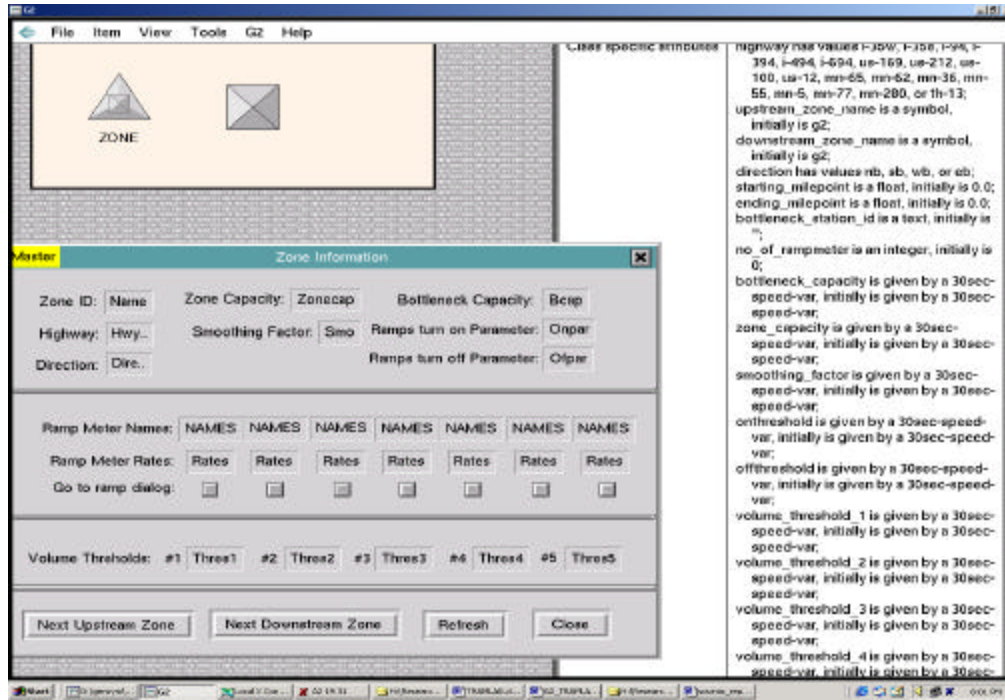


Figure 3.8 Table and dialog of Zone

3.3.3.8 Incident

Figure 3.9 shows the icon of the class Incident and its corresponding object beside it. The object of this class is called incident. This class inherits some of its attributes from its direct superior, class object. The table shows the class-specific attributes of incident. Unlike camera and detector stations, incident does not have any specific icon. The class-specific attributes of incidents are location of the incident, time at which the incident took place, duration, vehicle types involved in the incidents, number of vehicles involved, number of lanes affected by the incident, severity of the incident, etc.

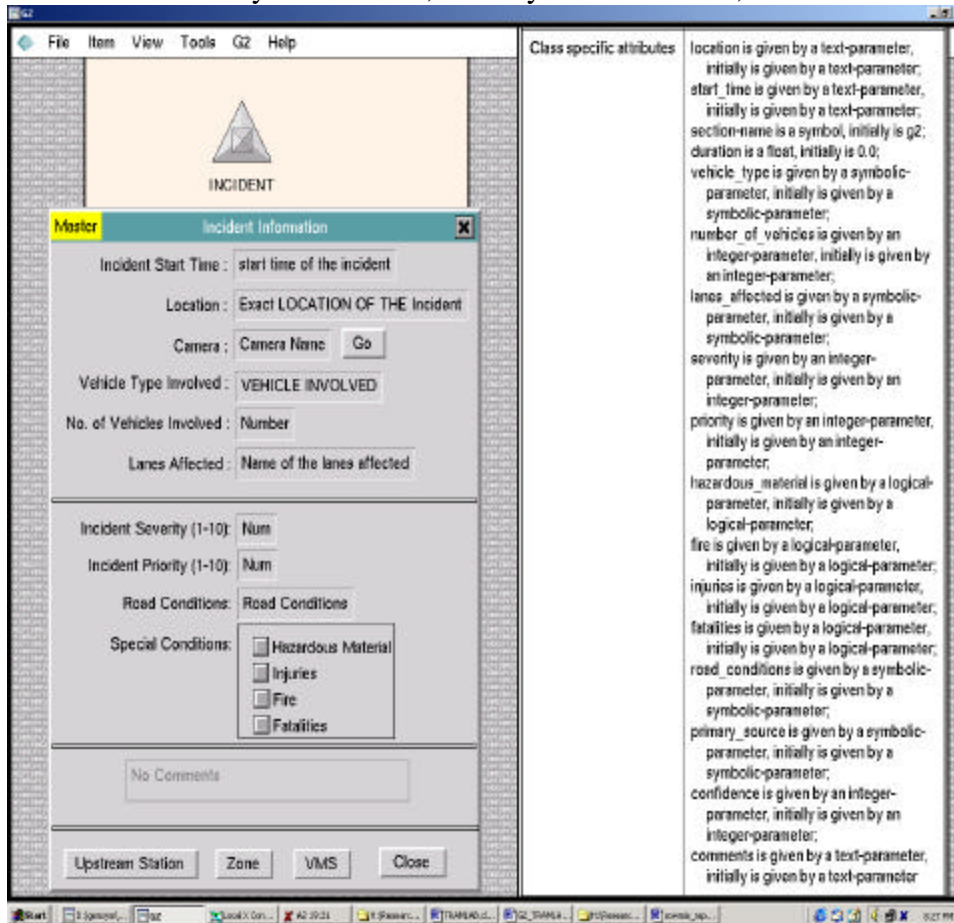


Figure 3.9 Table and dialog of Incident

3.3.4 The Laboratory Environment

As mentioned earlier, the objective of TRAMLAB is to create a laboratory environment for testing various control room operations and to evaluate emerging technologies such as new traffic control techniques and incident detection and management schemes. Therefore, the laboratory environment of TRAMLAB was designed in such a way that the various physical and software components are integrated together. The laboratory environment of TRAMLAB can be further classified into: The physical environment and the computing environment.

3.3.4.1 The Physical Environment

The physical environment of TRAMLAB consists of physical entities such as ITS lab video wall (1), operator workspace (3) and operator's personal monitor (4) (Figure 3.10). TRAMLAB can currently accommodate up to two operators (2).

The ITS lab Video wall consists of 16 TVs, which receive video feeds from the various CCTV cameras placed at different freeway locations. In other words, the video wall is used for monitoring the freeway traffic. But, presently the ITS lab users do not have control over the pan, tilt, and zoom properties of the CCTV cameras. This is because Mn/DOT TMC has not yet been able to provide the lab users with full camera control. However, in the future if camera control is provided, then TRAMLAB can also be used as a real-time traffic management tool.

It has been observed that the control room environment and the design of the operator workspace and system interface, such as height of the workstation, system controls, and computer displays, can have a major impact on the efficiency of the operators. To help ensure that the operators perform their tasks effectively with minimum stress and errors, the operator workstations have been designed in such a way that the operator can adjust them according to their comfort. The appropriate positioning of the personal monitors and the video wall helps operators to concentrate and work efficiently.

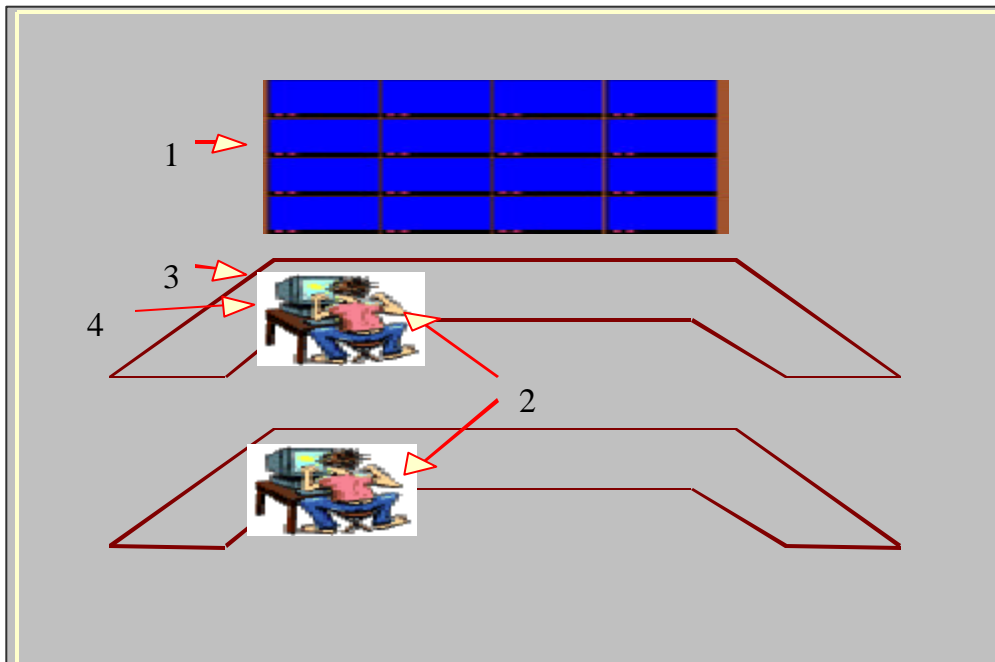


Figure 3.10 TRAMLAB Physical Environment

3.3.4.2 The Computing Environment

The computing environment of TRAMLAB consists of three software components: TRAMLAB Graphical User Interface (GUI), a traffic simulator, and the control plan interface. The following sections explain these software components in detail.

3.3.4.2.1 *Graphical User Interface*

The graphical user interface available in TRAMLAB provides an enhanced environment for conveying messages and displaying traffic conditions. Graphics makes the interaction between a computer and a user efficient by manipulating all the objects on the screen. TRAMLAB attracts users to its graphical display and improves usability by providing a user-friendly interface. As described earlier, all objects in TRAMLAB can have a visual representation. The user can use the mouse to operate the system without typing anything.

TRAMLAB provides a unique feature that allows the system to not only be easy to use but to easy alter too. Users can create new objects and displays and organize them in any form they please, even while the system is in operation. For example, the user can create graphs of any variable in the system and place them anywhere on the screen. These graphs get updated whenever the value of the variable changes.

As mentioned earlier, TRAMLAB is intended to demonstrate new ITS technologies, to test new freeway management functions, and to create a training platform for TMC operators. The flexibility of the system in terms of visualization and with the natural language programming interface allows untrained users to easily put their ideas into reality. Systems created under more traditional development platforms suffer from long turnaround between design and implementation.

3.3.4.3 *Knowledge Representation*

In TRAMLAB, the knowledge associated with various objects is accessed and manipulated through mechanisms called rules and procedures. A rule is a statement that generally has the form of “if ... then...” where the “if” part is called the antecedent and the “then” part the consequent. The antecedent of a rule specifies certain situations; the consequent of a rule decides what actions should be taken when the situations are met. Rules provide a formal way of representing recommendations and strategies and are best suited for expressing heuristic knowledge that is derived from experience. Rules are invoked by either forward-chaining or backward-chaining mechanisms. Forward chaining (data-driven reasoning) is “an inference method where rules are matched against facts to establish new facts” (3). Backward chaining (goal-driven reasoning) is “an inference method where the system starts with what it wants to prove. In addition, rules can be executed in a timely manner, e.g., every 30 seconds rules start the update chain for the volume information in the detector stations.

In TRAMLAB, a rule can apply to a specific object of the object, or apply to all the objects of the object. For example, there is only one rule that controls the change of the color of operating sections based on the occupancy of its detector station; this rule applies to all the operating sections currently present in the system. Rules that apply to a class of objects instead of a specific object are called generic rules. The ability to write generic rules makes the system more compact and efficient. It is important to note at this point that the system invokes the rule simultaneously for every object in the given class without any additional consideration on the part of the developer.

Example of a generic rule is as follows:

“whenever

the override of any ramp-meter r receives a value and when the override of r is true then

start set_override_rate(the name of r, the field_meterrate of r) across polaris”

The rule shown above is used for overriding the ramp metering rates. Whenever the operator wishes to override a particular ramp meter, he fills in the new field rate and sets the override button to true, in the ramp meter dialog. This action changes the attribute override of that ramp meter to true and thus this rule is initiated. This rule then calls a remote procedure named set_override_rate and passes two arguments, name and the changed field rate of that ramp meter, to this procedure. This procedure then overrides the rate of this ramp meter in the simulator through TRAMLAB-simulator interface.

In addition to the rules, in cases where repeated sequential manipulations of data is needed, the user can define procedures. A procedure in TRAMLAB is much like a function in ordinary programming languages; it contains the sequence of operations. The control path of rule operations cannot be explicitly defined, while in procedures the control path is determined by the developer. Procedures can be invoked from rules or other procedures. For instance, in TRAMLAB all of the AID algorithms are written in the form of procedures. Like rules, TRAMLAB’s procedures work in real time. Procedures and rules execute concurrently, based on priorities. Apart from rules and procedures, functions and formulas are also used to represent knowledge in TRAMLAB. G2 provides an interface for linking foreign functions written in C or C++.

An Example of a procedure is as follows:

graphs (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win: item-or-value, OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)

```
T, S : class trend-chart;
i, j: integer;
O: symbol;
begin
O = the name of OrigDialogLauncher;
for T = each trend-chart upon graphswksp do
  i = the item-x-position of T;
  j = the item-y-position of T;
  create a trend-chart S by cloning T;
  transfer S to the subworkspace of the object named by O;
  make S permanent;
end;
show S;
end
```

The procedure shown above is called when the operator presses the button ‘Graphs’ on the detector dialog, so as to observe the variation of volume, occupancy, and speed of that detector with time. This procedure transfers the pre-made trend charts (graphs) of volume, occupancy, and speed to the sub-workspace of that detector and opens the subworkspace.

Explanation of the code: As it is really time consuming to update the volume, occupancy, and speed trend charts for all the detectors every 30 seconds, it was thought to update the trend chart of only those detector whose trend charts the operator wishes to observe. For this purpose, three trend charts corresponding to volume, occupancy, and speed were made and placed in a workspace named *graphswksp*. The updating capabilities of these trend charts are kept in such a way that these trend charts take their updating values from their superior workspace. So, whenever the operator wishes to observe trend charts for a particular detector, the trend charts from the workspace *graphswksp* are transferred to the sub-workspace of that detector. Since detector values get updated every 30 seconds, the trend charts also gets updated. The history-keeping ability of detector variables makes it possible to view the variation since the start of the simulation.

3.3.4.2 Data Interface

TRAMLAB is a data-driven system that responds to events. The system obtains its data in several ways. Including sensor input, user input, the inference engine, a simulator, and other external processes. Execution of rules and procedures, user input, and asynchronous data from data servers will provide values asynchronously, while the inference engine, the simulators, and broadcasted external data from data distribution servers provide values when they are needed or scheduled.

When a variable needs a value, the inference engine will attempt to get a value for the variable through three sources: specific formulas, generic formulas, and rules. The inference engine first checks all specific formulas about the variable and then the generic formulas and, if both fail, the inference engine will search rules via backward chaining. G2 provides several data interfaces for supplying data from external processes. One of those interfaces, G2 Gateway, is used in TRAMLAB to exchange data with the simulator. As mentioned earlier, AIMSUN2 also provides the facility to integrate an external control system in order to provide rates for various ramp-metering rates that are modeled. A module called the Control Plan Interface (CPI) was developed here at the University of Minnesota, through which implementation of various ramp metering algorithms could be done in AIMSUN2. The G2 Gateway functions are integrated with the CPI so that the TRAMLAB interface is integrated with the simulator as shown in the figure 3.11. First of all, freeway network geometry is created in Traffic EDItor (TEDI) and historical data is fed into it so as to generate the simulation states. After the simulation states are generated, AIMSUN2 is run for the whole simulation period. During this period, the detector data is transferred to the CPI every simulation cycle. CPI calculates the ramp metering rates using the ramp control algorithm and sends the ramp rates back to AIMSUN every 30 seconds. G2 Gateway requires extensive data structures into which the simulator can transfer the data corresponding to each detector, detector station, and ramp meter. It is from these data structures that the TRAMLAB interface receives data regarding the volume, occupancy, speed and density of detectors and metering rates for any ramp meter.

The Ramp control algorithm is used to calculate the ramp meter rates and these rates are passed on to the simulator through the control plan interface (CPI). The interface also helps in transferring data from the simulator to TRAMLAB and vice versa.

3.3.4.4 Data structures in G2 Gateway for data storage

Adaptive ramp metering could be implemented through the use of CPI. AIMSUN2 simulates data for each of the detectors in the network for a cycle. This information is passed on to the data structures present in the CPI-Gateway combined module. Based on that data, the metering strategy for the next cycle is determined by the ramp-metering module. During this process, all of the data regarding the current detector information and the ramp meter information is stored in a detector map and a meter map respectively. A map is a class wherein objects could be stored based on a predefined key. The key is utilized in retrieving information from the map about the object. In our case, the detector map consists of objects of the class “detector”, which are stored based on their identification number that serves as the key. Similarly, the meter map consists of objects of class ‘ramp meter’, which are stored based on their identification number. As mentioned earlier, AIMSUN2 sends the detector information to the detector map, based on which ramp metering rates are decided by the ramp metering module and is stored in the meter map. This information is later transferred to AIMSUN2 for modeling the next cycle.

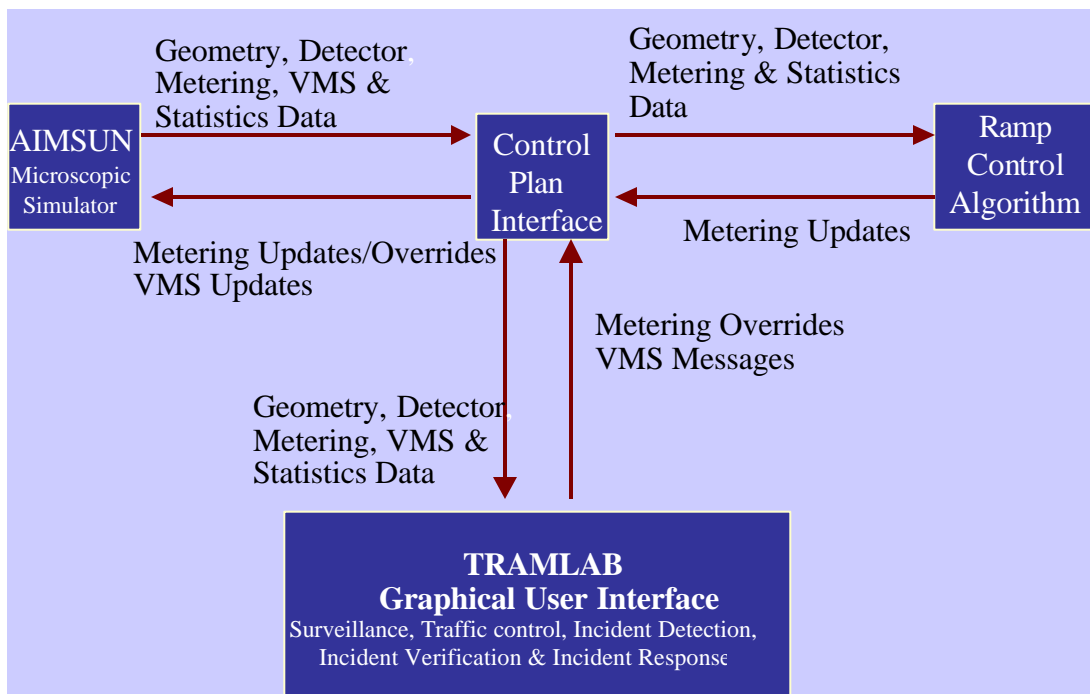


Figure.3.11 TRAMLAB-Simulator interface

G2 Gateway utilizes the information stored in these maps to transfer data regarding the detectors and the ramp meters to the TRAMLAB interface. On requests from the variables, which are created in TRAMLAB, G2 Gateway searches for information relevant to the requesting variable and returns the data to it. This is done for all the variables registered with the G2 Gateway and that seek data from it. The same process holds true for retrieving the ramp metering rates from the meter map.

3.3.4.5 G2 Standard Interface (GSI) description

G2 Gateway or GSI can be utilized to develop interfaces, or bridges, that support two-way communication between dynamic external processes and G2 applications (like TRAMLAB). Through GSI to an external system, TRAMLAB can quickly obtain real-time data that it needs to make intelligent control decisions in a time-critical processing environment. TRAMLAB can also update the state of an external system for example, by writing or updating a record in a database etc. GSI bridge process and a TRAMLAB process run concurrently, enabling applications like TRAMLAB to continue to perform their task while GSI manages the communication between TRAMLAB and an external system. GSI can communicate across networks that use TCP/IP or DECnet protocol. GSI handles details of network communication automatically, enabling the development of distributed systems among heterogeneous platforms without having detailed knowledge of protocols or of network software.

3.3.5 Module Description

TRAMLAB is an integrated system with control room operation functions (modules) assigned to the major freeway management functions. In this section the module design and capabilities are presented; the functions (modules) that comprise the system are:

3.3.5.1 Traffic monitoring

The Traffic Monitoring (TM) module has two major functions. The first function is to provide operators with the means to effectively control the surveillance system and the second is to provide an overall view of the current traffic conditions in the network. The main part of the TM module is the freeway network map display. In the map display, a graphical view of the whole network is presented to the user. The map display contains static and dynamically changeable objects. The static objects are part of the background image and are map signs showing freeway names, side streets with their names and other geographical information. Other objects controlled by the system are placed on top of the static image; for instance, the I-494 freeway is represented with one line in each direction in Figure 3.12.

Operators have an option of viewing only the desired field objects (camera, ramp meter, etc.) on the map by selecting the corresponding objects from the list containing all possible objects and they can access information on those objects by simply clicking on the graphical view under the text “Select the items to be viewed on the map” in the above figure. The boxes under the text “Devices in use” shows the number of presently overridden ramps, number of presently active incidents, and the number of VMS, which has some relevant message displayed on them and the slide bar box under these boxes show their respective locations.

Based on the indicators (the color of the roadway sections) shown in the map, operators can easily judge the level of congestion at different locations. They can choose the region that they would want to survey and have easy access to information like the video display, detector station, and ramp metering information. In order to view more specific information of a particular location, for example, to monitor the variation of

volume, occupancy and speed, operators can select the option “mainline detector” from the menu and can see the graphs corresponding to each detector (Figure 3.13).

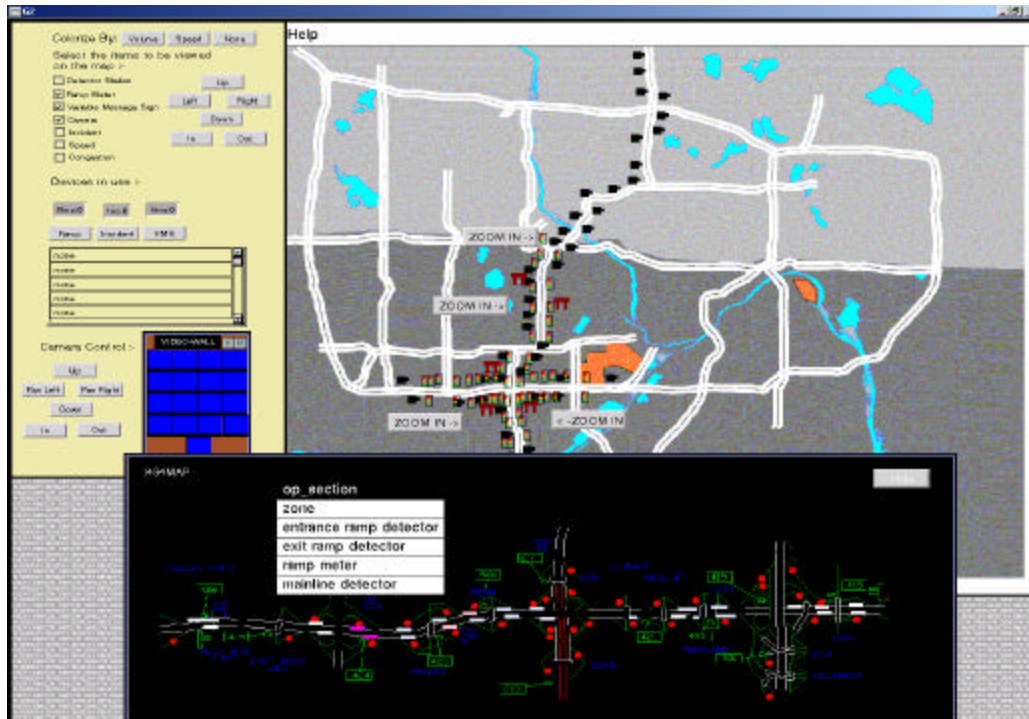


Figure 3.12 Traffic Monitoring

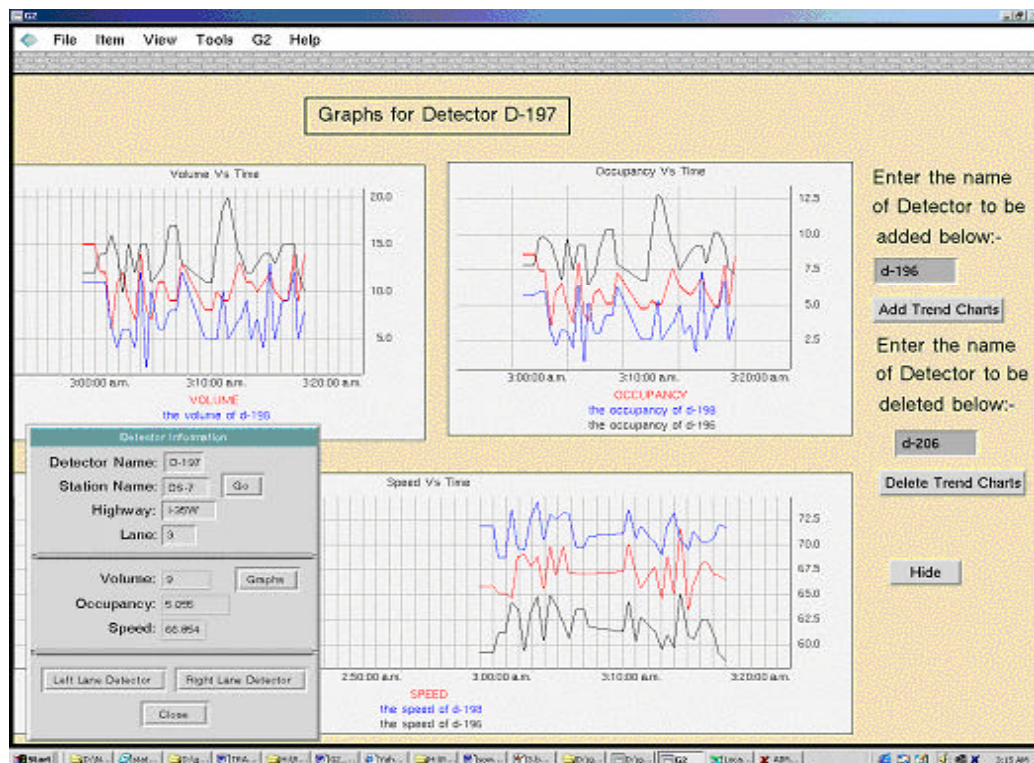


Figure 3.13 Traffic Monitoring through graphs

Furthermore, operators have full control over the devices so that they can manipulate them in order to handle the conditions existing in the freeway. The operators can click on the camera icon and tune their workstation TV monitors to any of those cameras. The map display of the TM module provides an easy way to monitor current traffic conditions by simply observing the color of the freeway sections and by monitoring the volume, occupancy, and speed variation at different locations. It also allows easy access to the CCTV surveillance cameras and all other displayed information (incidents, events etc).

3.3.5.2 Traffic control

TRAMLAB facilitates control of freeway traffic through the objects (ramp meters, detectors, etc.) and procedures existing in it. The system has representations of every detector, ramp meter, variable message sign, traffic control signs, and lane control signals currently existing in the field. Through the extensive procedures and methods incorporated into the system, the operator has full control over the devices existing in the field. The graphical interface facilitates easy monitoring of the status of the devices and also manipulates them whenever needed.

Ramp meter overrides and information dissemination through variable message signs is easily possible through TRAMLAB. Figure 3.14 shows the dialog boxes corresponding to a camera, a ramp meter, and a VMS selected from the display map. The operator can override the ramp meter rates by putting the desired rate in the box "Field Meter rate" and checking the button "Override false" on the ramp meter dialog box. As soon as the operator selects the button "Override false", it changes its text to "Override true". Then by pressing the button "Ok" or "Apply" the operator can easily override the ramp meter rates. Operators can display desired messages in VMS by either selecting predefined messages or by writing a text of the message in the message text box shown in the VMS dialog. TRAMLAB also facilitates implementation of flexible control strategies through the CPI. Through this interface, various ramp metering algorithms could be implemented and evaluated with the aid of the microscopic simulator AIMSUN2. This effectively implies that traffic simulation is integrated into the system.

Thus, TRAMLAB is a fully operational freeway management system and will allow the operators to change ramp-metering strategies in real time and observe the results. Incidents can be simulated and their response and clearance operations are possible.

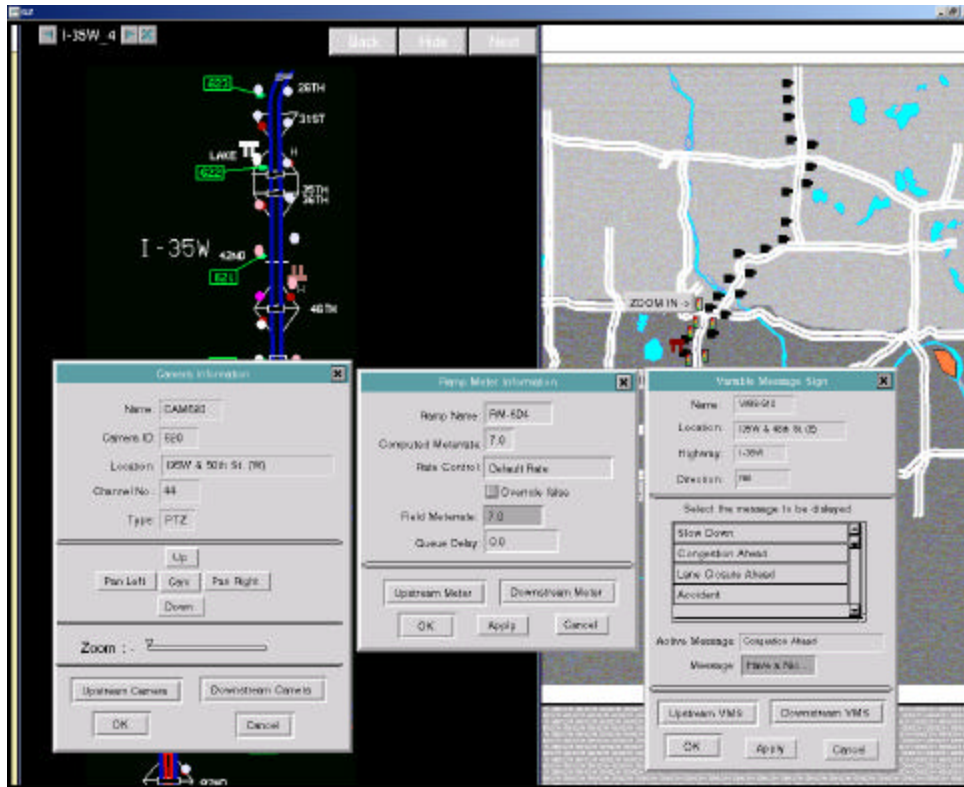


Figure 3.14 Traffic Control

3.3.5.3 Incident detection

The incident detection module in TRAMLAB consists of four automatic incident detection algorithms, interfaces to verify incidents reported through telephone calls, and also an interface to manually report an incident by the operator. The four automatic incident detection algorithms implemented in parallel are the DELOS, the California, Algorithm #7, and Algorithm #8. Incidents can be detected in three possible ways. First, if someone at the incident location calls to report the incident, second, if any of the automatic incident detection algorithms sends an alarm and finally, if operators detect the incident while monitoring the traffic on the monitor or the video wall.

shown in Figure 3.15) to attract the attention of the operator. TRAMLAB is also equipped with an incident report interface that handles numerous reports of the same incident thus avoiding multiple reports. This is achieved by making the data regarding the existing incidents available to multi-users. The operators can also create an incident directly onto the graphical user interface map. The information about the incident can be obtained by just clicking on a button in the user interface.



Figure 3.15 Incident detection

When any of the AID algorithms detect an incident, the system raises an alarm (as

3.3.5.4 Incident verification and Response

The incident verification (IV) module provides the means for the operator to verify the validity of an incident. Mainly the IV module is useful when the report (alarm) for an incident comes from one (or more) of the AID algorithms.

In the event of an alarm, the system is designed to pick the first available operator and switch its monitor to the camera closest to the location of the alarm. Through a graphical interface (as shown in Figure 3.16a) supplied by the system, the operator can control the camera to inspect the area covered or can tune to cameras upstream or downstream from the initial location. In the ITS lab where the system controls the video wall, the top row of monitors (four) are designed to be automatically tuned to the four cameras nearest to the location of the alarm (two upstream and two downstream). At this moment, the operator is asked to verify the alarm. If he/she comes to the conclusion that it was a false alarm, they click the button “False Alarm” in the Incident Alarm window and the system returns all monitors to their normal view and dismisses the alarm.

In the case that the operator verifies the alarm, he/she clicks the button “Incident Confirmed” and the system proceeds by providing an incident form where the operator can enter information about the incident. Finally when all information is collected, an incident object is created in the system and placed at the incident location (on the big map, which covers all the freeways). The operator can access and update the incident information by clicking on the incident icon placed on the map display near the reported location. As soon as the incident is confirmed, a star-shaped object (as shown in the Figure 3.16b) is shown (in the section map) at the incident location. The operator then responds to the incident by applying various combinations of interactive traffic

management strategies, such as rerouting the upstream traffic by displaying incident information on the VMS, overriding the ramp metering rates and others.

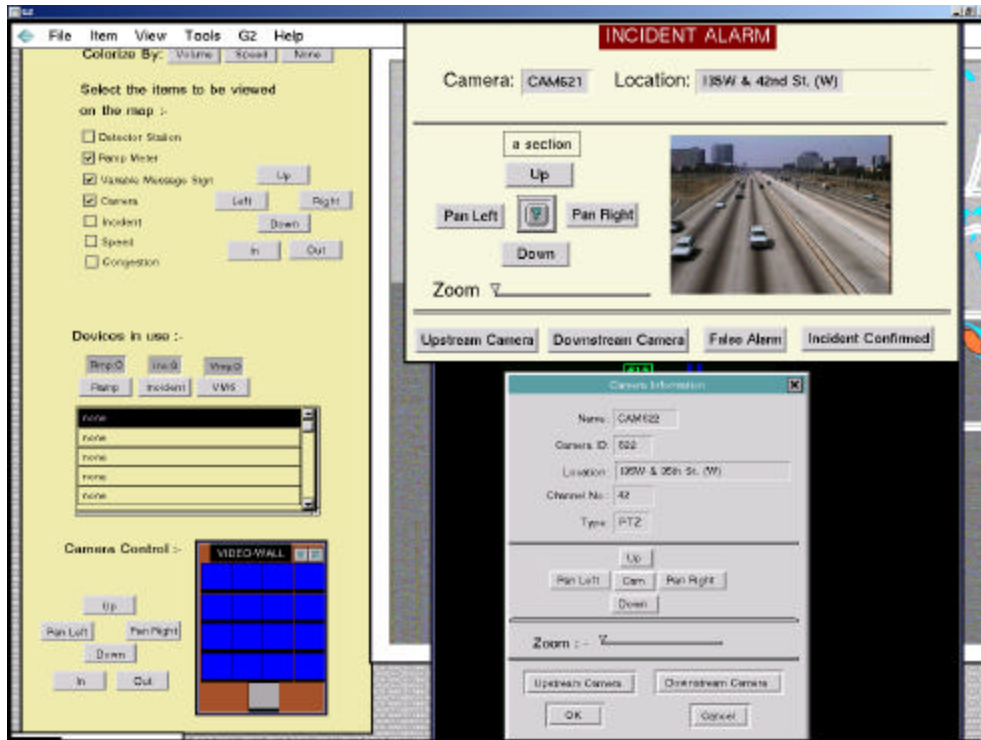


Figure 3.16a Incident Verification

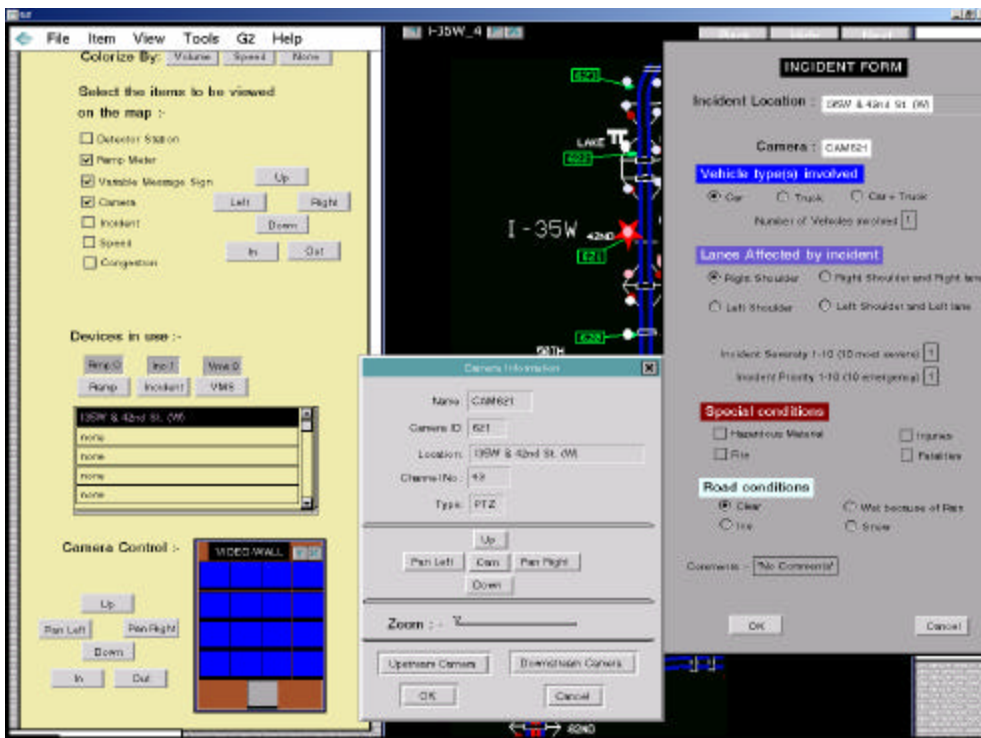


Figure 3.16b Incident Verification

Chapter 4. Graphical User Interface

This chapter explains the usage of the graphical user interface of TRAMLAB. It gives a detailed description of each and every information dialog and explains how to use it for performing various functions. It also provides information about the rules and procedures (code) written to perform the desired function in any dialog. Finally, this chapter explains the differences of using the user interface in different modes.

4.1 TRAFFIC DISPLAY MAP

Figure 4.1 shows the traffic display map on the right side and traffic information window (a workspace) on the left. It is possible to observe the variation in volume, occupancy, and speed on the freeway by clicking the push buttons near the text “Colorize By:” at the top of the workspace. For example, if the operator wishes to observe the variation in speed in different locations on the freeway, he just clicks the button “Speed” and the color of different freeway sections changes depending on their speed values. Since different colors are assigned to different speed ranges, as the speed value increases the intensity of the red color also increases. Similarly, the variation in volume and occupancy can be observed by selecting the respective buttons. The map gets updated every 30 seconds when the detector receives new values. Thus, this interface is unlike other interfaces where it is needed to refresh the system to observe the changes.

The operator has an option of selecting the field items he wants to view on the map by selecting the desired check buttons under the text ‘Select the items to be viewed on the map’. For example, in the above figure, the check buttons camera, ramp meters, and variable message signs were selected and thus their corresponding objects are shown in the display map.

If the operator wishes to view the information about any particular object on the map, he can do so by just clicking the mouse on that object. As soon as the mouse is clicked on any object, the dialog containing the object information is shown.

It is very important for the operator to have full information about all the devices (ramp meters, VMS, etc.) that are in use. For example, at any portion in time, the operator should have full information about all the overridden ramp meters, all the active variable message signs, and all active incidents. The text boxes under the text “Devices in use:” has a default value of Rmp:0, Inc:0, and Vms:0 and the scroll area has a default value “none” in all its rows. These default values show that presently there are no active incidents, none of the ramp meters have been overridden, and none of the VMS have active messages. If any of these events, (for example, incident, ramp overriding and VMS) occur, the text in the edit box changes accordingly as shown in Figure 4.1. For example, if there are two active incidents, the texts in the text box will change to “Inc: 2”. The operator can get the exact location of those incidents by clicking the mouse on the push button “Incident”. The locations corresponding to these events are shown in the scroll area as shown in the Figure 4.1.

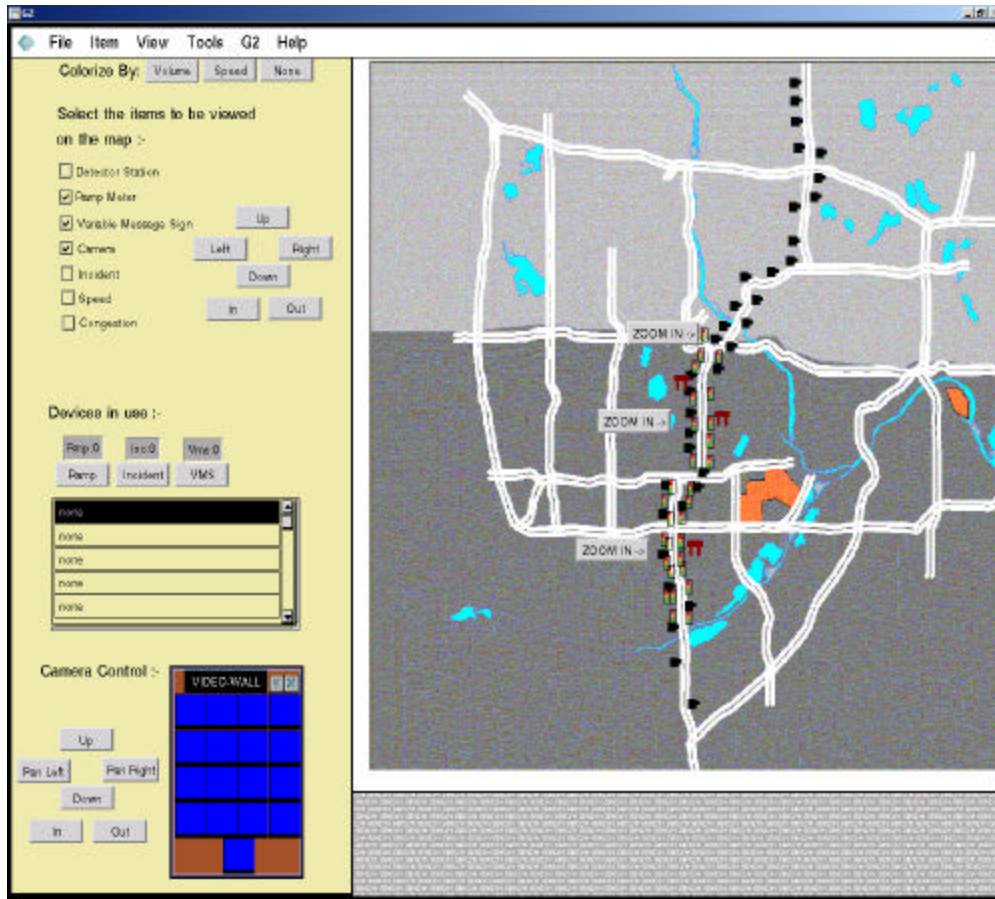


Figure. 4.1 Traffic display map

The video wall shown in the figure 4.1 represents the 16 TV monitors, which are kept as a wall in the ITS lab for monitoring the traffic, and a personal TV monitor, which is kept at the operator's desk. If the operator wishes to tune a particular camera on a particular TV monitor, he/she can do so just by clicking on the desired camera object and then by selecting the TV monitor in the video wall, where he/she wants the camera to be tuned. The operator can select the TV monitor by clicking the mouse on the desired monitor. This action changes the color of the selected TV monitor to gray (as shown in the figure). It is also possible to manipulate the image received by the cameras by tilting the camera up, down, left, or right. This can be done by clicking the desired buttons placed near the video wall. The camera pictures can be zoomed in and out with the help of the buttons "in" and "out" respectively. But presently these functions are not enabled, as control over the field devices is not available. If the operator wishes to have a closer look of the freeway section, he/she can do so by clicking the button "Zoom in" placed on the display map. This action shows a sections map of a small region as shown in the Figure 4.2.

4.2 SECTION MAP

Figure 4.2 shows a part of the freeway I-35W. The exact locations of all the field objects, for example, cameras, ramp meters, variable message signs (VMS), etc., is shown in the map. The green symbols (boxed text) show the locations of the camera and the red circles (on ramps) show the locations of the ramp meters. VMS locations in the southbound direction of the freeway are shown by the symbol pi (Π) whereas the northbound VMS locations are shown by the symbol inverted pi ($_ _ _$). Since it is difficult to show the icons corresponding to each field object (camera, VMS, ramp meters, etc.) on the map, a class name OP_SECTION (operating section) was defined and the access to various field



Figure 4.2 Section map

objects is established through it. The icon of the operating section is kept transparent and the entire freeway is covered by it so that no single section has more than one instance of any of the following: detector station, camera, ramp meter, and variable message sign. The icon of the operating section consists of a circle, a star, a triangle, and a rectangle, all of which are connected to each other. Whenever an incident is detected at a particular section of the freeway, the color of the star portion of the corresponding operating section icon turns red (shown in the Figure 4.2). The small (white) circles shown in the Figure

4.2 are part of the operating section icon. These circles change color depending on the congestion levels, which are calculated based on the occupancy of the detector stations. Different occupancy ranges are defined and colors are assigned to these ranges such that as the occupancy value becomes higher, the intensity of the red color increases.

If the operator desires to view the information about the field objects present at a particular location of the freeway, he/she clicks on that location on the map. As soon as the operator clicks on the desired section, he/she is shown a menu showing all the field objects present in that section (as shown in the Figure 4.2). The operator then can view the object's information by selecting the menu choices. The table below shows what procedures are invoked when the operator selects options from the operating section menu choices:

Menu Choice	Procedure Invoked	Refer*
1. Mainline Detector	show-opsection-user-menu	1.1
2. Ramp Meter	show-opsection-user-menu	1.1
3. Camera	show-opsection-user-menu	1.1
4. Entrance Ramp Detector	show-opsection-user-menu	1.1
5. Adjacent Ent. Ramp Detector	show-opsection-user-menu	1.1
6. Exit Ramp Detector	show-opsection-user-menu	1.1
7. VMS	show-opsection-user-menu	1.1
8. Zone	show-opsection-user-menu	1.1
9. Adjacent Ramp Meter	show-opsection-user-menu	1.1
10. Incident	show-opsection-incident-user-menu	1.2

* - The number under the column "Refer" refers to sections of Appendix A.I.

4.3 DETECTOR STATION INFORMATION DIALOG

When the operator chooses the option "Mainline Detector" from the menu choices of the operating section, the mainline detector station dialog box corresponding to that operating section is shown (as shown in Figure 4.3).

The top half portion of the detector station information dialog gives information about the name, location, type, and status of the detector station. The status button shows whether the selected detector station is working fine or malfunctioning. If the operator wishes to view the information of the upstream detector station, he/she clicks the button "Next Upstream Station". As soon as the operator clicks this button, the presently open station dialog gets closed and the dialog corresponding to the next upstream detector station is shown. Similarly, when the operator wishes to view the information of the downstream detector station, he/she clicks the button "Next Downstream Station" to view the desired information.

The lower half of the detector station dialog shows the present volume, occupancy, and speed of the detector station and the detectors belonging to the detector station. The volume, occupancy and speed values get updated every 30 seconds. If the operator wishes to view the information of a particular detector in this detector station, he/she clicks the button '1' or '2' or '3' or '4' depending on which lane detector information he/she wants to view. If the operator wants to close the dialog, he/she can either click the button "Close" or the button "X" at the upper right corner of the dialog.

The list below shows the procedures that are invoked when the operator selects any button from the station dialog box.

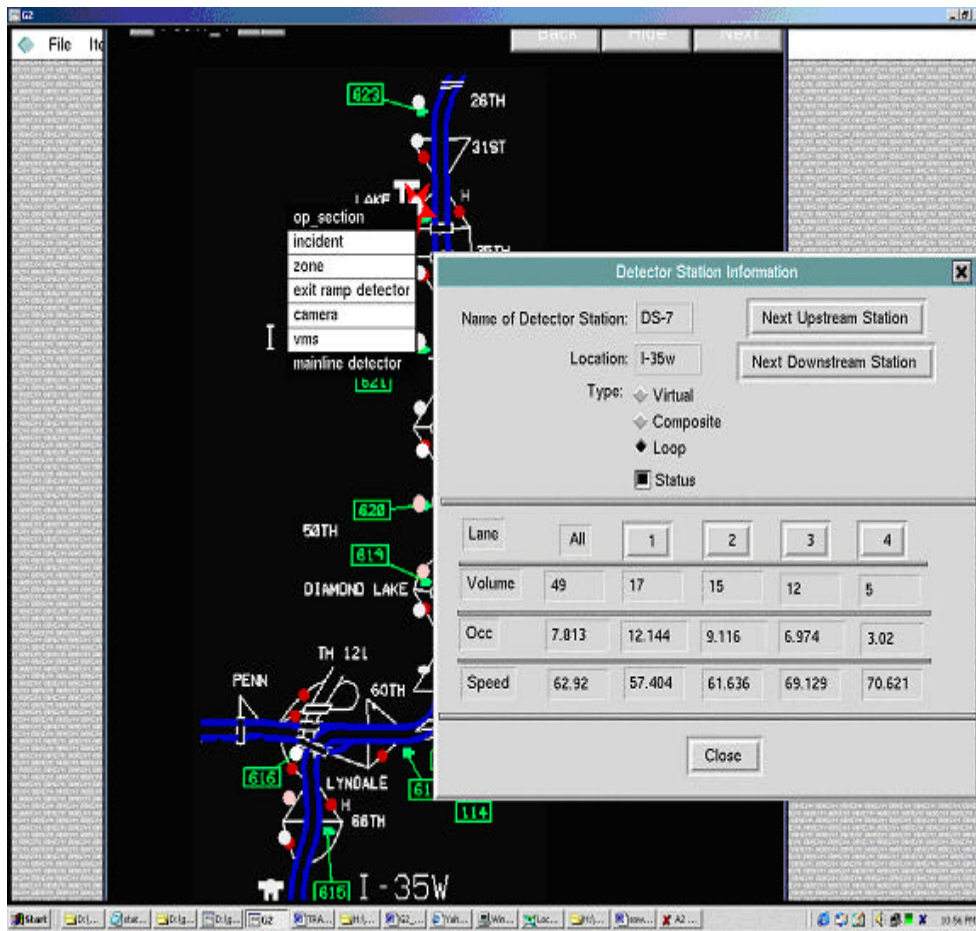


Figure 4.3 Detector station information

- | | | |
|--------------------------------|---------------------------------------|-----|
| 1. Next Upstream Station | upstream_or_downstream_station_dialog | 1.3 |
| 1. Next Downstream Station | upstream_or_downstream_station_dialog | 1.3 |
| 2. " 1 " (Lane One Detector) | Station_to_Det | 1.4 |
| 3. " 2 " (Lane Two Detector) | Station_to_Det | 1.4 |
| 4. " 3 " (Lane Three Detector) | Station_to_Det | 1.4 |
| 5. " 4 " (Lane Four Detector) | Station_to_Det | 1.4 |

For getting the values of the detectors in the detector station dialog, the procedure named "Detector_info_in_station_dialog" is invoked. Refer : - 1.5

For updating the detector station dialog, a whenever rule is invoked, which in turn invokes a procedure named update_dialog: - 1.6

4.4 DETECTOR INFORMATION DIALOG

When the operator selects the choice “Entrance Ramp Detector” or “Adjacent Entrance Ramp Detector” or “Exit Ramp Detector” amongst the available choices in the operating section menu or when he/she clicks the button “1” or “2” or “3” or “4” (lane number) on the detector station dialog, the detector information dialog of the corresponding detector is shown (as shown in the Figure 4.4).

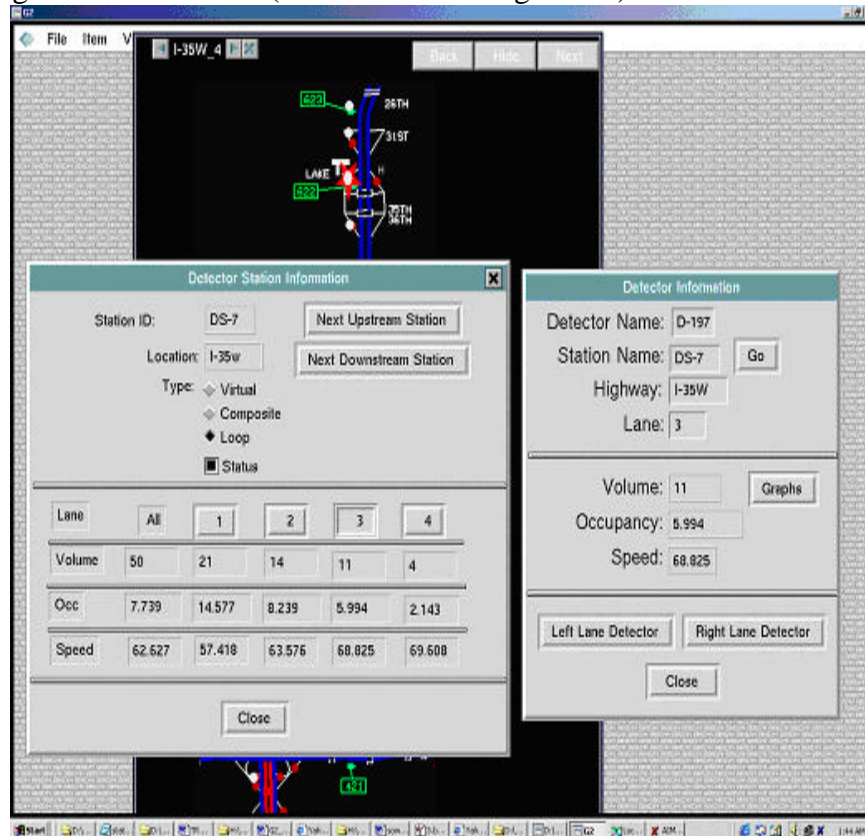


Figure 4.4 Detector information dialog

The upper half of the detector information dialog gives the information about detector’s name, corresponding detector station name, the name of the highway/freeway on which this detector lies, and its lane number. The button “Go” opens the detector station information dialog. Therefore, if the operator wishes to view the information of the detector station to which this detector belong, he/she clicks the button “Go”. The other half-portion of the detector information dialog shows the present volume, occupancy, and speed, which gets updated every 30 seconds. If the operator desires to view the information of the other detectors belonging to the same detector station, he/she clicks the button “Left Lane Detector” or “Right Lane Detector”; depending on which lane detector information he/she wants to view. Clicking the button “Left Lane Detector” or “Right Lane Detector” closes the open detector dialog and opens the dialog of the desired detector.

If the operator wants to view the variation of volume, occupancy, and speed with time, he/she clicks the button “Graphs” on the detector dialog. As soon as the button “Graphs” is clicked, a workspace showing the volume, occupancy, and speed variation shows up (as shown in the Figure 4.5). These graphs are known as trend charts.

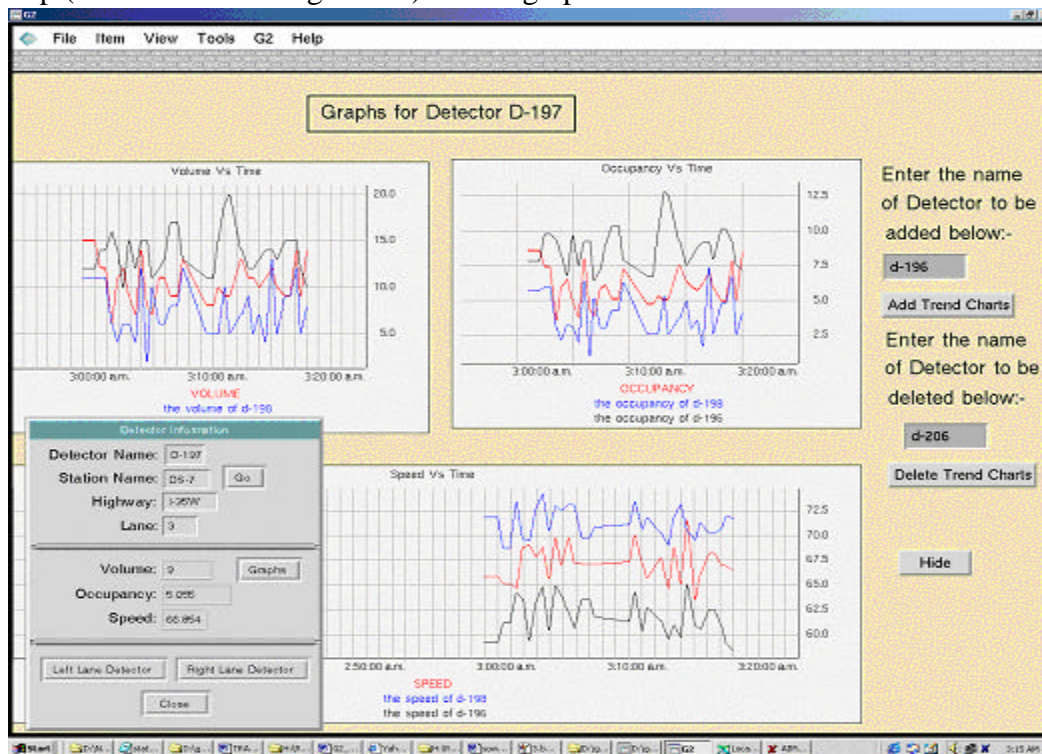


Figure 4.5 Detector graphs

These trend charts get updated whenever the detector receives new values of volume, occupancy and speed (every 30 seconds). Initially, only the red color graph, which corresponds to the volume, occupancy, and speed of the currently open detector, is shown in the trend charts. Now, if the operator wishes to compare the volume, occupancy, and speed variation of this detector with some other detector, he enters the desired detector name in the edit box shown at the right top in the Figure 4.5 and clicks the button “Add Trend Charts”. Each time a detector is added, a new plot with a different color is created (as shown in Figure 4.5). The color and text of the legend shown at the base of each trend chart makes clear which trend chart belongs to which detector. Similarly, if the operator wishes to remove plots of any detector, he/she puts the detector name in the edit box placed just above the button “Delete Trend Charts” and then he/she clicks the button “Delete Trend Charts”. The button “Hide” hides the graph workspace. The list below shows the procedures that are invoked when the operator selects any button from the detector information dialog box and the graph workspace: -

BUTTON SELECTED	PROCEDURE INVOKED	REFER
1 Go	Go1	1.7
2. Graphs	Graphs	1.8
3. Left Lane Detector	Adjacent1	1.9
4. Right Lane Detector	Adjacent 2	1.10
5. Add Trend Charts	Add_Trend-Charts	1.11
6. Delete Trend Charts	Delete_Trend-Charts	1.12
7. Hide	Hide-Chart-Wks	1.13

4.5 RAMP METER INFORMATION DIALOG

As shown in the Figure 4.2, the small circles (red color) show the locations of the ramp meters in the freeway network. If the operator wishes to view the ramp meter information of a particular ramp meter, he/she clicks on the ramp meter location (near the red circle) on the map. The operating section menu shows ramp meter as one of its menu choices. When the operator selects the choice “Ramp Meter” from the operating section menu, he/she is shown the ramp meter information dialog (as shown in the Figure 4.6).

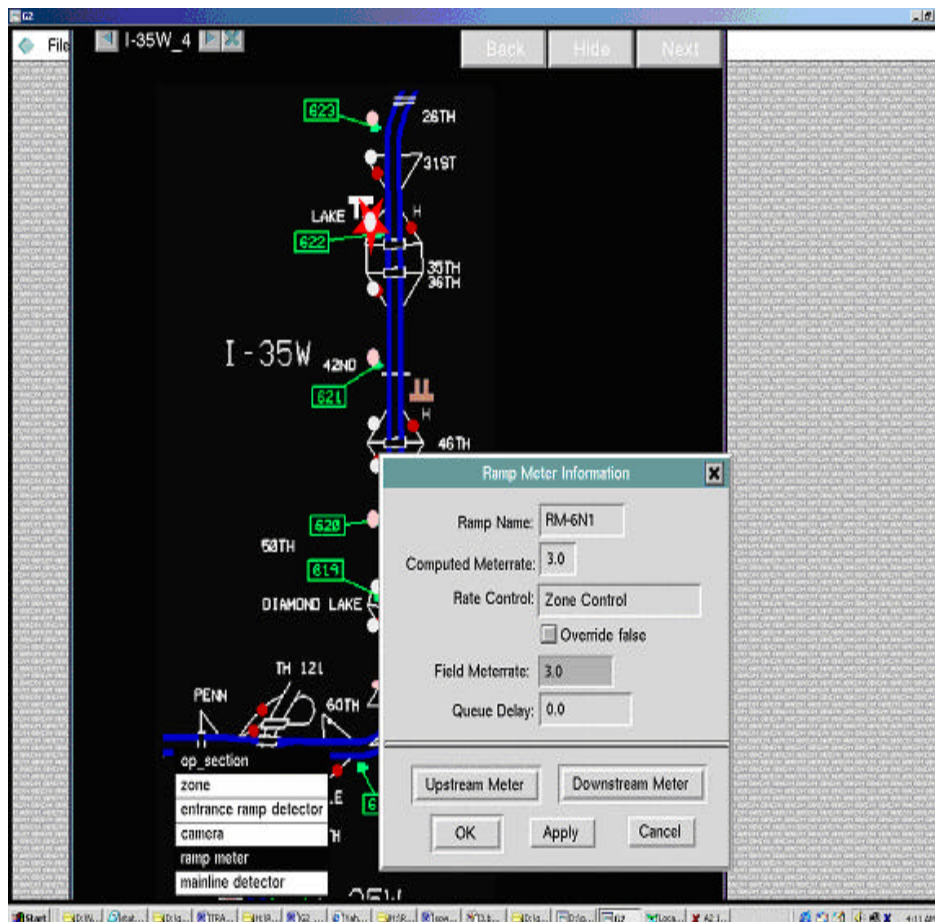


Figure 4.6 Ramp meter information dialog

The ramp meter dialog gives the information about the name of the ramp meter, computed meter rate, field meter rate, presently dominating control, and override. The computed meter rate is the rate calculated using ramp control algorithm (presently it is using Mn/DOT ramp control strategy), whereas, the field meter rate is the actual rate of the ramp meters on the field. The field meter rate is same as the computed meter rate if the ramp meter is not overridden. The text box corresponding to the 'Rate Control' shows the name of the presently dominating control ("Zone Control", "Occupancy Control" or default rate). The operator can override the ramp meter rate by entering the new meter rate in the "Field Meter rate" edit box and clicking the check box "Override False" (so as to make override true). If the operator wants to remove the override, he/she can do so by clicking the button 'override true' and pressing "OK". The buttons "Upstream Meters" and "Downstream Meters" are used to view the information of the ramp meters upstream or downstream to the presently open ramp meter. The table below shows the procedures that are invoked when the operator selects any button from the Ramp Meter information dialog box :

BUTTON SELECTED	PROCEDURE INVOKED	REFER
1. Upstream Meter	upstream_or_downstream_ramp_dialog	1.14
2. Downstream Meter	upstream_or_downstream_ramp_dialog	1.14
3. Override false and OK (for overriding)	A whenever rule and remote procedure named set_override_rate	1.15
4. Override true and OK (for removing override)	A whenever rule and remote procedure named set_override_rate	1.16

4.7 CAMERA INFORMATION DIALOG

The green symbol (numbered boxes) on the freeway map (Figure 4.2) show the locations of the camera in the freeway network. If the operator wishes to view a particular camera, he clicks at the camera location (near the green symbol) on the map. As soon as the operator clicks on the map, a menu choice for the corresponding operating section shows up. Now this menu choice has "camera" as one of its choices. When the operator selects 'Camera' from the menu, he is shown the Camera Information Dialog of that camera (shown in the fig 4.7) and his personal TV gets tuned to this camera.

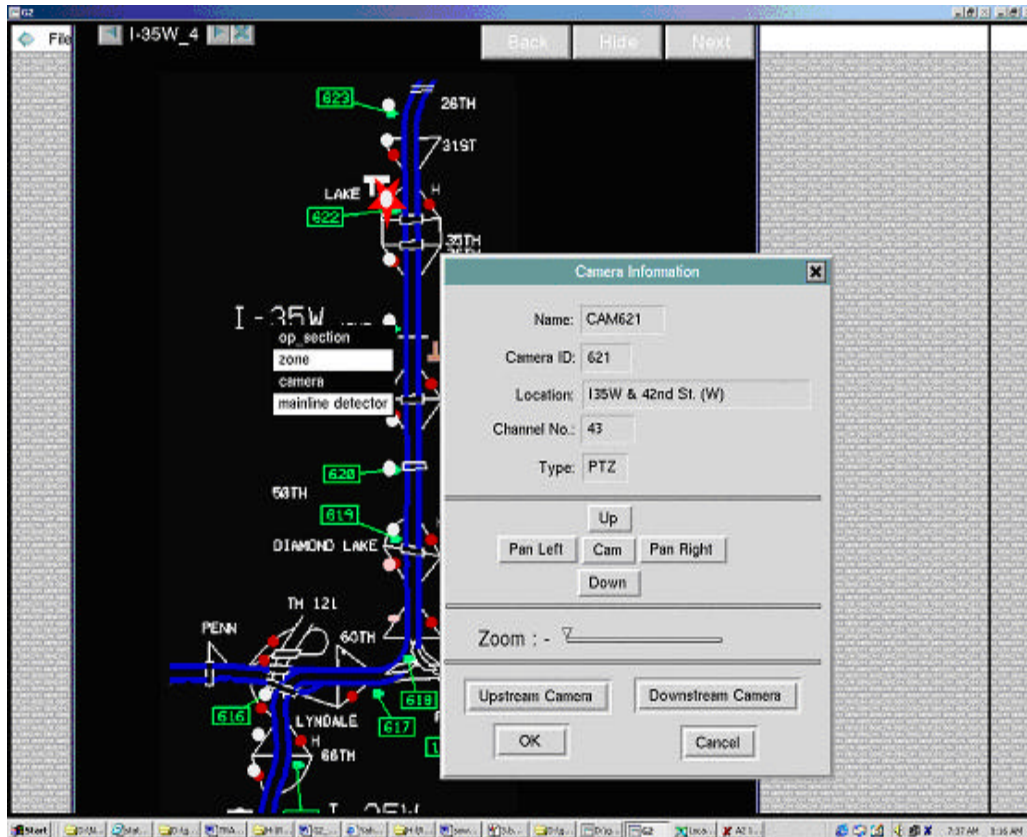


Figure 4.7 Camera information dialog

The camera information dialog gives the information about the name of the camera, camera ID, channel Number, location, and type of the camera. Since the operator's personal TV gets tuned to the camera, the operator now can view the pictures from the camera on his/her TV. The middle portion of the camera dialog helps the operator to have a better view from the camera. The operator can adjust the camera according to his/her comfort by clicking the button 'Up', 'Down', 'Pan Left', 'Pan Right' and by zooming in or out using the slider (as shown in the fig 4.7). If the operator selects the button 'Upstream Camera' or 'Downstream Camera', the presently open camera dialog gets closed, the camera dialog for the selected camera is shown and the operator's personal TV gets tuned to this camera. The list below shows which procedures are invoked when any button on the camera dialog is clicked:

- | | | |
|----------------------|--------------------------------------|------|
| 1. Upstream Camera | upstream_or_downstream_camera_dialog | 1.17 |
| 2. Downstream Camera | upstream_or_downstream_camera_dialog | 1.17 |

4.8 VARIABLE MESSAGE SIGN DIALOG

When the operator selects the option 'VMS' from the menu choice of the operating section, a Variable Message Sign dialog for that VMS is shown (figure 4.8). This dialog gives the information about the name of VMS, location, highway/freeway name, and the direction. If the operator wants to display any predefined message, he clicks on the message in the scroll bar and selects 'OK'. But if the operator wants to display a new message, he can write the message in the edit box besides the text 'Message' on the

dialog box. The text box corresponding to ‘Active Message’ shows the text of the presently selected message, if any. If the operator selects the button ‘Upstream VMS’ or ‘Downstream VMS’, the presently open VMS dialog gets closed and the VMS dialog for the selected VMS is shown.

The list below shows which procedures are invoked when any button on the VMS dialog is clicked:

1. Upstream VMS upstream_or_downstream_vms_dialog 1.18
2. Downstream VMS upstream_or_downstream_vms_dialog 1.18
3. When any message is selected or entered in the message box and when the OK button is pressed a whenever rule is invoked, which in turn invokes a remote procedure named set_vms_msg. Refer : - 1.17

4.9 METERING ZONE INFORMATION DIALOG

The information dialog for any metering zone can be seen by clicking the mouse on any section in that zone and then selecting the choice ‘Zone’ among the other menu choices (shown in the figure 4.9). The zone information dialog gives information about the name of the metering zone, highway/freeway name, direction, the capacity of the zone, smoothing factor, the bottleneck capacity of the zone, ramp turn-on parameter, ramp turn-off parameter, and the volume thresholds. All these attributes are specifically defined for each zone.

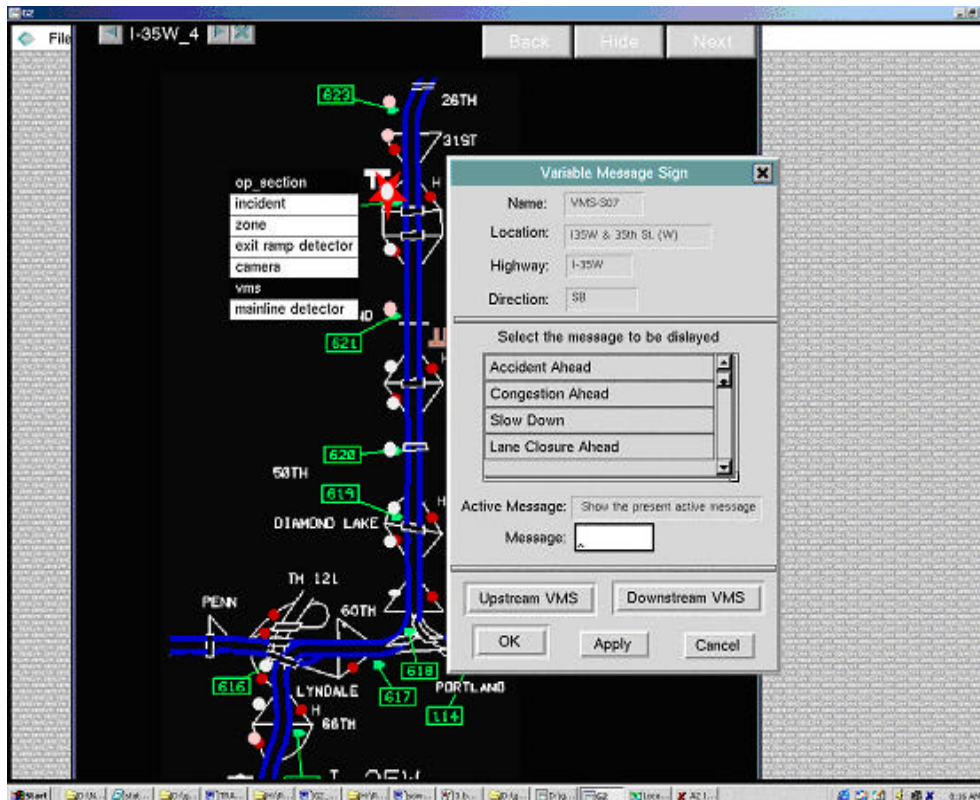


Figure 4.8 Variable Message Sign dialog

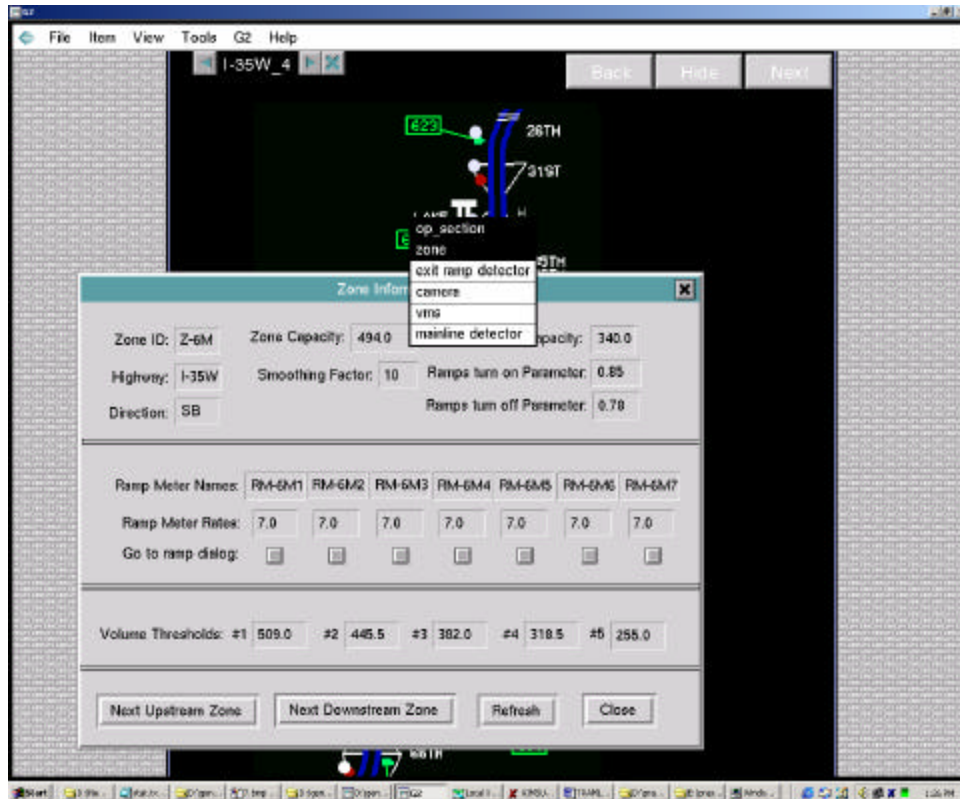


Fig 4.9 Metering zone information dialog

The center half of the zone information dialog shows the names of the ramp meters belonging to that zone. The ramp meter rates shown are the field meter rates of each ramp meter in that zone. If the operator desires to view the information about a particular ramp meter in the zone, he clicks the 'Go to Ramp Dialog' button of that particular ramp meter (as shown in figure 4.10).



Figure 4.10 Zone and ramp dialog

The operator can view the upstream or downstream zone information by clicking the buttons ‘Upstream Zone’ or ‘Downstream Zone’. The refresh button gets the updated rates of the ramp meters. The list below shows which procedures are invoked when any button on the zone dialog is clicked:

- | | | |
|-------------------------|------------------------------------|------|
| 1. Next Upstream Zone | upstream_or_downstream_zone_dialog | 1.19 |
| 2. Next Downstream Zone | upstream_or_downstream_zone_dialog | 1.19 |
| 3. Go to Ramp Dialog | zone_to_meter | 1.20 |
| 4. Refresh | Refresh | 1.21 |

4.10 INCIDENT INFORMATION DIALOG

Whenever any incident detection algorithm detects an incident, the incident notification workspace is shown (as shown in the figure 4.11) and the operator’s personal TV gets tuned to the camera nearest to the incident location.

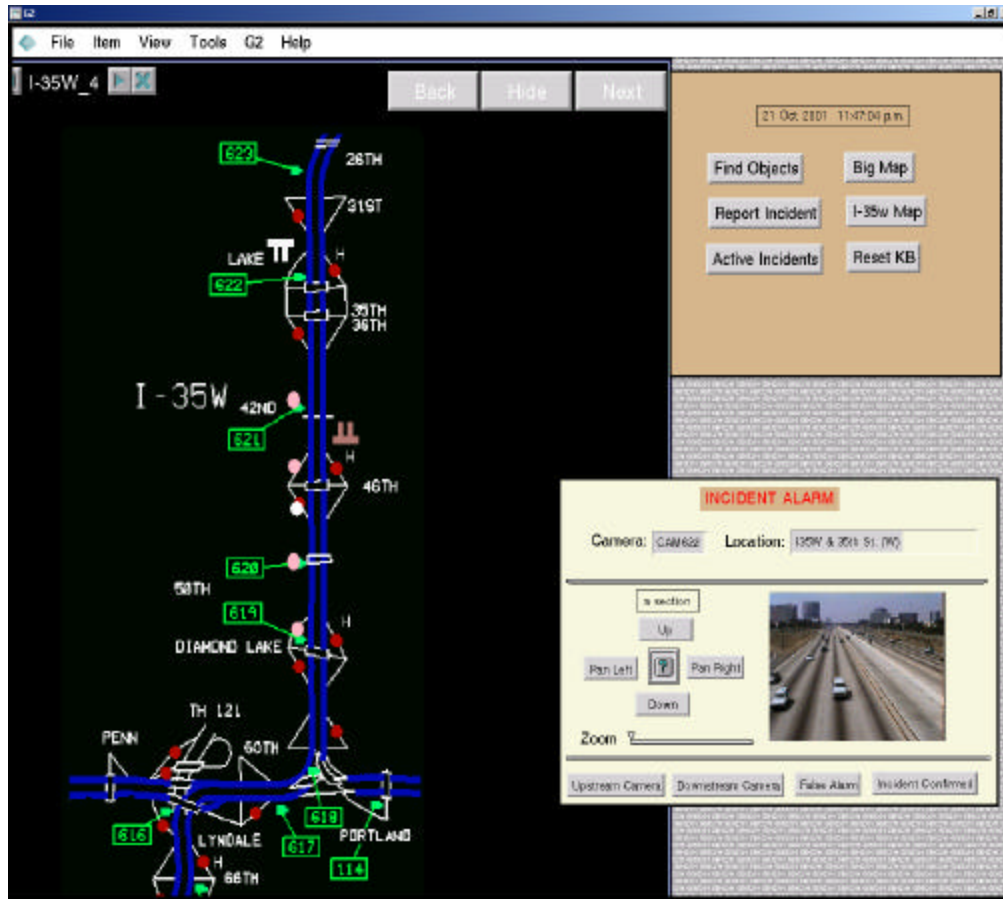


Figure 4.11 Incident notification workspace

The incident notification workspace consists of the name of nearest camera and the location of the incident. The operator can have proper view of the incident location by panning the camera up, down, left and right and also by zooming in and out. This helps the operator determine whether the incident has really occurred or not. If the operator desires to view the location upstream or downstream of the present location, he clicks the buttons, 'Upstream Camera' and 'Downstream Camera'. When the operator clicks these buttons he is shown the camera dialog information of the upstream or downstream camera, depending on the button he clicked. Thus, the operator confirms whether the incident alarm is false or not. If he doesn't encounter any incident, he clicks the button 'False Alarm' and the incident notification workspace gets hidden. But if the operator sees an incident at that location, he clicks the button 'Incident Confirmed'.

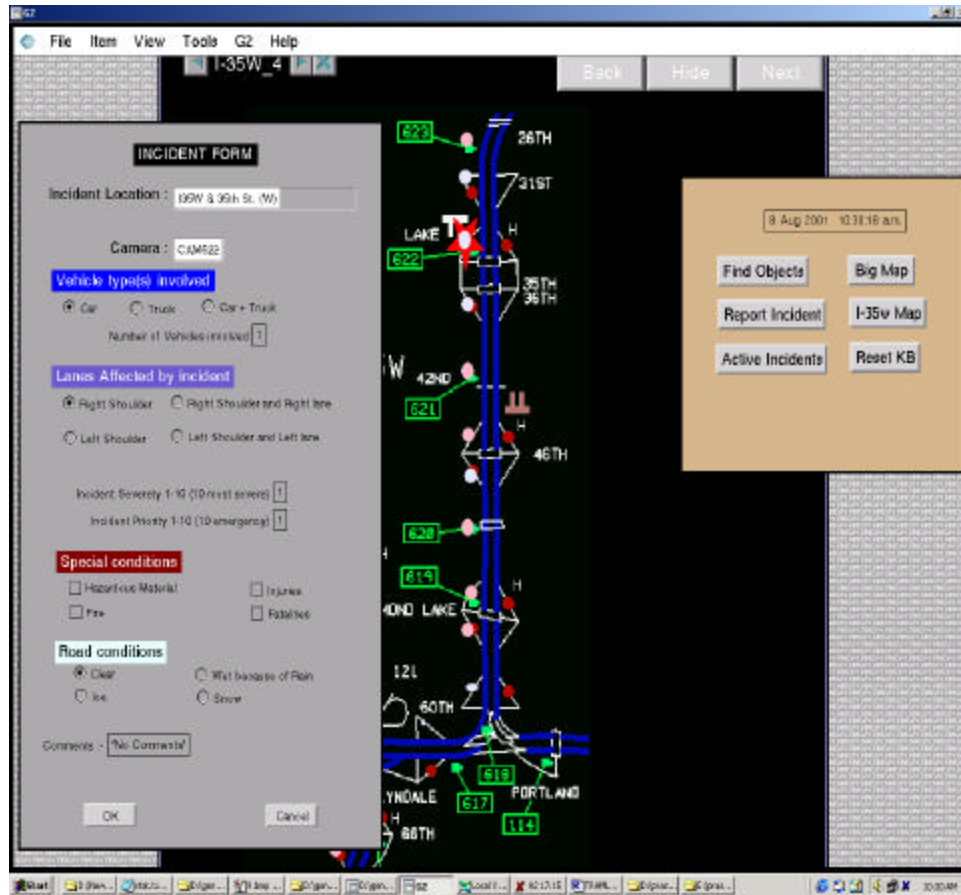


Figure 4.12 Incident form

When the operator clicks the button ‘Incident Confirmed’, the star icon-color of the corresponding operating section turns red and the operator is given the incident form to fill in the incident information (shown in the figure 4.12). The incident form consists of the name of the camera (nearest to the incident location) and the location of the incident. The operator selects the appropriate button for the vehicle type involved, lanes affected by incident, special conditions, and road conditions. Depending on the severity of the incident, the operator filling a number on the scale of 1-10. Similarly, the priority number shows what priority has to be given to this incident. If the operator wants to add some comments about the incident, he can do so by writing the text in the ‘Comments’ type-in-box. After filling out the incident form, the operator clicks ‘OK’ and all the information is stored in the attribute named ‘Incident’ of the corresponding operating section.

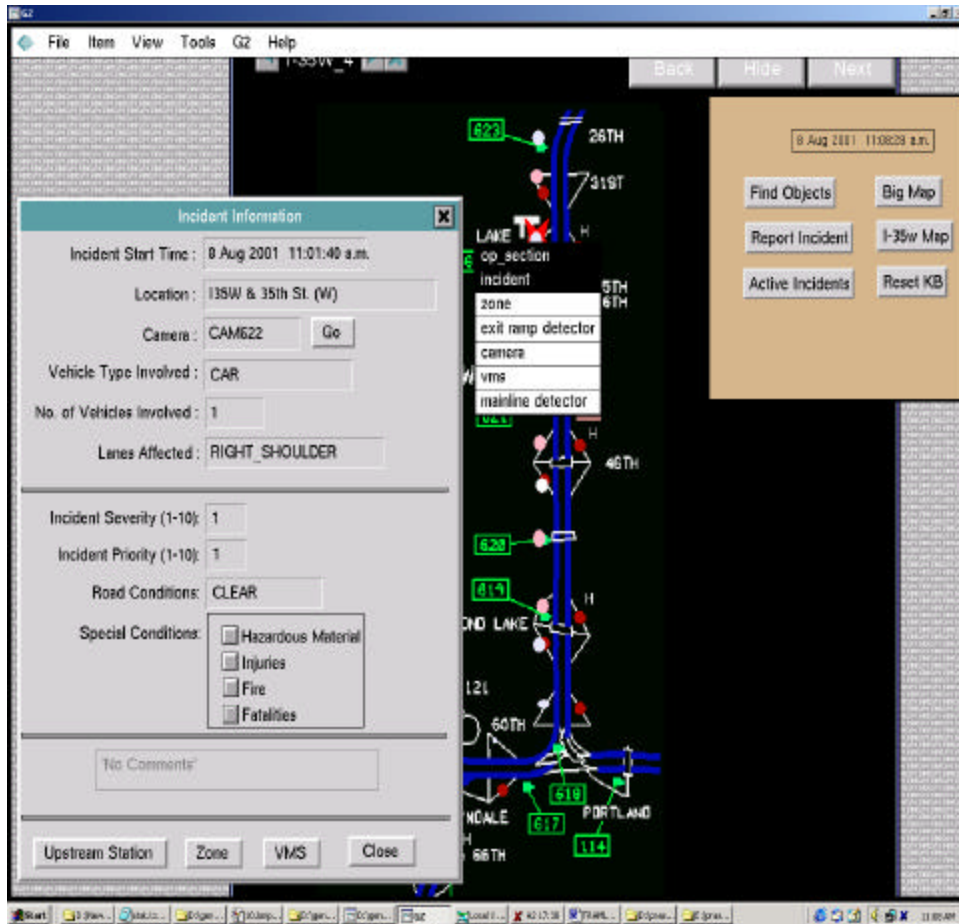


Figure 4.13 Incident information dialog

The color of the star remains red till the incident is active. At any time if the operator wants the incident information, he clicks on the corresponding operating sections and selects the choice 'Incident' from the operating section menu. As soon as 'incident' is selected, the Incident information dialog is shown (as shown in the Figure 4.13).

The incident information dialog shows all the information filled in by the operator in the incident form. It gives the information about the incident start time, location of the incident, name of the camera nearest to the incident location, vehicle type involved, number of vehicles involved, name of the lane affected due to the incident, incident severity, incident priority, road conditions, and special conditions. The incident dialog also shows the comments entered by the operator on the incident form. The button 'Go' near the camera name opens the camera information dialog. If the operator wishes to view the information of the detector station upstream to the incident section, he can do so by clicking the button 'Upstream Station'. Similarly, the button 'Zone' and 'VMS' opens the zone information dialog and VMS information dialog respectively. The list shown below show which procedures are invoked when any button on the incident-notification workspace or Incident form or incident information dialog is clicked: -

1. Upstream Camera (IN)	Notification-to-camera	1.22
2. Downstream Camera (IN)	Notification-to-camera	1.22
3. False Alarm (IN)	False-Alarm	1.23
4. Incident Confirmed (IN)	Incident-Verified	1.24
5. 'OK' (Incident Form)	Incident-form-info	1.25
6. 'Upstream Station' (IID)	Upstreamstation-from-incidentdialog	1.26
7 'Go' (IID)	Incident-cam	1.27
8 'Zone' (IID)	Upstreamstation-from-incidentdialog	1.26
9 'VMS' (IID)	Upstreamstation-from-incidentdialog	1.26

where, IN – Incident Notification Workspace and IID- Incident Information Dialog.

4.11 REPORTING OF INCIDENTS

It is sometimes necessary to create incidents artificially to generate real life traffic scenarios for training or testing the operators. There are two ways to create incidents artificially:

- 1) By pressing Ctrl + Alt + clicking any mouse button on the operating section icon.
 - 2) By pressing the button 'Report Incident' (as shown in Figure 4.14).
- **By clicking on the operating section:** As soon as the operator presses Ctrl + Alt + any mouse button in any operating section, the incident notification workspace is shown to the operator and then the normal incident verification process continues.

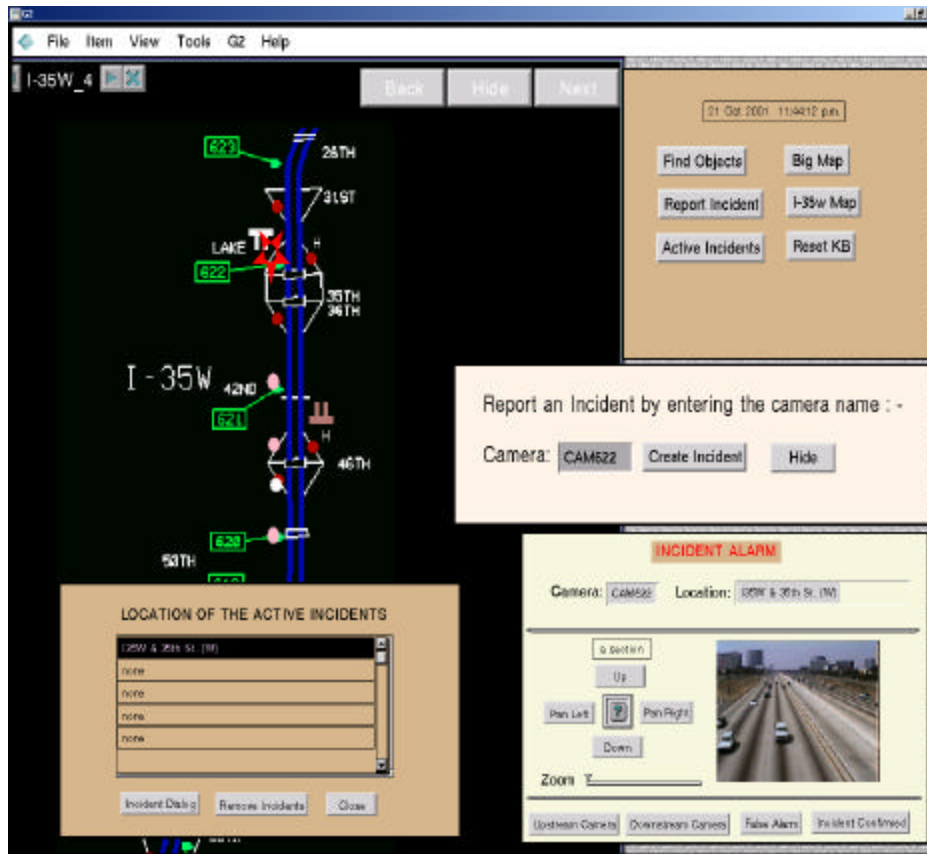


Figure 4.14 Report Incident and Active Incidents

- **By pressing ‘Report Incident’ button:** When the operator presses the button ‘Report Incident’, a workspace named Report-Incident shows. In this workspace the operator enters the camera name and creates an incident at the camera location. After the operator enters the camera name and presses ‘Create Incident’, he is shown the incident notification workspace and the normal incident verification process continues.

4.12 ACTIVE INCIDENTS

Sometimes the operator needs to view all the active incidents. When the operator clicks on the button ‘Active Incident’, a workspace showing the location of all the active incidents is shown (as shown at the lower left corner of the Figure 4.15).

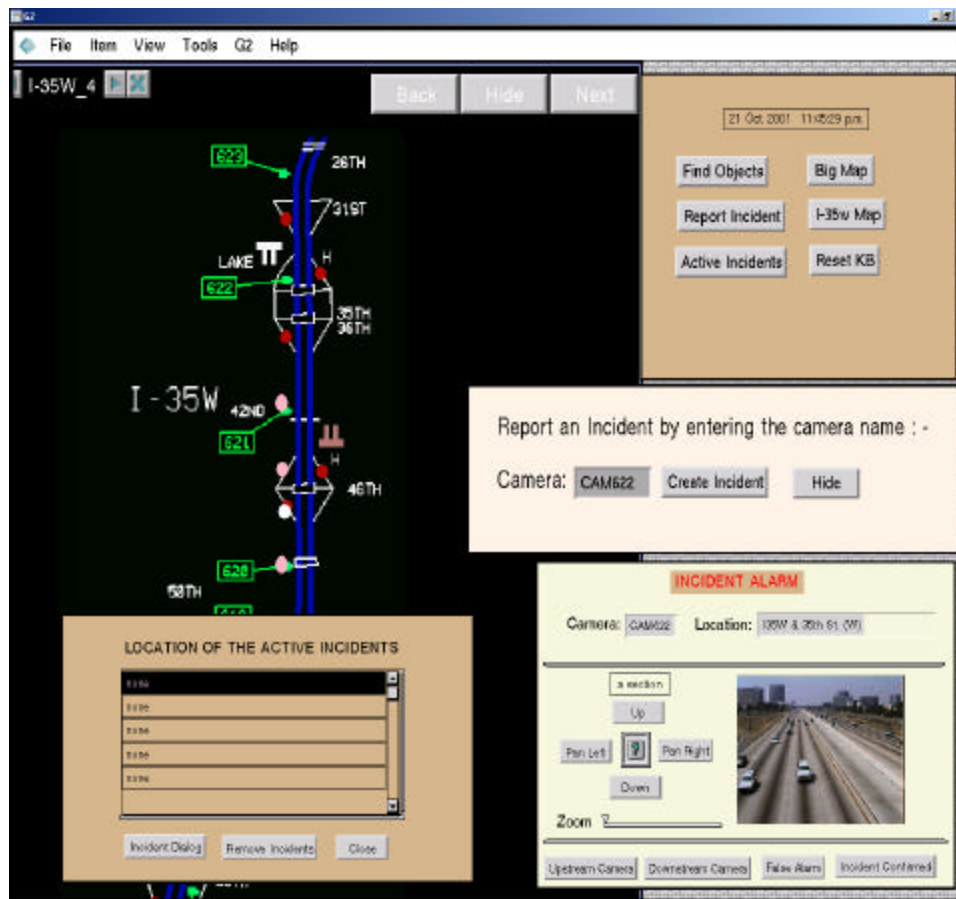


Figure 4.15 Remove incident

If the operator desires to view the information about a particular incident, he selects the location from the scroll area and clicks the button ‘Incident Dialog’. ‘Incident Dialog’ opens the incident information dialog of the incident at the selected location. Sometimes, it is also necessary to artificially remove the incident from a particular location. This is done by selecting the incident location from the scroll area and clicking the button ‘Remove Incident’. When the button ‘Remove Incident’ is clicked, the incident gets removed. The star icon-color of the incident associated with the operating section icon turns from red to transparent and the text of the location of the incident changes to

none (this can be observed by comparing the Figure 4.14 and Figure 4.15). The list shown below show which procedures are invoked when any button on the report incident workspace or active-incident workspace is clicked:

- | | | |
|--|-----------------|------|
| 1. Remove Incidents (AI) | remove_incident | 1.28 |
| 2. Incident Dialog (AI) | incident-camera | 1.29 |
| 3. Ctrl + Alt + any mouse
button on any opsection | createincident | 1.30 |
| 4. Create Incident | report_incident | 1.31 |
- # Action button 'Report Incident' shows the workspace named Report-Incident.
 # Action button 'Active Incident' shows the workspace named Remove-Incident.

4.13 USER MODES

The system has been set in such a way that G2 starts in operator mode. G2 has an option of configuring the user interface such that when used in different modes it performs different functions for the same action. Since it is not necessary for a user in operator mode to see the procedures and rules made for executing different functions, these procedures and rules are kept in workspaces that are not accessible in operator mode. Some of these differences of different modes are as follows:

- **Main Menu:** The Main menu in the administrator mode has lots of options in its menu choices (as shown in the Figure 4.16), for example, creating a new workspace

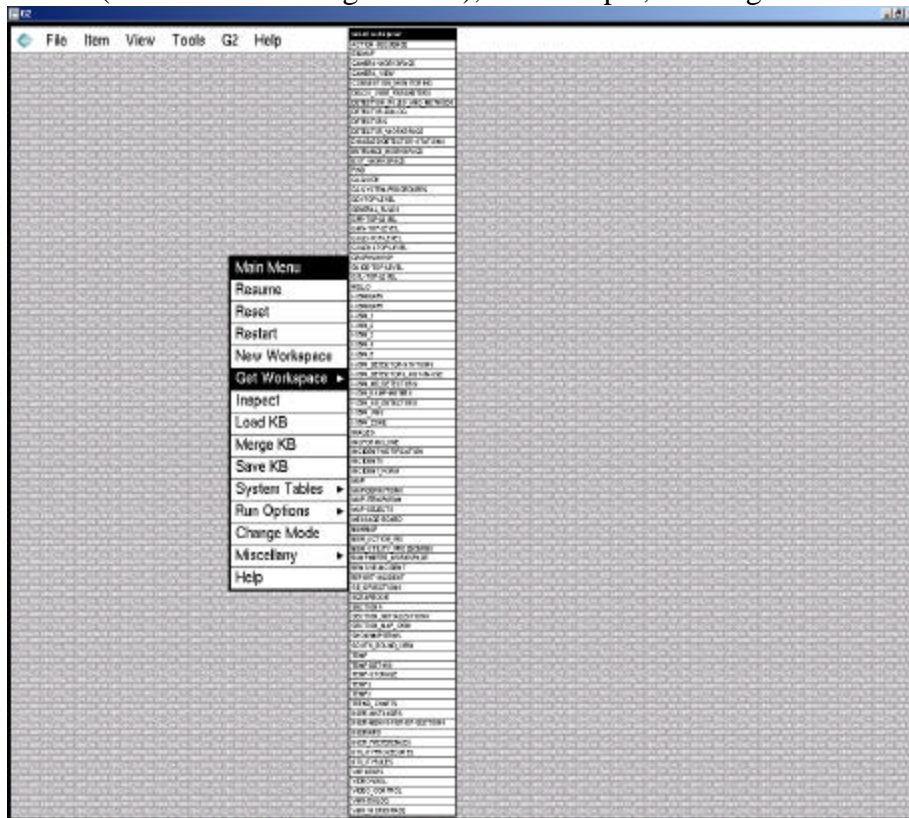


Figure 4.16 Menu choices in administrator mode

to show the presently available workspaces, to change the present mode to some other mode, etc. These options are not included in the operator's mode for security reasons. This stops the user in the operator mode from making any changes in the building blocks of the user interfaces and changing his working mode from operator to some other mode. In the operator mode, only the necessary choices are kept in the main menu, i.e., Pause, Reset, Restart, Save KB, Resume, Load KB, and Merge KB.

- **KB Workspace Menu:**

When the user is in the administrator mode, clicking any mouse button on any workspace shows the table of that kb-workspace. This table consists of many choices (as shown in Figure 4.16), which enables the user to perform necessary functions. For example, the user in administrator mode can clone the workspace, hide the workspace, delete the workspace, create a new object on the workspace, etc. All these functions are necessary when working in administrator mode. The user interface for the kb-workspace is configured in such a way that when in operator mode, clicking any mouse button on any workspace does nothing.

- **Objects**

When the user is using the system in administrator mode, clicking the mouse button on any object (detector, detector station, ramp meter, VMS, etc.) shows the table for that corresponding object (this was explained in Chapter 3). The user interface is configured such that when in operator mode, clicking the mouse button on any object does different actions for different objects, as explained below:

1. **Operating Section:** When an operating section is clicked by the mouse button, it shows a menu containing the names of all the objects present under that operating section. For example, if that operating section has a detector station, ramp meter, camera, and a VMS, then it shows a menu containing these names (as shown in Figure 4.17)
2. **Ramp Meter:** If a ramp meter is clicked by any mouse button or if the option 'ramp meter' is selected from operating section menu, the ramp meter information dialog of the corresponding ramp meter is shown (as shown in Figure 4.18).
3. **Variable Message Sign:** If a variable-message-sign is clicked by any mouse button or if the option 'VMS' is selected from the operating section menu, then VMS information dialog of the corresponding VMS object is shown (as shown in Figure 4.18).
4. **Incident:** If an incident icon is clicked by any mouse button or if the option 'incident' is selected from the operating section menu, the incident information dialog of the corresponding incident is shown (as shown in the Figure 4.17).
5. **Camera:** If a camera icon is clicked by any mouse button or if the option 'camera' is selected from the operating section menu, the camera information dialog of the corresponding camera is shown (shown in Figures 4.17 and 4.18).

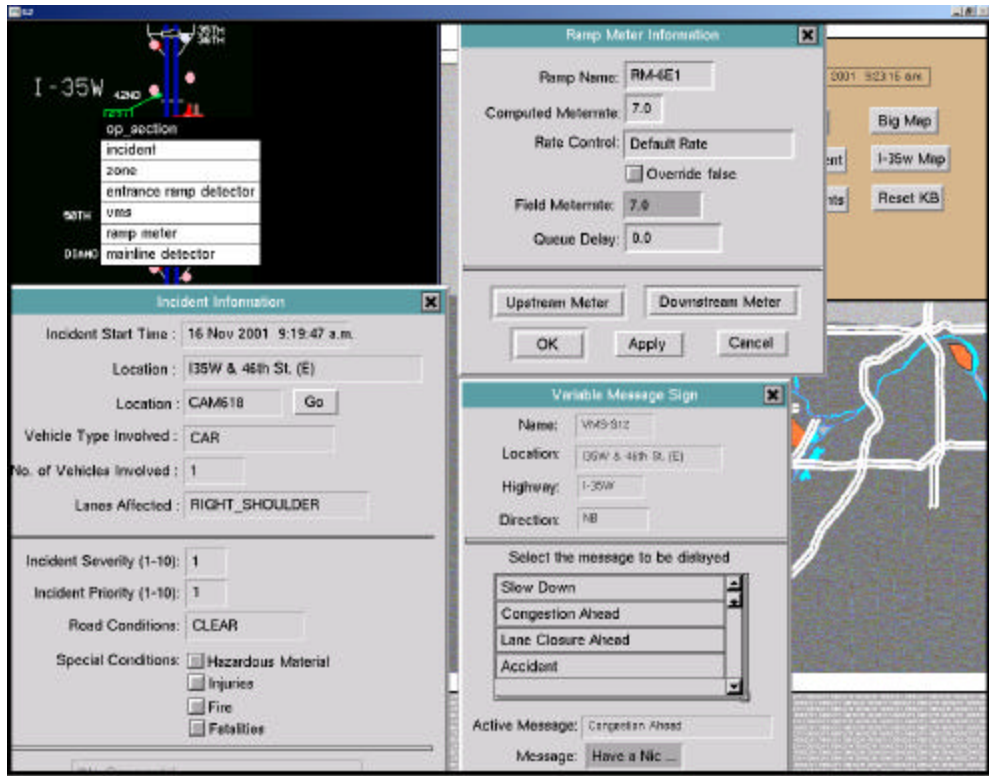


Figure 4.17 Workspace showing ways to get dialog box

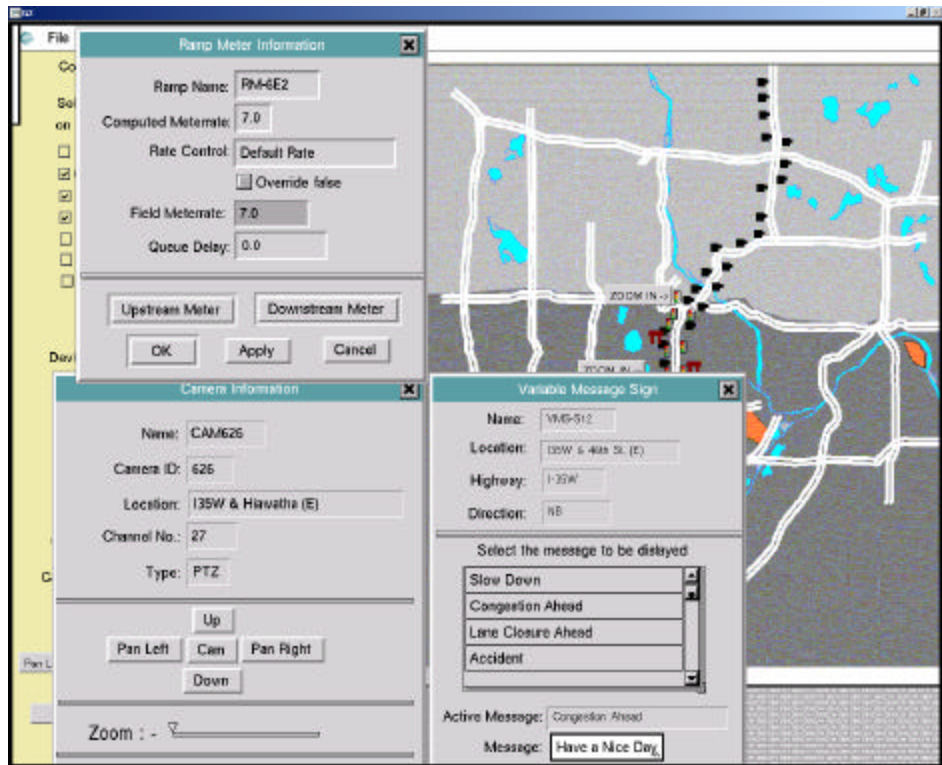


Figure 4.18 View dialog box

Chapter 5. TRAMLAB Implementation

As mentioned in Chapter 3, one of the objectives of TRAMLAB is to provide an offline environment for evaluating combinations of traffic management strategies prior to real time deployment and to train operators. This chapter presents two examples demonstrating the use of TRAMLAB for such purposes. The first example considers an incident management scenario in a real freeway corridor for evaluating two traffic management strategies. Evaluation of these strategies involved performance analysis for each and comparison of their effects in the entire corridor. The second example is related to training operators on the major TMC functions, i.e., Traffic Monitoring, Traffic Control, Incident Detection, Incident Verification, and Incident Response, in an eight mile freeway section in the Twin Cities area.

5.1 STRATEGY EVALUATION PROCEDURE

Effective traffic management, especially during incidents, requires fast and efficient deployment of strategies that would minimize the effects of congestion. This is currently done based on the operator's experience; however, evaluation of various alternative strategies prior to actual deployment would be preferable. For this purpose, a general procedure was developed allowing evaluation of alternative traffic management strategies for responding to incidents. The steps of the evaluation procedure are presented in the flow chart shown in the left part (shaded portion) of Figure 5.1.

The evaluation procedure is based on estimating the effects of each alternative and comparing key Measures of Effectiveness (MOEs), such as Total travel time, Total delay, and Average speed. The procedure begins with some offline preparations, which involves selecting the test site, building the network geometry, and entering the demand data into the simulator. Once these preparations are done, the network is completely ready for simulating any incident scenario as well as for estimating the MOEs resulting from application of the strategies that would be used as a response to the scenario. Next, two offline (without operator's interaction) simulation analyses are conducted to estimate the network's baseline conditions. In the first case, the network is simulated with normal traffic conditions (without incident). The MOEs resulting from this case are referred as baseline conditions1 (without incident). In the second case, the same network is simulated again with the same demand, but this time a predefined incident is also injected into the network at a certain stage of the simulation. The MOEs resulting from this case are referred to as baseline conditions2 (with incident). Comparison between the MOEs resulting from the two baseline conditions determines the performance deterioration in the network due to the incident.

Once both the baseline conditions are estimated, the network is simulated again with the same incident at the same location. But this time the operator, with the aid of various functional modules of TRAMLAB, performs the necessary steps of incident management (Figure 5.1), i.e., incident detection, incident verification, and incident response by applying appropriate traffic management strategies (or the strategies that need to be evaluated), which includes encouraging the drivers to take alternative routes, reducing the entrance flow rate in appropriate on-ramps, etc. Incidents are detected with the help of automated incident detection algorithms or by manually monitoring the traffic display maps, which changes its sections' color depending on the amount of congestion.

Therefore, if any section in the display map shows intense red color and if the speed at that section and a few upstream sections is highly reduced then

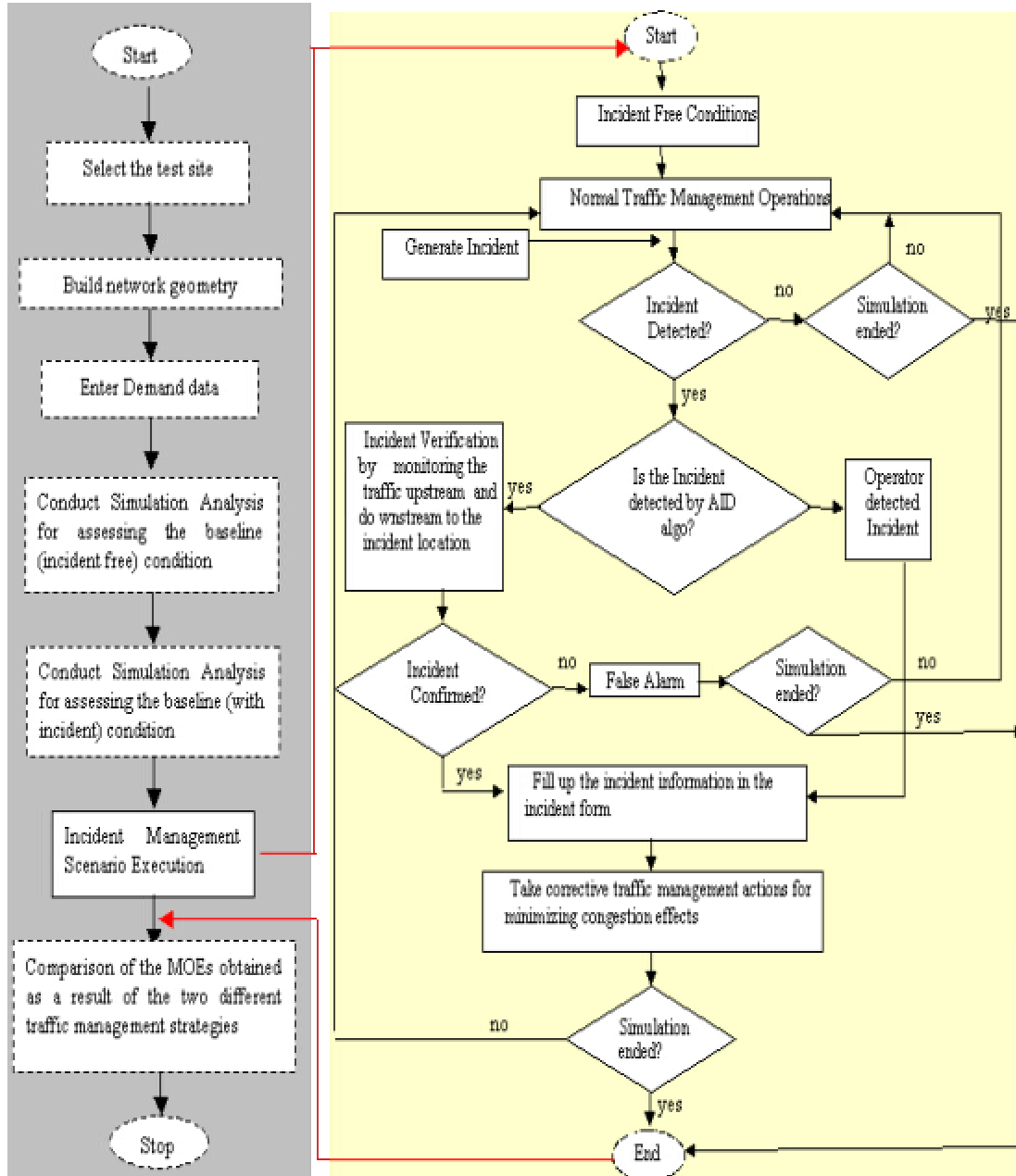


Figure 5.1 Flow chart for traffic management strategy evaluation

one can expect an incident at that location. However, it is essential to verify the incident before taking any traffic calming measures. The incident is verified by monitoring the incident location using CCTV cameras. Since a microscopic traffic simulator is used to represent the reality, verification is done by observing the speed values at the immediate upstream locations and by monitoring the simulated traffic at the incident location. After confirming the incident, the operator takes corrective measures to minimize the

congestion effects by employing an appropriate traffic management strategy from a set of pre-defined interactive strategies. In order to determine which of the pre-defined strategies would be most appropriate for the given incident scenario, it is necessary to estimate their effects separately. The resulting sets of MOEs are then compared to assess the relative performance change caused by the different strategies applied to alleviate the congestion caused due to the incident.

5.2 TEST SITE AND DATA FOR STRATEGY EVALUATION EXAMPLE

The test site consists of an eight-mile segment of the Interstate 494 and the adjacent sub-network of surface streets south of the Twin Cities encompassing four parallel and ten perpendicular arterial streets, which provide travelers with diversion routes in case of any

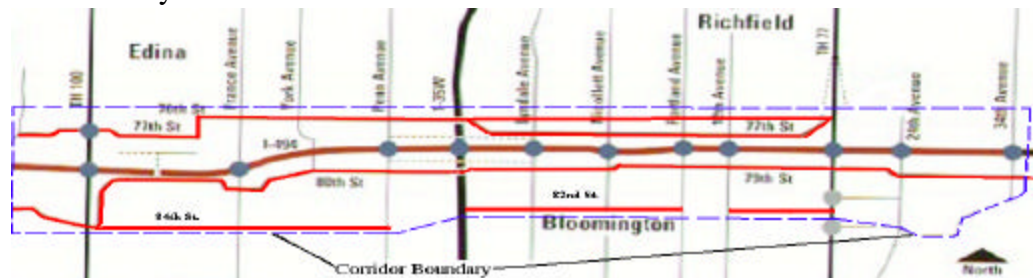


Figure 5.2. I-494 Transportation Corridor (ICTM)

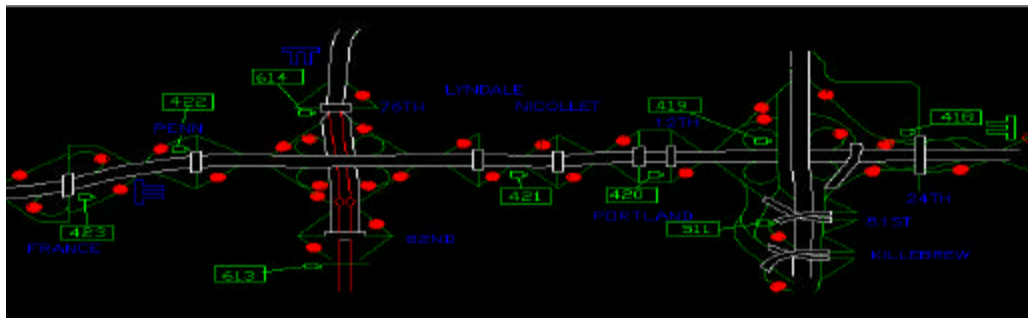


Figure 5.3. Location of various field devices in ICTM corridor

incident or abnormal traffic conditions (Figure 5.2). This section, which is also known as the Integrated Corridor Traffic Management (ICTM) corridor, was specifically chosen, because it is one of the most instrumented corridors in the Twin Cities. Moreover, it includes most of the common geometry sections in the city. The corridor is bounded by 76th Street in the North, 82nd Street in the South, 34th Avenue in the East, and East Bush Lake Road in the West. It also intersects with three principal highways, I-35W, TH-77 and TH-100 to provide sufficient access between suburban areas and from the suburbs to downtown Minneapolis. Therefore, the freeway-to-freeway ramps connecting these highways to I-494 carry an enormous amount of traffic during peak hours. For the extensive motorist information and surveillance system, the corridor is equipped with four freeway Variable Message Signs, nine arterial dynamic message signs, 11 Arterial Closed Circuit TV Cameras, and 81 Trailblazing Signs. The freeway variable message signs provide alternate non-freeway routes in the event of an incident or congestion.

Table 5.1 Interstate 494

Property	I-494 Eastbound	I-494 Westbound
Length (km)	12	12
Freeway Category	Metropolitan Area Beltway	Metropolitan Area Beltway
Upstream Boundary	E Bush Lake Road	34 th Ave
Downstream Boundary	34 th Ave	E Bush Lake Road
Metering Period	AM & PM	AM & PM
Zones	4D, 4E, 4F	4G, 4H
Bottleneck Locations	4D: Xerxes Avenue 4E: Nicollet Avenue 4F: TH 5	4G: Xerxes Avenue 4H: East Bush Lake Road
# of On-ramps	14	14
# of Off-ramps	13	13
# of Weaving Sections	3	3
# of Lane Drop Locations	0	1
Alternative Routes (arterials)	Present	Present

The corridor has about 75 signalized intersections and more than 250 un-signalized intersections. The geometric properties of the freeways Interstate 494 are presented in Table 5.1.

A hypothetical OD data set was used to represent the traffic demands for the test network. This was primarily due to the fact that the real OD data was unavailable and moreover it was not very important for the purpose of this implementation, which was just to demonstrate TRAMLAB's capability to evaluating traffic management strategies prior to real time deployment. The total time for which the network was simulated was three hours (between 2:05 to 5:05 pm) and the C-logit route choice model was used to represent the dynamic behavior of the traffic.

5.3 STRATEGY EVALUATION SCENARIO

An incident scenario is generated by injecting a pre-defined incident into the network. The predefined incident is injected in a three-lane section, just west of the I-35W and I-494 crossing, on westbound I-494. The incident starts at 4:00 pm, when all three lanes of the freeway section are blocked. At 4:20 pm the blockage in the leftmost lane is removed and then after five more minutes the middle lane is made free and finally at 4:30 pm, the whole incident is removed. The area around the location of the incident is generally very congested during the peak hours as the upstream freeway-to-freeway on-ramp carries a high volume of southbound I-35W traffic to westbound I-494. However, the presence of parallel and perpendicular arterials helps the travelers to divert to alternative routes, in case the downstream is highly congested. The two variable message signs that are known to have an effect on the traffic approaching the incident section are VMS1 (which is installed on southbound I-35W, just upstream to the freeway-to-freeway ramp) and VMS2 (which is installed on westbound I-494, upstream to the incident location) (Figure 5.4).

The two traffic management strategies that are evaluated as a response to the above described scenarios are:

Strategy1:

- Only immediately related VMS are used to display messages.
- No ramp meters will be overridden.

Strategy2

- Any VMS that are known to have an effect on the incident-affected route can be activated.
- Appropriate ramp meters can be overridden depending on the traffic conditions.

Scenario Execution

The scenario began with normal traffic conditions during which the selected operator used the GUI of TRAMLAB to perform the routine traffic management processes, such as traffic monitoring through traffic display maps and by viewing the variation of volume, occupancy, and speed at different freeway locations. In the first case, at about 4:01 pm the operator noted the occurrence of congestion in a particular section (as the color of that sections turned red), which was due to the predefined incident. Next, the operator observed the speed at the section immediately upstream. He found the speed to be zero, which made him conclude that the upstream traffic was at halt. To inform the approaching traffic about this incident, he immediately displayed an appropriate message on VMS1. As a result of this, most of the southbound I-35W traffic which, in the absence of the incident, would have continued through the freeway-to-freeway ramp on the westbound I-494 (represented by white arrows in the Figure 5.4), exited onto Penn Avenue (100%) and diverted equally (50%) to 76th and 80th Street (represented by the black arrows in Figure 5.4).

Also, a part of the southbound I-35W traffic, which in normal traffic conditions would have taken the freeway-to-freeway ramp, decided to continue straight on I-35W and meet 80th Street through 82nd Street (represented by the black arrows on I-35W in Figure 5.4).

These actions demanded almost constant supervision of the incident situation. After some time, while monitoring the traffic conditions, the operator observed that the speed of the section immediately upstream increased to 15 MPH. At this time he selected a different message (from the list of predefined messages), which enabled a small percentage (20%) of traffic to continue straight on westbound I-494. However, most of the traffic was still diverted to the parallel arterials. Finally, at 4:35pm the operators observed that the speed of the upstream section increased to 30 MPH, which made him sure that the incident was removed and congestion reduced; therefore, he/she removed the displayed message.

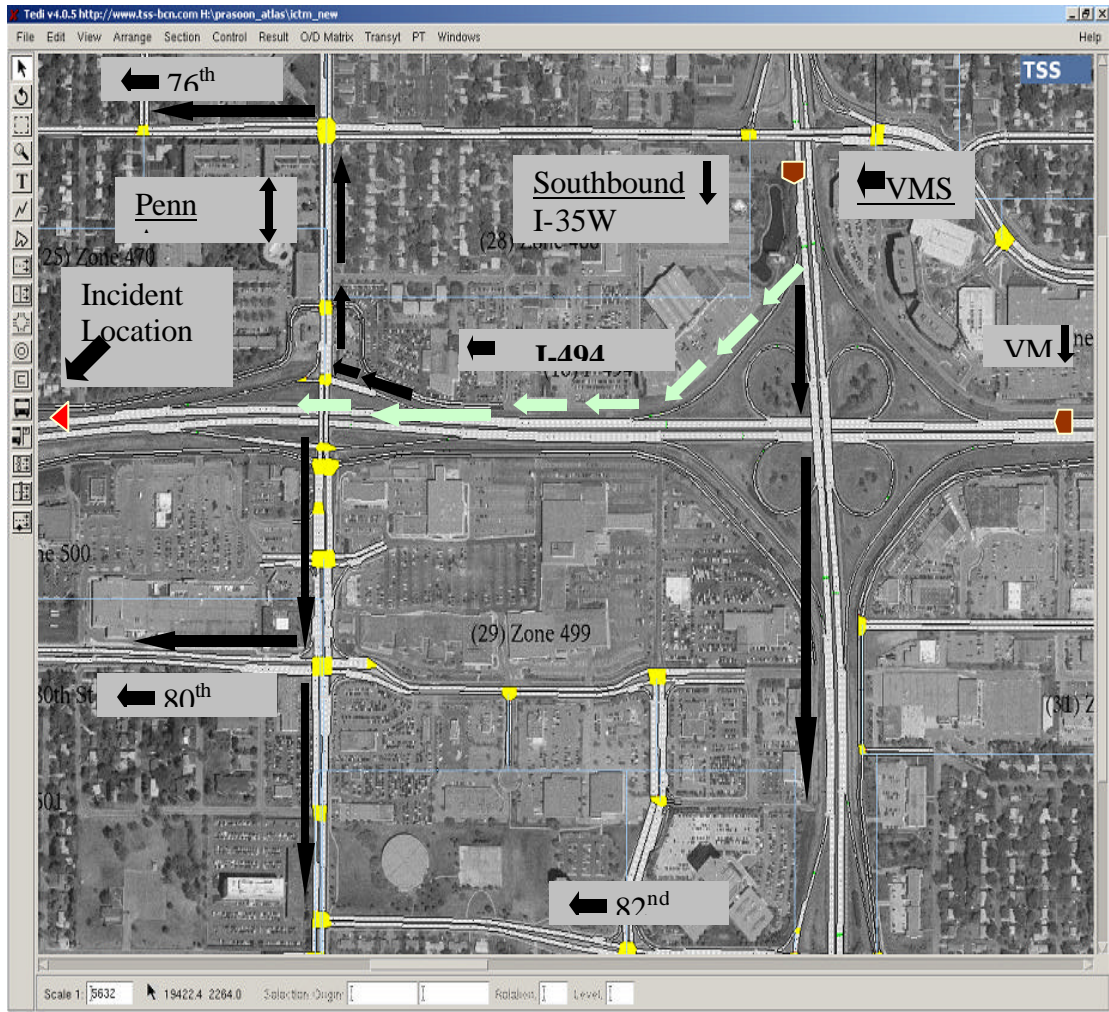


Figure 5.4 Incident location and traffic diversion

In the second case, the more general strategy (strategy2) was employed as a response to the incident. In this case, the same messages were displayed by VMS1 but in addition messages were also displayed on VMS2 to inform and divert the upstream westbound I-494 traffic. The first message displayed on VMS2 diverted the traffic through the freeway-to-freeway ramp to southbound I-35W (40%) and through the exit ramp to Penn Avenue (as represented in Figure 6.5). A few upstream ramp meters (4H2, 4H3, and 4H4), already working on rate 5, were overridden to their most restrictive rate (rate 6) to restrict the flow of traffic entering the southbound I-494. These ramp meters are represented by the symbol ●.



Figure 5.5 Effects of Strategy2

5.4 RESULTS

The three MOEs considered for the evaluation were total travel time, average speed, and total delay.

5.4.1 Travel Time

Travel time refers to the time it takes a vehicle to traverse a given link. The total network travel time refers to the summation of all link travel times through the corridor. The total network travel time was broken down into total mainline, ramp, and arterial travel times in order to determine how they react to the given various cases considered.

Table 5.2 MOE comparisons

MOE	Aggregation Level	B1	B2	Stg1	Stg2	B1and B2	B2 and Stg1	B2 and Stg2	Stg1 and Stg2
Total Travel Time (Veh-Hrs)	Mainline	2368	2901	2414	2327	22.50	-16.7	-19.7	-3.60
	Ramp	622	702	675	722	12.86	-3.84	2.84	6.96
	Arterial	1154	1211	1345	1441	4.93	11.0	18.9	7.13
	Entire Site	4144	4814	4434	4490	16.16	-7.89	-6.73	1.26
Total Travel (Veh-Miles)	Mainline	276352	255344	294602	301484	-7.60	15.3	18.06	2.33
	Ramp	6005	5955	6122	6044	-0.83	2.80	1.49	-1.27
	Arterial	34942	30419	33442	34563	-12.94	9.93	13.6	3.35
	Entire Site	317299	291718	334166	342091	-8.06	14.5	17.2	2.37
Speed (MPH)	Mainline	45	32	40	45	-28.8	25	40.6	12.5
	Ramp	40	30	35	41	-25	16.6	36.6	17.1
	Arterial	32	30	28	24	-6.25	-6.66	-20	-14.2
	Entire Site	38	30	34	36	-21.0	13.3	20	5.88
Delay (Veh-Hrs)	Mainline	942	1320	1208	1044	40.1	-8.48	-20.9	-13.5
	Ramp	211	221	218	243	4.73	-1.35	9.95	11.4
	Arterial	612	714	740	784	16.6	3.64	9.80	5.94
	Entire Site	1765	2255	2166	2071	27.7	-3.94	-8.15	-4.38

Where,

B1- Baseline Condition1 (without incident)

B2- Baseline Condition2 (with incident)

Stg1- (Case 3) Strategy 1

Stg2 – (Case 4) Strategy 2 (The last four columns denote the percentage change in MOEs)

Travel time generally decreases when any control strategy is applied as a response to a bad traffic condition. When compared to the incident base case (Baseline2), both the cases 3 and 4 provided mainline travel time reduction of 16.7 and 19.7 percent, respectively. It can be observed that the arterial travel time increased by 18.9% in case 4 as compared to 11% in case 3. This was because in case 4 (strategy 2) the operators displayed the incident information on both VMS1 (placed I-35W) and VMS2 (placed on I-494) and thus larger amounts of traffic were diverted to the arterials as compared to case 3 where only the I35W travelers got the incident information, resulting in less diversion and therefore less arterial travel time. Also, since in case 4 the operators had overridden the upstream ramp meters to the most restrictive rates, the ramp travel time is

higher as compared to case 3 in which none of the ramp meters were overridden. Finally, it was observed that the total travel time increased by 1.26% in case 4 as compared to case 3. This implies that although both the strategies used were effective for the entire network, there was not much of a difference between them.

5.4.2 *Speed*

It can be observed from the estimated MOEs that both strategies provided the mainline and ramps with higher speeds. The speeds in the arterials were reduced because of congestion due to the diverted traffic. Overall, strategy 2 was more effective in providing the entire network with a higher speed (5.88%) as compared to strategy 1.

5.4.3 *Delay*

It can be observed that strategy2 decreased the mainline delay by 13.5% as compared to strategy1. The ramp and arterial delays were increased (11.4 % for ramps and 5.94 % for arterials) when strategy 2 was applied, as compared to strategy1, because of larger diversion to the arterials. The reason for increased ramp delay is that the strategy restricted the ramp traffic entering the freeway. But the entire network delay was reduced (4.38 %) when strategy 2 was applied as compared to strategy1.

Following common sense one usually assumes that travel time and delay follow similar trends. But the MOE comparison (percentage change between strategy1 and strategy 2) presented here shows that although the entire network total travel time increased by 1.26 %, the delay decreased by 4.38 %. This is because of the way delay is calculated by the simulator. The delay calculated here is not the summation of the expected and actual time (for each vehicle) difference between any two points (which is the delay in a practical sense). Instead, it is the summation of the difference in the free flow travel time of a route and the actual time each vehicle took to reach its destination through that route. For example, two vehicles might have the same origin and destination but the route they took to reach the destination might be of different length and travel time due to the diversion; unless they travel below free flow speed, delay is not generated.

Based on the estimated MOEs, it can be concluded that strategy2 was a better control alternative as compared to strategy1. However, there was not much of a difference between the effects of the two strategies used for evaluation. It should be noted that the strategy evaluation was performed just to demonstrate TRAMLAB's capability as an effective evaluation tool.

5.5 OPERATOR TRAINING

Control room operators are generally trained by making them observe other operators performing the various traffic management operations. This way of training the operators is fine, but for the operator to work efficiently, he/she should be well trained on the use of the various tools, and be able to display a VMS message, override ramp meters, etc. Further, the operator should be well aware of the various real-life scenarios and the different ways to tackle them. This can be achieved by training the operators in an offline environment, on different real-life scenarios. TRAMLAB's capability of

fulfilling these training needs is presented here by considering an incident management scenario.

The first stage of the training process was to plan what type of incident scenario should be considered in order to train the operators on all the major TMC functions. It was decided to simulate an incident on a section of southbound I35W, the chosen training scenario site. Next, the planned network (Figure 5.1) was created in TEDI and the traffic demands, observed on March 23, 2000, were entered in the form of five minute detector counts. The total time the network was simulated was two hours (between 3:05 to 5:05pm). The other characteristics of the site are described in Table 5.1. An incident involving two blocked right-hand lanes was planned to occur on southbound I35W between the Diamond Lake Road and 50th street crossings at 4:00pm (Figure 5.1). The total time the incident was active was 25 minutes.

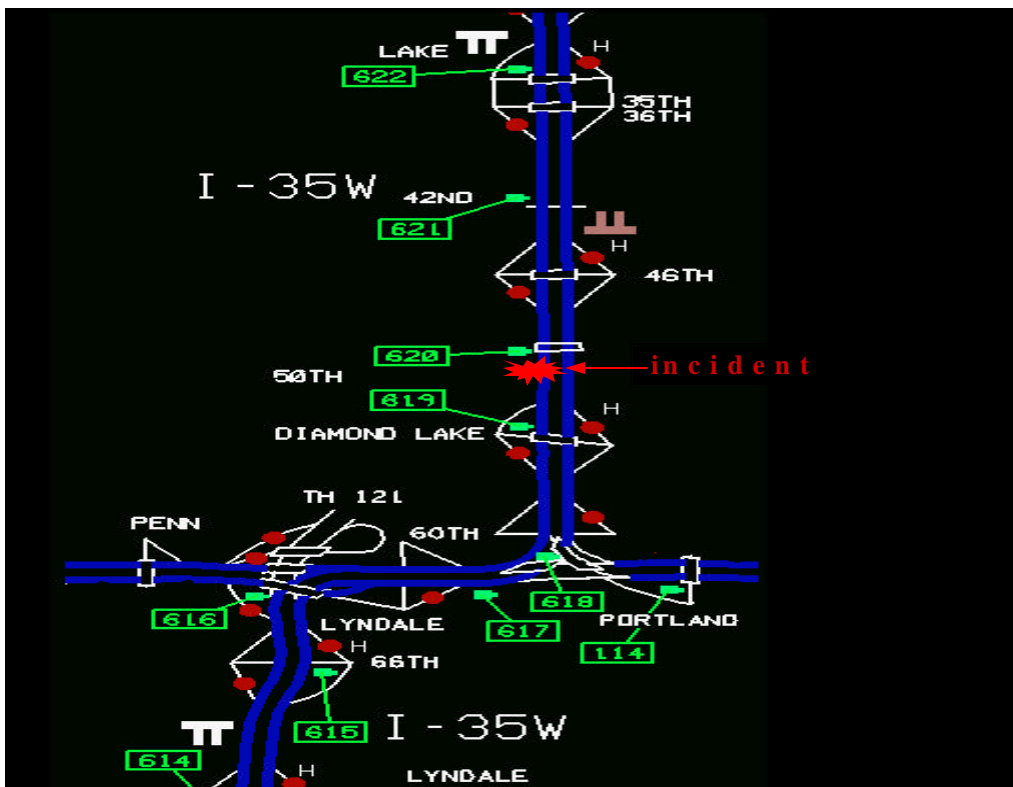


Figure 5.1 Training site (I-35W)

Table 5.3 Training site characteristics

Characteristic	I-35W
Boundaries	Downtown Mpls – TH 13
Length (km)	24 (both ways)
Metered Ramps	
Off ramps	17(N) + 20(S)
VMS	2(N) + 2(S)
Geometric Complexity	Very Complex
Traffic Levels	Very High

The implementation of this process started by selecting two operators who needed to be trained. Both operators were provided with separate operator workstations each consisting of a monitor that represents the GUI of TRAMLAB. Once both the operators were logged in the operator mode, the simulation was started in the background. At first, both the operators were familiarized with the various objects (detectors, stations, ramp meters, VMS, traffic display maps, incident dialog, etc.) incorporated in TRAMLAB. They were then shown how to access the information dialog of each object (explained in Chapter 4). Further, they were trained on the modules of TRAMLAB in order to teach them how to monitor current traffic conditions with the help of traffic display maps, CCTV cameras, graphs of volumes, occupancy, speeds etc, as well as how to detect, verify, and respond to incidents (selecting VMS messages, etc.) or any abnormal traffic conditions by making use of the GUI tools. Subsequent to this, the operators were then allowed to familiarize themselves with the system. During this process, they encountered the congestion resulting from the incident (they had no prior knowledge about this incident). They were then guided through the various stages of incident management (explained in Chapter 4).

During the training process, the operators were encouraged to ask questions in case they had doubts or problems. Some of the questions asked were:

- How can the occupancy value at a particular section be found?
Click on the display map at that section. Select the option “detector station” from the section menu. It will show the detector station dialog containing the volume, occupancy, speed, and other attributes.
- Is it possible to detect any malfunctioning field device?
It is not possible at present but algorithms can easily be incorporated.
- Is it possible to store information about a previous incident?
Yes, the information is stored in the section where the incident happened.
- How can I see speed and occupancy graphs?
By clicking the button “Graphs” in the detector information dialog.

Finally, it was found that the operators were quite comfortable in using the GUI of TRAMLAB to deal with the scenarios. Therefore, the operator training was successfully accomplished.

Chapter 6. Concluding Remarks

This report presents the development of a virtual traffic management center called TRAMLAB, which can be used by researchers, practitioners, TMC managers, and operators to serve a number of practical needs. These include designing a new TMC or improving an existing one, testing new technologies separately or in combination (incident management schemes, ramp metering) prior to real time deployment, estimating the effects of integrating the major functions (Traffic Monitoring, Traffic Control, Incident detection, Incident Verification, Incident Response), evaluating a combination of traffic management strategies, and improving control room operations and training operators offline.

The design and development of TRAMLAB was based on considerations resulting from a survey of the Mn/DOT control room operations, and the standards and guidelines set forth in the National ITS Architecture. Realizing the objectives of TRAMLAB, a flexible, friendly user-defined graphical user interface was developed and integrated with a microscopic simulator to allow operator's real time interaction with the system; examples include manual overriding of the ramp metering rates, and displaying appropriate VMS messages depending on the current traffic conditions.

In order to illustrate implementation of TRAMLAB as an efficient evaluation and training tool, two different incident management scenarios were considered. In the strategy evaluation example, two different interactive traffic management strategies were selected and evaluated for an incident management scenario on the ICTM corridor. It was found that the strategy (strategy2) which used ramp meter overrides and displayed incident information on two VMS (one on I35W and other on I494) was better as compared to the other (strategy1) which informed only the southbound I-35W about the incident. Further, in the operator-training example, two operators were trained to use various modules of TRAMLAB for performing efficient traffic management. This was performed by considering an incident management scenario on a part of freeway I-35W. It was found that the operators were able to learn to use the GUI quite easily and quickly.

During the course of development, many presentations and demonstrations were made to visitors from FHWA, Mn/DOT, and other transportation engineers, based on their reactions and comments considerable changes and improvements were made.

In spite of the completion of the TRAMLAB objectives, there is still room for further expansion and improvements. One example includes incorporation of algorithms for detecting the malfunctioning of field devices such as detectors. Another improvement would be to expand TRAMLAB for urban street actuated and adaptive control (as opposed to pre-timed only) incorporating actuated control interfaces as well as SCOOT and SCATS or other traffic-responsive schemes. Further, a web interface can be developed to provide the non-university users with access to TRAMLAB.

REFERENCES

1. <http://www.tfhrc.gov/pubrds/marapr98/toc.htm>.
2. "Development of Intelligent Transportation Systems Using the Nation ITS Architecture", July 1998- An Executive Edition for Senior Transportation Engineers.
3. "Developing Freeway and Incident Management Systems Using the National ITS Architecture", August 1998.
4. Sowmitra Thallapragada, "Methodology for Testing and Evaluating Traffic Control Room Operations", M.S Plan B Report, University of Minnesota, 2000.
5. "Metropolitan Transportation Management Center- Concept of Operations: Houston TranStar", October 1999.
6. "Metropolitan Transportation Management Center- Concept of Operations: San Antonio TransGuide", October 1999.
7. "Metropolitan Transportation Management Center- Concept of Operations: Milwaukee MONITOR", October 1999.
8. "Metropolitan Transportation Management Center- Concept of Operations: Georgia NaviGator", October 1999.
9. [http://www.its.dot.gov/TravelManagement/traffic_research_laboratory_\(trel\).htm](http://www.its.dot.gov/TravelManagement/traffic_research_laboratory_(trel).htm)
10. CORSIM User Guide (on-line), Developed by ITT Industries for FHWA Office of Operations Research, Development and Technology.
11. TTI Research Report 1271-1F, *PASSER IV-94, Version 1.0, User/Reference Manual*, Nadeem A. Chaudhary and Carroll J. Messer. 1993.
12. Logi, F., Rindt, C.R., McNally, M.G., and Ritchie, S.G., "TRICEPS/CARTESIUS: An ATMS Testbed Implementation for the Evaluation of Inter-Jurisdictional Traffic Management Strategies", UCI-ITS-WP-00-20, 2000.
13. Jayakrishnan, R., Mahmassani, H.S., and Hu, T.Y. (1994), "DYNASMART: An Evaluation Tool for ATIS/ATMS in Urban Traffic Networks," *Transportation Research C*, Vol. 2, No.3, pp. 129-147
14. Wicks, D.A., "INTRAS- a microscopic freeway simulation model. Overview of simulation in highway transportation,"1, pp.95-107, 1977.
15. Ben-Akiva, Moshe, Haris Koutsopoulos, Rabi Mishalani, and Qi Yang, "Simulation Laboratory for Evaluating Dynamic Traffic Management Systems," *ASCE Journal of Transportation Engineering*, Vol.123, No. 4, 1997, pp.283-289
16. a) <http://www.gensym.com/product/#g2>
b) Gensym Corporation (1995), G2 System Procedures, Reference Manual Version 4.0.
c) Gensym Corporation (1995), G2 GUIDE/UIIL, Procedure Reference Manual Version 4.0.
d) Gensym Corporation (1995), G2 GUIDE User's Guide Version 4.0.
e) Gensym Corporation (1997), G2-Gateway, Bridge Developer's Guide, Version 5.0.
f) Gensym Corporation (1995), G2 Standard Interface, User's Manual Version 4.0.
17. AIMSUN2, Version 4.0, User Manual, Draft Version, Transport Simulation Systems, 2001.
18. Reis, Robert A., Nathan H. Gartner, and Stephen L. Cohen, "Dynamic Control and Traffic Performance in a Freeway Corridor: A Simulation Study," *Transportation Research A*, Vol 25A, No. 5, 1991, pp. 267- 276.

19. Hourdakis, J., and Michalopoulos, P.G., "Towards the Development of Next Generation Traffic Management Centers: The TRAMLAB System".
20. Koka, M., Hourdakis, J., and Michalopoulos, P.G., "Computer Aided Testing and Evaluation of Adaptive Ramp Control Strategies", *Transportation Research Board*.

Bibliography

1. Mn/DOT Traffic Management Center, Traffic Control Room Operators Manual.
2. U.S. Department of Transportation, FHWA, Comparable Systems Analysis: Design and Operation of Advanced Control Centers.
3. U.S. Department of Transportation, FHWA, Design of an ITS-Level Advanced Traffic Management System – A Human Factors Perspective.
4. Van Aerde, M.W., and S. Yagar, "Combining Traffic Management and Driver information in Integrated Traffic Networks," Third International Conference on Road Traffic Control – May 1-3, 1990, IEE, 1990, pp. 11-15.
5. Abdel-Rahim, A., Khanal, M. and Rust, P.S., "Evaluation of Freeway Diversion Route Plans Using Microscopic Simulation and Real-Time Data: The Case of Idaho's Treasure Valley Corridor", Transportation Research Board, 2002.
6. Mass, G., Maggio, M., Shafie Hadi and Stough, R., "Incident Management and Intelligent Transportation Systems Technology: Estimating Benefits for Northern Virginia", The Intelligent Transportation Society of America, 1998.
7. Cuneo, D., Jha, M., and Ben-Akiva, M., "An Evaluation of Integrated Freeway Traffic Control and Route Diversion using Microscopic Simulation".

Appendix A – Rules and Procedures (Code)

1.1 SHOW-OPSECTION-USER-MENU (PROCEDURE)

```
show-opsection-user-menu(T:text, s:symbol, win: item-or-value, op : class op_section)
ds : class object;
begin
  if the object named by s exists then
  begin
    ds = the object named by s;
    start uil-start-or-refocus-dialog(T, ds, win) ;
  end;
end
```

1.2 SHOW-OPSECTION-INCIDENT-USER-MENU (PROCEDURE)

```
show-opsection-incident-user-menu(T:text, Inc: class incident, win: item-or-value)
begin
  start uil-start-or-refocus-dialog(T, Inc, win) ;
end
```

1.3 UPSTREAM_OR_DOWNSTREAM_STATION_DIALOG (PROCEDURE)

```
upstream_or_downstream_station_dialog (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value,
Win: item-or-value, OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)
op : class op_section ;
DS : class detector-station;
flg:integer = 0;
C: class station-connection;
obj: symbol;
begin
  for op = each op_section do
  if (the mainline_detector of op = the name of OrigDialogLauncher) then
  begin
    if ( the id of ThisDialogLauncher = "upstream") then
    begin
      C = the station-connection connected at the input of op;
      if ( there exists an object connected at the input of C) then
      begin
        DS = the detector-station named by the mainline_detector of the op_section connected at the input
of C;
        end
        end
      else
      begin
        C = the station-connection connected at the output of op;
        if ( there exists an object connected at the output of C) then
        begin
          DS = the detector-station named by the mainline_detector of the op_section connected at the output
of C;
          end;
          end;
        if DS exists then
        begin
```

```

    call uil-start-or-refocus-dialog ("Detector Station Information", DS, Win);
    flg = 1;
    exit if flg = 1;
end
end;
end;
if flg = 0 then
begin
    if (the id of ThisDialogLauncher = "upstream") then
    begin
        call uil-post-generic-dialog("There is no detector station upstream to this detector station", the
symbol message, the symbol large, Win, the symbol uil-information-icon)
    end
    else
    begin
        call uil-post-generic-dialog("There is no detector station downstream to this detector station", the
symbol message, the symbol large, Win, the symbol uil-information-icon)
    end;
    end;
end
end

```

1.4 STATION_TO_DET (PROCEDURE)

```

station_to_det (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win: item-or-value,
OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)
v: item-or-value;
d: class detector;
lane_number:integer;
flg:integer = 0;
begin
lane_number = quantity (the id of ThisDialogLauncher);
v = call uil-get-active-dialog-on-window-if-one-exists( "Detector Information", Win) ;
for d = each detector do
    if ( the detector_station of d = the name of OrigDialogLauncher) and ( the lane of d = lane_number )
then
    begin
        call uil-start-or-refocus-dialog ("Detector Information", d, Win);
        flg = 1;
        exit if flg = 1;
    end;
end;

    if flg = 0 then
    begin
        call uil-post-generic-dialog("There is no detector in this lane of this detector station", the symbol
message, the symbol large, Win, the symbol uil-information-icon)
    end;
end

```

1.5 DETECTOR_INFO_IN_STATION_DIALOG (PROCEDURE)

```

detector_info_in_station_dialog (G: class item, W: class ui-client-item, D-or-W: class item, B: item-or-
value, action-queue: item-or-value)
T: text;
quan: text;
lane_number: integer;

```

```

dt : class detector;
v: value;
flg:integer = 0;
begin
  {-- The update method for an object takes the value specified by the object's uil-event-source-object and
  uil-event-source-attribute and reflects it in the object's state. (Note that for objects residing on workspaces,
  the update method works only if the source object is specified by name.)

```

G is the UIL object on which to run the update method
 W is the window on which D-or-W is managed
 D-or-W is the dialog or workspace containing the object
 B is the button or object that initiated the update action (optional)
 action-queue is the list of pending actions for the dialog initiated by B (optional)

No values are returned by this method. -- }

T = the id of G;

quan = get-from-text(T, 2, 4);

lane_number = quantity(T);

for dt = each detector do

 if (the detector_station of dt = the name of B) and (the lane of dt = lane_number) then

 begin

 case (quan) of

 "vol":

 begin

 collect data (timing out after -1 seconds)

 v = the volume of dt;

 flg = 1 ;

 end;

 end;

 "occ":

 begin

 collect data (timing out after -1 seconds)

 v = the occupancy of dt;

 flg = 1 ;

 end;

 end;

 "spd":

 begin

 collect data (timing out after -1 seconds)

 v = the speed of dt;

 flg = 1 ;

 end;

 end;

 end;

 end;

 end;

 if flg = 1 then

 begin

 call uil-update-text(G, v);


```

    end
    else
    begin
    call uil-update-text (G, " N/A");
    end;
end

```

1.6 RULE FOR STARTING THE PROCEDURE 'UPDATE_DIALOG' TO UPDATE VALUES IN DETECTOR-STATION DIALOG

“whenever the volume of ds-9 receives a value and when there exists a g2-window W then start update_dialog ("Detector Station Information", W)”

UPDATE_DIALOG (PROCEDURE)

```

update_dialog( T: text, win: item-or-value )
B: class object;
begin
wait for 0.5 seconds;
  B = call uil-get-active-dialog-on-window-if-one-e-exists( T, win ) ;
  call uil-call-update-method-for-groobj (B, win) ;
end

```

1.7 GO1 (PROCEDURE)

```

go1 (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win: item-or-value,
OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)
ssymbol: item-or-value ;
begin
ssymbol = the detector-station named by the detector_station of OrigDialogLauncher ;
call uil-start-or-refocus-dialog ("Detector Station Information", ssymbol, Win);
end

```

1.8 GRAPHS (PROCEDURE)

```

graphs (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win: item-or-value,
OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)
T, S : class trend-chart;
i, j: integer;
O: symbol;
begin
O = the name of OrigDialogLauncher;
for T = each trend-chart upon graphswksp do
  i = the item-x-position of T;
  j = the item-y-position of T;
  create a trend-chart S by cloning T;
  transfer S to the subworkspace of the object named by O;
  make S permanent;
end;
show S;
end

```

1.9 ADJACENT1 (PROCEDURE)

```

adjacent1 (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win: item-or-value,
OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)
i:integer;

```

```

flg: integer = 0;
s:symbol;
ssymbol: item-or-value;
dsymbol: item-or-value;
d: class detector;
begin
s= the detector_station of OrigDialogLauncher;
i = the lane of OrigDialogLauncher;
ssymbol = the detector-station named by the detector_station of OrigDialogLauncher;

for d= each detector do
  if (the detector_station of d = s and the lane of d = i - 1) then
  begin
    call uil-start-or-refocus-dialog ("Detector Information", d, Win);
    flg = 1;
  end
end ;
  if flg = 0 then
  begin
    call uil-post-generic-dialog("There is no detector to the left of this detector", the symbol
message, the symbol large, win, the symbol uil-information-icon)
  end
end

```

1.10 ADJACENT2 (PROCEDURE)

```

adjacent2 (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win: item-or-value,
OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)
i:integer;
flg: integer = 0;
s:symbol;
ssymbol: item-or-value;
d: class detector;
begin
s= the detector_station of OrigDialogLauncher;
i = the lane of OrigDialogLauncher ;
ssymbol = the detector-station named by the detector_station of OrigDialogLauncher;

for d= each detector do
  if (the detector_station of d = s and the lane of d = i + 1) then
  begin
    call uil-start-or-refocus-dialog ("Detector Information", d, Win);
    flg = 1;
  end
end ;
  if flg = 0 then
  begin
    call uil-post-generic-dialog("There is no detector to the right of this detector", the symbol
message, the symbol large, win, the symbol uil-information-icon)
  end
end

```

1.11 ADD_TREND-CHARTS (PROCEDURE)

```

add_trend-charts (arr: class symbol-array, w: class kb-workspace, darr: class symbol-array)

```

```

d: class detector;
T : class trend-chart;
S: text ;
M: text;
flg:integer =0;
objname:symbol;
dlen:integer= the array-length of darr;
u: class uil-edit-box-medium;
len: integer = the array-length of numberofcharts;
begin
inform the operator that "[len]";
for u = each uil-edit-box-medium upon w do
if the id of u = "add" then
begin
S = the message-contents of u;
objname = symbol(S);
if ( the object named by objname exists) then
begin
change the array-length of arr to (len + 1);
conclude that arr[len] = objname;
call g2-array-copy-elements-to-initial-values (arr);
for T = each trend-chart upon w do
case ( the title of T ) of
"Volume Vs Time":
begin
M = " the volume of [S]" ;
end;
"Speed Vs Time":
begin
M = " the speed of [S]" ;
end;
"Occupancy Vs Time":
begin
M = " the occupancy of [S]" ;
end;
"Density Vs Time":
begin
M = " the density of [S]" ;
end;
end;
call g2-add-trend-chart-component( T, the symbol plots) ;
call g2-add-trend-chart-component( T, the symbol connector-formats) ;
call g2-set-text-of-trend-chart-component ( T, the symbol plots, the symbol Use-local-history@?,
len + 2, "no");
if dlen /= 0 then
begin
call g2-set-text-of-trend-chart-component ( T, the symbol connector-formats, the symbol Names,
len + 2, "[darr[dlen - 1]]");
call g2-set-text-of-trend-chart-component ( T, the symbol connector-formats, the symbol
connector-line-color, darr[dlen - 1], "[darr[dlen - 1]]");
call g2-set-text-of-trend-chart-component ( T, the symbol plots, the symbol connector-format-
name-or-number, len + 2, "[darr[dlen - 1]]");
flg = 1;
end
else
begin

```

```

    call g2-set-text-of-trend-chart-component ( T, the symbol connector-formats, the symbol Names,
len + 2, "[symbol(graphcolor[len])]");
    call g2-set-text-of-trend-chart-component ( T, the symbol connector-formats, the symbol
connector-line-color, symbol(graphcolor[len]), graphcolor[len]);
    call g2-set-text-of-trend-chart-component ( T, the symbol plots, the symbol connector-format-
name-or-number, len + 2, "[symbol(graphcolor[len])]");
    end;
    call g2-set-text-of-trend-chart-component (T, the symbol plots, the symbol expression, len + 2, M);
end;
if (flg = 1) then
begin
conclude that darr[dlen - 1] =the symbol g2;
call g2-array-copy-elements-to-initial-values (darr);
change the array-length of darr to (dlen - 1);
call g2-array-copy-elements-to-initial-values (darr);
end;
end;
end;
end;
end

```

1.12 DELETE_TREND-CHARTS (PROCEDURE)

```

delete_trend-charts(arr1: class symbol-array, w: class kb-workspace, darray: class symbol-array)
u : class uil-edit-box-medium;
T: text;
chart: class trend-chart;
objname: symbol;
flg, i, j, k, m, n, p, q, r: integer;
dlen: integer = the array-length of darray;
len: integer = the array-length of numberofcharts;
begin
for u = each uil-edit-box-medium upon w do
if the id of u = "delete" then
begin
T = the message-contents of u;
objname = symbol (T);
for i = 0 to (len - 1) do
if (numberofcharts[i] =objname) then
begin
j = i;
end;
end;
if (len - 2) >= j then
begin
for k = j to (len - 2) do
conclude that arr1[k] = numberofcharts[k + 1];
end;
end;
conclude that arr1[len - 1] = the symbol g2;
call g2-array-copy-elements-to-initial-values (arr1);
change the array-length of arr1 to (len - 1);
call g2-array-copy-elements-to-initial-values (arr1);
for chart = each trend-chart upon w do
call g2-delete-trend-chart-component (chart, the symbol plots, j + 2);
r = j;

```

```

if dlen >= 1 then
begin
for m = j to 2 do
for p = 0 to (dlen - 1) do
if (graphcolor[m] = "[darray[p]]" ) then
begin
flg = 0;
exit if flg = 0;
end
else
begin
flg = 1;
end
end;
r = m;
exit if flg = 1;
end;
end;

call g2-delete-trend-chart-component (chart, the symbol connector-formats,
symbol(graphcolor[r]));
end;
change the array-length of darray to (dlen + 1);
conclude that darray[dlen] = symbol(graphcolor[j]);
call g2-array-copy-elements-to-initial-values (darray);
end;
end;
end

```

1.13 HIDE-CHART-WKS

hide-chart-wks(w: class kb-workspace, chartnum: class symbol-array, delchartcolor : class symbol-array)

```

len:integer = the array-length of chartnum;
dlen:integer = the array-length of delchartcolor;
T: class trend-chart;
i, j :integer;
begin
for i = 0 to (len - 1) do
conclude that chartnum[i] = the symbol g2;
end;
for j = 0 to (dlen - 1) do
conclude that delchartcolor[j] = the symbol g2;
end;
call g2-array-copy-elements-to-initial-values (chartnum);
call g2-array-copy-elements-to-initial-values(delchartcolor);
change the array-length of chartnum to 0;
change the array-length of delchartcolor to 0;
call g2-array-copy-elements-to-initial-values (chartnum);
call g2-array-copy-elements-to-initial-values(delchartcolor);
for T= each trend-chart upon w do
make w transient;
delete T;
make w permanent;
hide w;
end
end

```

end

1.14 UPSTREAM_OR_DOWNSTREAM_RAMP_DIALOG (PROCEDURE)

upstream_or_downstream_ramp_dialog (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win: item-or-value, OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)

```
op : class op_section ;
DS : class ramp -meter ;
flg:integer = 0;
C: class ramp -connection;
obj: symbol;
begin
for op = each op_section do
if (the ramp of op = the name of OrigDialogLauncher) or (the name of OrigDialogLauncher = the
ramp2 of op) then
begin
if ( the id of ThisDialogLauncher = "upstream") then
begin
C = the ramp -connection connected at the input of op;
if ( there exists an object connected at the input of C) then
begin
DS = the ramp -meter named by the ramp of the op_section connected at the input of C;
end
end
else
begin
C = the ramp -connection connected at the output of op;
if ( there exists an object connected at the output of C) then
begin
DS = the ramp -meter named by the ramp of the op_section connected at the output of C;
end;
end;
if DS exists then
begin
call uil-start-or-refocus-dialog ("Ramp Meter Information", DS, Win);
flg = 1;
exit if flg = 1;
end
end;
if flg = 0 then
begin
if (the id of ThisDialogLauncher = "upstream") then
begin
call uil-post-generic-dialog("There is no ramp meter upstream to this ramp meter", the symbol
message, the symbol large, Win, the symbol uil-information-icon)
end
else
begin
call uil-post-generic-dialog("There is no ramp meter downstream to this ramp meter ", the
symbol message, the symbol large, Win, the symbol uil-information-icon)
end;
end;
end;
end
```

1.15 RULE FOR OVERRIDING RAMP METERS

“whenever the override of any ramp-meter r receives a value and when the override of r is true then start **set_override_rate**(the name of r, the field_meterrate of r) across polaris”

(the SET_OVERRIDE_RATE is a remote procedure declaration that changes the rate of the ramp meter in AIMSUN through the G2-AIMSUN interface)

The remote procedure is declared as : -

```
declare remote set_override_rate(value, float)=()
```

1.16 RULE FOR REMOVING THE OVERRIDE OF THE RAMP METERS

“whenever the override of any ramp-meter r receives a value and when the override of r is false then start **set_override_rate**(the name of r, 0.0) across polaris”

(the SET_OVERRIDE_RATE is a remote procedure declaration that changes the rate of the ramp meter in AIMSUN through the G2-AIMSUN interface. For removing the override, the rate 0 is passed to the G2-AIMSUN interface code, which removes the override from that particular ramp meter)

The remote procedure is declared as : -

```
declare remote set_override_rate(value, float)=()
```

1.17 STREAM_OR_DOWNSTREAM_CAMERA_DIALOG (PROCEDURE)

upstream_or_downstream_camera_dialog (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win: item-or-value, OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)

```
op : class op_section ;
```

```
DS : class camera ;
```

```
flg:integer = 0;
```

```
C: class camera-connection;
```

```
obj: symbol;
```

```
begin
```

```
for op = each op_section do
```

```
  if (the camera of op = the name of OrigDialogLauncher) then
```

```
    begin
```

```
      if ( the id of ThisDialogLauncher = "upstream") then
```

```
        begin
```

```
          C = the camera-connection connected at the input of op;
```

```
          if ( there exists an object connected at the input of C) then
```

```
            begin
```

```
              DS = the camera named by the camera of the op_section connected at the input of C;
```

```
            end
```

```
          end
```

```
        else
```

```
          begin
```

```
            C = the camera-connection connected at the output of op;
```

```
            if ( there exists an object connected at the output of C) then
```

```
              begin
```

```
                DS = the camera named by the camera of the op_section connected at the output of C;
```

```
              end;
```

```
            end;
```

```
          if DS exists then
```

```
            begin
```

```
              call uil-start-or-refocus-dialog ("Camera Information", DS, Win);
```

```
                flg = 1;
```

```
                exit if flg = 1;
```

```
            end
```

```
          end;
```

```
        end;
```

```
      if flg = 0 then
```



```

begin
  if (the id of ThisDialogLauncher = "upstream") then
    begin
      call uil-post-generic-dialog("There is no camera upstream to this camera", the symbol message,
the symbol large, Win, the symbol uil-information-icon)
    end
  else
    begin
      call uil-post-generic-dialog("There is no camera downstream to this camera", the symbol
message, the symbol large, Win, the symbol uil-information-icon)
    end;
  end;
end
end

```

1.18 STREAM_OR_DOWNSTREAM_VMS_DIALOG (PROCEDURE)

upstream_or_downstream_vms_dialog (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win: item-or-value, OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)

```

op : class op_section ;
DS : class vms ;
flg:integer = 0;
C: class vms -connection;
obj: symbol;
begin
for op = each op_section do
  if (the vms of op = the name of OrigDialogLauncher) then
    begin
      if ( the id of ThisDialogLauncher = "upstream") then
        begin
          C = the vms -connection connected at the input of op;
          if ( there exists an object connected at the input of C) then
            begin
              DS = the vms named by the vms of the op_section connected at the input of C;
            end
          end
        else
          begin
            C = the vms -connection connected at the output of op;
            if ( there exists an object connected at the output of C) then
              begin
                DS = the vms named by the vms of the op_section connected at the output of C;
              end;
            end;
          if DS exists then
            begin
              call uil-start-or-refocus-dialog ("Variable Message Sign", DS, Win);
              flg = 1;
              exit if flg = 1;
            end
          end;
        end;
      if flg = 0 then
        begin

```

```

    if (the id of ThisDialogLauncher = "upstream") then
    begin
        call uil-post-generic-dialog("There is no vms upstream to this vms", the symbol message, the
symbol large, Win, the symbol uil-information-icon)
    end
    else
    begin
        call uil-post-generic-dialog("There is no vms downstream to this vms", the symbol message, the
symbol large, Win, the symbol uil-information-icon)
    end;
end;
end
end

```

1.19 RULE FOR STARTING A REMOTE PROCEDURE FOR SENDING VMS MESSAGE TO AIMSUN

“whenever the message of any vms v receives a value then start set_vms_msg (the name of v, the message of v) across polaris”

(the SET_VMS_MSG is a remote procedure declaration which sets the message in the VMS in AIMSUN through the G2-AIMSUN interface)

The remote procedure is declared as : -

```
declare remote set_vms_msg(value, text)=()
```

1.20 UPSTREAM_OR_DOWNSTREAM_ZONE_DIALOG (PROCEDURE)

upstream_or_downstream_zone_dialog (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win: item-or-value, OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)

```
op: class op_section;
```

```
flg : integer = 0;
```

```
begin
```

```
for op= each op_section do
```

```
if (the zone of op = the name of OrigDialogLauncher ) then
```

```
begin
```

```
if ( the id of ThisDialogLauncher = "upstream") then
```

```
begin
```

```
if ( the upstream_zone_name of OrigDialogLauncher /= the symbol g2) then
```

```
begin
```

```
flg = 1;
```

```
call uil-start-or-refocus-dialog ("Zone Information", the zone named by the upstream_zone_name of OrigDialogLauncher , Win);
```

```
end
```

```
else
```

```
begin
```

```
call uil-post-generic-dialog("There is no zone upstream to this zone ", the symbol message, the symbol large, Win, the symbol uil-information-icon)
```

```
end;
```

```
end
```

```
else
```

```
begin
```

```
if ( the downstream_zone_name of OrigDialogLauncher /= the symbol g2) then
```

```
begin
```

```
flg = 1;
```

```
call uil-start-or-refocus-dialog ("Zone Information", the zone named by the downstream_zone_name of OrigDialogLauncher , Win);
```

```
end
```

```

        else
        begin
            call uil-post-generic-dialog("There is no zone downstream to this zone ", the symbol message, the
symbol large, Win, the symbol uil-information-icon)
            end;
        end;
        return;
    end;
end;
end

```

1.21 ZONE_TO_METER (PROCEDURE)

```

zone_to_meter (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win: item-or-value,
OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)
T:text;
r : class ramp -meter;
zone_name:text;
array_elemno:integer;
ramp_name:symbol;
begin
array_elemno = quantity (the id of ThisDialogLauncher);
zone_name = "[the name of OrigDialogLauncher]";
if (the ramp -meter named by (the zone_ramp_name
of OrigDialogLauncher)[array_elemno - 1] exists) then
begin
r = the ramp -meter named by (the zone_ramp_name
of OrigDialogLauncher)[array_elemno - 1] ;

call uil-start-or-refocus-dialog ("Ramp Meter Information", r, Win);
end
else
begin
call uil-post-generic-dialog("There is no ramp meter", the symbol message, the symbol large, Win, the
symbol uil-information-icon)
end;
end

```

1.22 REFRESH (PROCEDURE)

```

refresh(Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win: item-or-value,
OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)
begin
call uil-call-update-method-for-grobject (dlg, win)
end

```

1.23 NOTIFICATION-TO-CAMERA (PROCEDURE)

```

notification-to-camera(L:text, w: class kb-workspace, Win: item-or-value)
op, op1: class op_section;
u : class uil-medium-text;
C: class camera-connection;
N : class camera;
flg:integer =0;
T : text;
Cam : symbol;

```

```

begin
for u = each uil-medium-text upon w do
if ( the id of u = "camera") then
begin
T = the message-contents of u;
Cam = symbol(T);
for op= each op_section do
if ( the camera of op = Cam) then
begin
flg = 1;
exit if flg = 1;
end;
end;
if ( L = "up") then
begin
C = the camera-connection connected at the input of op;
op1 = the op_section connected at the input of C;
end
else
begin
C = the camera-connection connected at the output of op;
op1 = the op_section connected at the output of C;
end;
if ( the camera named by the camera of op1 exists)
then
begin
N = the camera named by the camera of op1;
call uil-start-or-refocus-dialog("Camera Information", N, Win);
end;
return;
end;
end;
end

```

1.24 FALSE-ALARM (PROCEDURE)

```

false-alarm( w: class kb-workspace)
begin
delete w;
end

```

1.25 INCIDENT-VERIFIED (PROCEDURE)

```

incident-verified(w: class kb-workspace)
u, v : class uil-medium-text;
T, L, K, N: text;
M : class uil-message-object;
S: symbol;
i :integer = 0;
j:integer = 1;
flg:integer = 0;
flg1:integer = 0;
ds: class detector-station;
op: class op_section;
begin
for u = each uil-medium-text upon w do
if ( the id of u = "location" ) then

```

```

begin
  T = the message-contents of u;
  for op = each op_section do
    if ( the location of op = T ) then
      begin
        flg1 = 1;
        exit if flg1 = 1;
      end;
    end;
  end
  else
    begin
      N = the message-contents of u;
    end;
  end;
  conclude that the accident of op = 1 ;
  for v = each uil-medium-text upon incident_form do
    if ( the id of v = "camera" ) then
      begin
        call uil-update-text(v, N);
      end
    else
      begin
        call uil-update-text(v, T);
      end;
    end;
  end;
  L = the text of the item-configuration of op;
  make op transient;
  K = insert-in-text("incident,", L, 103);
  change the text of the item-configuration of op to K;
  make op permanent;
  change the star icon-color of op to red;
  show the workspace of op;
  show incident_form;
tag1: for M = each uil-message-object upon remove-incident do
  if (the id of M = "[j]") then
    begin
      if (the message-contents of M = "none") then
        begin
          flg = 1;
          call uil-update-text(M, "[the location of op]");
          exit if flg = 1;
        end
      else
        begin
          j = j + 1;
          if j <= 5 then
            begin
              go to tag1;
            end;
          end;
        end
      end;
    end;
  end;
  delete w;
end

```

1.26 INCIDENT-FORM-INFO (PROCEDURE)

```
incident-form-info(w: class kb-workspace)
T: text;
S: symbol;
Inc: class incident;
Current_time : integer = the current real time;
Current_t : text ;
flg1: integer =0;
op : class op_section;
u : class uil-medium-text;
begin
for u = each uil-medium-text upon w do
if ( the id of u = "location" ) then
begin
T = the message-contents of u;
for op = each op_section do
if ( the location of op = T ) then
begin
flg1 = 1;
exit if flg1 = 1;
end;
end;
Inc = the incident of op;
conclude that the location of Inc = the location of op;
current_t= call g2-unix-time -to-text(current_time,true);
conclude that the start_time of Inc = the text of the readout-table-display-value of time-value;
conclude that the vehicle_type of Inc = the vehicle_type of CURRENT_INCIDENT;
conclude that the number_of_vehicles of Inc = the number_of_vehicles of CURRENT_INCIDENT;
conclude that the lanes_affected of Inc = the lanes_affected of CURRENT_INCIDENT;
conclude that the severity of Inc = the severity of CURRENT_INCIDENT;
conclude that the priority of Inc = the priority of CURRENT_INCIDENT;
conclude that the hazardous_material of Inc = the hazardous_material of CURRENT_INCIDENT;
conclude that the fire of Inc = the fire of CURRENT_INCIDENT;
conclude that the injuries of Inc = the injuries of CURRENT_INCIDENT;
conclude that the fatalities of Inc = the fatalities of CURRENT_INCIDENT;
conclude that the road_conditions of Inc = the road_conditions of CURRENT_INCIDENT;
conclude that the primary_source of Inc = the primary_source of CURRENT_INCIDENT;
conclude that the comments of Inc = the comments of CURRENT_INCIDENT;
hide w;
end;
end
end
```

1.27 UPSTREAMSTATION-FROM-INCIDENTDIALOG (PROCEDURE)

```
upstreamstation-from-incidentdialog (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win:
item-or-value, OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)
T:text;
opsection:symbol;
op, op1:class op_section;
C : class section-connection;
z: class zone;
ds : class detector-station;
v: class vms;
begin
```

```

T = the id of ThisDialogLauncher;
case(T) of
"upstation":
  begin
    opsection = the section-name of OrigDialogLauncher;
    op = the op_section named by opsection;

    tag1: C = the section-connection connected at the input of op;
    op1 = the op_section connected at the input of C;
    if ( the detector-station named by the mainline_detector of op1 exists) then
      begin
        ds = the detector-station named by the mainline_detector of op1 ;
        call uil-start-or-refocus-dialog ("Detector Station Information", ds, Win);
        return;
      end
    else
      begin
        op = op1;
        go to tag1;
      end;
    end;
"zone":
  begin
    opsection = the section-name of OrigDialogLauncher;
    op = the op_section named by opsection;
    z = the zone named by the zone of op;
    call uil-start-or-refocus-dialog("Zone Information", z, Win);
  end;
"vms":
  begin
    opsection = the section-name of OrigDialogLauncher;
    op = the op_section named by opsection;
    tag2: if ( the vms named by the vms of op exists) then
      begin
        v = the vms named by the vms of op;
        call uil-start-or-refocus-dialog ("Variable Message Sign", v, Win);
        return;
      end
    else
      begin
        C = the section-connection connected at the input of op;
        op1 = the op_section connected at the input of C;
        op = op1;
        go to tag2;
      end;
    end;
  end;
end
end

```

1.28 INCIDENT-CAM (PROCEDURE)

```

incident-cam (Dlg: class uil-dialog, ThisDialogLauncher: item-or-value, Win: item-or-value,
OrigDialogLauncher: item-or-value, DialogActions: class symbol-list)
T:text;
len: integer;
opsection: symbol;

```

```

C1, C2: class section-connection;
op, op1, op2: class op_section;
begin
  opsection = the section-name of OrigDialogLauncher;
  op = the op_section named by opsection;
  op1 = op;
  op2 = op;
tag1: if (( the camera of op1 /= the symbol g2) or (the camera of op2 /= the symbol g2)) then
  begin
    if ( the camera of op1 /= the symbol g2) then
    begin
      call uil-start-or-refocus-dialog ("Camera Information", the camera named by the camera of op1,
Win);
    end
    else
    begin
      call uil-start-or-refocus-dialog ("Camera Information", the camera named by the camera of op2,
Win);
    end;
  end
  else
  begin
    C1 = the section-connection connected at the input of op1;
    C2 = the section-connection connected at the output of op2;
    if ( there exists an object connected at the input of C1) then
    begin
      op1 = the op_section connected at the input of C1;
      end;
      if ( there exists an object connected at the output of C2) then
      begin
        op2 = the op_section connected at the output of C2;
        end;
      go to tag1;
    end;
  end
end

```

1.29 REMOVE_INCIDENT (PROCEDURE)

```

remove_incident(w : class kb-workspace, win: item-or-value )
i:integer;
S, M, K:text;
j:integer = 0;
flg:integer = 0;
u : class uil-message-object;
opsection:symbol;
Inc, incdt: class incident;
op: class op_section;
t: class text-array;
begin
  t = call uil-get-selected-message(w, "incsection") ;
  i = the array-length of t ;
  if (i = 0 or i > 1) then
  begin
    call uil-post-generic-dialog("Please select a section name", the symbol message, the symbol large,
Win, the symbol uil-information-icon);
    return;
  end;

```



```

end
else
begin
  K = t[0];
  for op = each op_section do
    if ( the location of op = K ) then
      begin
        flg = 1;
        exit if flg =1;
      end;
    end;
    Inc = the incident of op;
    change the star icon-color of op to transparent;
    conclude that the accident of op = 0;
    S = the text of the item-configuration of op;
    make op transient;
    M= omit-from-text(S, 103, 112);
    change the text of the item-configuration of op to M;
    make op permanent;
    create an incident Incdt by cloning Inc;
  conclude that the location of Incdt = the location of Inc;
  conclude that the start_time of Incdt = the start_time of Inc ;
  conclude that the vehicle_type of Incdt = the vehicle_type of Inc;
  conclude that the number_of_vehicles of Incdt = the number_of_vehicles of Inc;
  conclude that the lanes_affected of Incdt = the lanes_affected of Inc;
  conclude that the severity of Incdt = the severity of Inc;
  conclude that the priority of Incdt = the priority of Inc;
  conclude that the hazardous_material of Incdt = the hazardous_material of Inc;
  conclude that the fire of Incdt = the fire of Inc;
  conclude that the injuries of Incdt = the injuries of Inc;
  conclude that the fatalities of Incdt = the fatalities of Inc;
  conclude that the road_conditions of Incdt = the road_conditions of Inc;
  conclude that the primary_source of Incdt = the primary_source of Inc;
  conclude that the comments of Incdt = the comments of Inc;
  make Incdt transient;
  transfer Incdt to the subworkspace of op at (0,0);
  make Incdt permanent;
end;
for u = each uil-message-object upon w do
  if ( the message-contents of u = "[t[0]]" ) then
    begin
      call uil-update-text(u, "none");
    end;
  end;
  delete t;
end
end

```

1.30 INCIDENT-CAMERA (PROCEDURE)

```

incident-camera(w: class kb-workspace, Win: item-or-value)
i:integer= 0;
j:integer = 0;
u : class uil-message-object;
S: text;
op: class op_section;
t: class text-array;

```

```

begin
  t = call uil-get-selected-message(w, "incsection") ;
  i = the array-length of t ;
  if (i = 0 or i > 1) then
  begin
    call uil-post-generic-dialog("Please select an incident location", the symbol message, the symbol
large, Win, the symbol uil-information-icon);
    return;
  end
  else
  begin
    S = t[0];
    if S /= "none" then
    begin
      for op = each op_section do
      if ( the location of op = S) then
      begin
        j = 1;
        exit if j =1;
      end;
      end;
      call uil-start-or-refocus-dialog("Incident Information", the incident of op, Win);
    end;
  end;
end
end

```

1.31 CREATEINCIDENT (PROCEDURE)

createincident(event : symbol, tracked-window : class g2-window, trigger-item : item-or-value, tracked-item : item-or-value, x-mouse-position : integer, y-mouse-position : integer, event-timing-in-milliseconds : integer, state-of-modifier-keys : integer) = ()

```

w: class kb-workspace;
u : class uil-medium-text;
op1, op2: class op_section;
C1, C2: class section-connection;
flg: integer = 0;
M: class action-button;
begin
  if event is enter then
  begin
    create a kb-workspace w by cloning camera_view;
    op1 = trigger-item;
    op2 = trigger-item;
    for M = each action-button upon w do
    if (the label of M = "False Alarm" ) then
    begin
      flg = 1;
      exit if flg =1;
    end;
    end;
    conclude that the action of M = "start false-alarm(this workspace)";
    for u = each uil-medium-text upon w do
    if ( the id of u = "camera") then
    begin
      tag1: if (( the camera of op1 /= the symbol g2) or (the camera of op2 /= the symbol g2)) then
      begin

```

```

if ( the camera of op1 /= the symbol g2) then
begin
call uil-update-text(u, "[the camera of op1]");
end
else
begin
call uil-update-text(u, "[the camera of op2]");
end;
end
else
begin
C1 = the section-connection connected at the input of op1;
C2 = the section-connection connected at the output of op2;
if ( there exists an object connected at the input of C1) then
begin
op1 = the op_section connected at the input of C1;
end;
if ( there exists an object connected at the output of C2) then
begin
op2 = the op_section connected at the output of C2;
end;
go to tag1;
end;
end
else
begin
call uil-update-text(u, the location of trigger-item);
end;
end;
end;
show w;
end

```

1.32 REPORT_INCIDENT (PROCEDURE)

```

report_incident(w: class kb-workspace)
u : class uil-medium-text;
S: class kb-workspace;
op: class op_section;
cam: symbol;
M : class action-button;
i: integer = 0;
flg: integer = 0;
T: text;
begin
T = the message-contents of the uil-edit-box-medium upon w;
cam = symbol(T);
for op= each op_section do
if ( cam = the camera of op) then
begin
i = 1;
exit if i =1;
end;
end;
create a kb-workspace S by cloning camera_view;
for M = each action-button upon S do

```

```

if (the label of M = "False Alarm" ) then
begin
  flg = 1;
  exit if flg = 1;
end;
end;
conclude that the action of M = "start false-alarm(this workspace)";
for u = each uil-medium-text upon S do
if ( the id of u = "camera") then
begin
  call uil-update-text(u, "[the camera of op]");
end
else
begin
  call uil-update-text(u, the location of op);
end;
end;
hide w;
show S;
end

```

1.33 FIND_OBJECT (PROCEDURE)

```

find_object(w: class kb-workspace, Win: item-or-value )
obj_workspace: class kb-workspace;
object_name: text;
object_type: text;
d: class object;
u : class uil-edit-box-medium;
symobjname: symbol;
obj: class object;
flg: integer = 0;
begin
for u = each uil-edit-box-medium upon w do
  if the id of u = "objname" then
  begin
  object_name = the message-contents of u;
  end
  else
  begin
  object_type = the message-contents of u;
  end;
end;
end;

```

```

symobjname = symbol(object_name);
obj = the object named by symobjname;
object_type = lower-case-text (object_type);
if (there exists an object named by symobjname) then
begin
case(object_type) of
  "detectors", "detector":
  begin
  for d = each detector do
  if the name of d = symobjname then
  begin
  flg = 1;

```

```

    obj_workspace = the workspace of obj;
    show obj_workspace;
    exit if flg = 1 ;
end
end;
end;
"detector stations", "detector station", "detector-station", "detector-stations", "detector_stations",
"detector_station":
begin
for d = each detector-station do
if the name of d = symobjname then
begin
    flg = 1;
    obj_workspace = the workspace of obj;
    show obj_workspace;
    exit if flg = 1 ;
end
end;
end;
"ramp meter", "ramp -meter", "ramp -meters", "ramp meters" :
begin
for d = each ramp -meter do
if the name of d = symobjname then
begin
    flg = 1;
    obj_workspace = the workspace of obj;
    show obj_workspace;
    exit if flg = 1 ;
end
end;
end;
"zone", "zones":
begin
for d = each zone do
if the name of d = symobjname then
begin
    flg = 1;
    obj_workspace = the workspace of obj;
    show obj_workspace;
    exit if flg = 1 ;
end
end;
end;
"vms", "variable message sign", "variable message signs":
begin
for d = each vms do
if the name of d = symobjname then
begin
    flg = 1;
    obj_workspace = the workspace of obj;
    show obj_workspace;
    exit if flg = 1 ;
end
end;
end;
end;

```

```

"opsections", "opsection", "operating section", "operating sections", "operating-section" ,
"operating-sections":
begin
  for d = each op_section do
  if the name of d = symobjname then
  begin
    flg = 1;
    obj_workspace = the workspace of obj;
    show obj_workspace;
    exit if flg = 1 ;
  end
  end;
end;
end;
end
else
begin
  call uil-post-generic-dialog("Please check the entered name", the symbol message, the symbol large,
Win, the symbol uil-information-icon)
  end;
end
end

```

1.34 USEROK (PROCEDURE)

```

userok(w : class kb-workspace, win: item-or-value )
userid:text;
u : class uil-edit-box-medium;
i:integer;
j:integer = 0;
usermode:symbol;
user: class user-info;
t: class text-array;
flg:integer = 0;
begin
  userid = the message-contents of the uil-edit-box-medium upon w;
  for user = each user-info do
  if (the names of user = symbol(userid) ) then
  begin
    t = call uil-get-selected-message(w, "modename");
    i = the array-length of t ;
    if (i = 0 or i > 1) then
    begin
      j = 1;
      call uil-post-generic-dialog("Please select a user mode", the symbol message, the symbol large, Win,
the symbol uil-information-icon);
      exit if j =1;
    end
  else
  begin
    flg = 1;
    usermode = symbol("[t[0]]");
    conclude that the g2-user-mode of win =usermode;
    hide w;
    show action-sequence;
  end
  end
  end
end

```

```
show i-35w_4;
exit if flg = 1;
end;
end;
end;
if flg = 0 then
begin
  call uil-post-generic-dialog("Please recheck your userid", the symbol message, the symbol large,
Win, the symbol uil-information-icon);
  end;
  delete t;
end
```