

UC Berkeley

Research Reports

Title

TRICEPS: An ATMIS Field Implementation for Control and Evaluation: Final Report

Permalink

<https://escholarship.org/uc/item/71h9100q>

Authors

McNally, M. G.

Rindt, C.

Logi, F.

Publication Date

2002-03-01

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

TRICEPS: An ATMIS Field Implementation for Control and Evaluation: Final Report

MG McNally

C. Rindt

F. Logi

California PATH Research Report

UCB-ITS-PRR-2002-9

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California. This report does not constitute a standard, specification, or regulation.

Final Report for MOU 346

March 2002

ISSN 1055-1425

PATH MOU 346

**TRICEPS:
An ATMIS Field Implementation for Control and Evaluation**

FINAL REPORT

MG McNally, C.Rindt, and F.Logi

Institute of Transportation Studies and
Department of Civil & Environmental Engineering
University of California Irvine
Irvine, CA 92697-3600

November 2001

Table of Contents

Abstract	2
Keywords	2
Chapter 1: Overview	3
1.1 The UCI Caltrans ATMIS Testbed	3
1.2 A Testbed Distributing Computer Architecture	3
1.3 The TRICEPS Architecture	4
1.4 CARTESIUS	4
Chapter 2: TRICEPS Architecture	6
2.1 Introduction	6
2.2 Design Concepts and Terminology	6
2.3 Requirements	8
2.4 Design Specification	9
2.5 Implementation	11
2.6 Summary	17
Chapter 3: CARTESIUS	19
3.1 Overview	19
3.2 CARTESIUS	19
3.3 CARTESIUS Architecture	20
3.4 TCM: Traffic Congestion Manager	24
3.5 The Distributed Algorithm	25
3.6 CARTESIUS Summary	28
Chapter 4: TRICEPS Simulation Components	29
4.1 Overview of Simulation Model Components	29
4.2 Development of a Hybrid Simulation Framework	31
4.3 Encapsulation of the Hybrid Simulator as a TRICEPS Module	33
Chapter 5: TRICEPS / CARTESIUS Evaluation	34
5.1 Introduction	34
5.2 Evaluation of System Performance	34
5.3 The Test Site	35
5.4 Evaluation Approach	36
5.5 The Simulated Environment	37
5.6 Evaluation Results	38
5.7 Evaluation Summary	50
Chapter 6: Sample Application of TRICEPS	55
6.1 System Monitoring and Incident Detection (slides 1-3)	56
6.2 Initial Response and Inter-agent Communication (slides 4-7)	59
6.3 Generate Response Strategies (slides 8-13)	63
6.4 Strategy Negotiations (slides 14-18)	69
6.5 Strategy Selection and Implementation (slides 19-23)	74
6.6 Incident Recovery and System Monitoring (slides 24-28)	79
Chapter 7: Summary and Future Research	84
7.1 Summary	84
7.2 Future Research	84
References	85

PATH MOU 346

**TRICEPS:
An ATMIS Field Implementation for Control and Evaluation**

MG McNally, C.Rindt, and F.Logi

Institute of Transportation Studies and
Department of Civil & Environmental Engineering
University of California Irvine
Irvine, CA 92697-3600

ABSTRACT

This report summarizes a comprehensive research project directed toward the development and implementation of an Advanced Transportation Management and Information System (ATMIS) as part of the Caltrans Advanced ATMIS Testbed Program at the Institute of Transportation Studies, University of California, Irvine.

The primary goal of this project was to implement this prototype ATMIS, designated **TRICEPS** (**T**estbed **R**ealtime **I**ntegrated **C**ontrol and **E**valuation **P**rototype **S**ystem), in the Irvine sub-area of the Advanced Testbed network. This sub-area represents a well-defined freeway corridor with a parallel major arterial alternative where traffic demand is predominantly along the corridor. This area, previously selected as the site for the federally-funded Irvine Field Operational Test to evaluate a centrally-controlled freeway/arterial corridor, thus provided an ideal environment in which to conduct the enhancements to and implementation of **TRICEPS**. The **TRICEPS** architecture allows for the introduction of a wide range of control and management capabilities, however, the focus of the initial implementation is the development and implementation of **CARTESIUS**, a real-time, multi-agent decision support system which integrates real-time control and simulation elements.

The report provides background information on the genesis of **TRICEPS**, an overview of each of its major components and of the system architecture, the results of a system evaluation study, and a sample application of the model system.

Key Words: ATMIS, **TRICEPS**, **CARTESIUS**, multi-agent corridor traffic control, congestion management, realtime data.

Chapter 1

OVERVIEW

1.1 The UCI Caltrans ATMIS Testbed

The Advanced Transportation Management Information Systems (ATMIS) Testbed sponsored by the California Department of Transportation (Caltrans) is an ongoing multi-year research and implementation project at the Institute of Transportation Studies, Irvine (ITS). The Testbed includes computerized laboratories, where algorithms are developed and tested, and the connection to the real world transportation system, through a state-of-the-art data communication network. The Testbed provides an environment for the development, implementation, and evaluation of advanced transportation management strategies and technologies.

The laboratory is a testing ground for the development of particular ATMIS *modules* and of integrated ATMIS *applications*. An ATMIS module is an algorithm that processes data to produce a particular type of output (e.g., a traffic control algorithm). An ATMIS application is a particular configuration of ATMIS modules that are integrated to manage and control transportation system operations. Thus, an ATMIS application is a distributed algorithm with various modules performing various tasks.

1.2 A Testbed Distributing Computer Architecture

Based on these concepts, a distributed computing platform was designed to implement ATMIS applications from existing Testbed research components. The first generation of this idea, based upon the UCI Distributed Algorithm Testing Environment (ELUCIDATE), demonstrated proof of the concept by integrating a series of analysis modules which were used in the validation of a traffic congestion management module using simulated data, but lacked the robustness needed for general application. This initial attempt at developing an implementation platform led to an architecture paradigm shift from ELUCIDATE to a commercial CORBA implementation (using the Orbix Object Request Broker). This second (current) generation implementation platform has proven to be more robust, and could thus be used for the evaluation of transportation management algorithms.

The objective of the Testbed was to construct an implementation platform that provides it with “plug and play” capabilities for the testing and evaluation of ATMIS modules. Such modules can be configured in an existing or a new ATMIS application, simply by developing the interfaces required for their

connection with other modules, without the need to modify or develop the required additional infrastructure. Any particular ATMIS application is connected to both simulated and real-world data, so that its effectiveness can be first assessed in the laboratory and then evaluated in the field. This concept has been implemented with the development of **TRICEPS** (Testbed Real-time Integrated Control and Evaluation Prototype System).

The results of the various projects of the Testbed research phase have been documented (Recker *et al.*, 1997). The development of the initial version of **TRICEPS**, including the role of the Irvine FOT project and the selection of system components was summarized in a preceding report (McNally *et al.*, 1999).

1.3 The TRICEPS Architecture

TRICEPS (Testbed Real-time Integrated Control and Evaluation Prototype System) consists of the control subsystem of the Testbed ATMIS workbench and a set of evaluation tools. **TRICEPS** is structured to interface both with real-time data provided through the Testbed's ATMIS real-time data interface as well as with simulation data provided by the Testbed's traffic simulation software. The architecture of **TRICEPS** allows for the introduction of a full range of current and evolving control and management techniques. The Testbed involves a number of local and regional transportation agencies including Caltrans District 12 (Orange County), the City of Anaheim, and the City of Irvine. Within the interface architecture, CORBA clients were developed, that provide the ATMIS applications with the transportation management infrastructure of these agencies. Data from these external agencies include standard inductance loop detectors (ILD), traffic signal and ramp meter parameters, and changeable message signs (CMS).

1.4 CARTESIUS

One of the key components of the **TRICEPS** platform is a distributed environment for the provision of real-time decision support to Transportation Management Center (TMC) operators, that provides a set of core transportation management applications for multi-jurisdictional traffic control and incident management. Indeed, the spatial and administrative organization of transportation management agencies in metropolitan networks requires a coordinated solution effort that preserves the different levels of authority, guarantees privileged data control, and in general reflects the inherent distribution of the decision-making power. A coordinated response to congestion avoids the implementation of operations that may otherwise conflict, and therefore be counter-productive. To address such issues, the multi-agent real time system **CARTESIUS** (Coordinated Adaptive Real-Time Expert System for Incident management on Urban Systems) was developed and incorporated within the **TRICEPS** platform. **CARTESIUS** employs

advanced cooperation and conflict resolution methodologies for coordinated traffic management operations among multiple agents.

The **TRICEPS / CARTESIUS** platform is designed to work in three operational modes which make it an extensible ATMISS that can optimize, control, and manage real-world traffic, as well as allow for the investigation of individual ATMISS technologies without relying on field implementation of the detection and sensor hardware:

1. *Simulation Mode* provides an interface between **CARTESIUS** and data from two traffic simulators that provide microscopic, low-level sensor data and are particularly suited for modeling driver response to the provision of traffic information. Simulated data is used for testing (prior to implementation) and for data synthesis, when complete coverage of real-data is not available.
2. *Real-time Mode* uses a real-time, CORBA-based (Common Object Request Broker Architecture) data communication link with the California Department of Transportation (Caltrans) District 12 (Orange County) data server which provides 30-second measurements from loop detectors, the current state of ramp meters and Changeable Message Signs on the Orange County freeway network, and video camera data. Real-time Mode with real-time data is the prototype interface for real-time traffic management.
3. *Integrated Mode* allows the simulation of near-future traffic states based on current conditions for the evaluation of alternative traffic control response schemes. This mode involves initializing and continuously synchronizing simulation with real-time data and performing faster-than-real-time simulations before or during the implementation of actual responses to traffic conditions.

The operability of the three operational modes has been, thus far, only partially tested. Completed tasks include the validation of the data communication interface, the validation of the simulation tools, and the evaluation of an AID algorithm. The **TRICEPS** platform was used in simulation mode to validate the ability of the multi-decision-maker algorithm in **CARTESIUS** in providing effective traffic control response to the occurrence of incidents. The environment was used to create a wide range of incident test scenarios based on which a quantitative and qualitative evaluation of the algorithms was performed. Results of the evaluation demonstrate the validity of the **CARTESIUS** approach in reducing congestion both at the local and network-wide level.

Chapter 2

TRICEPS Architecture

2.1 Introduction

TRICEPS, the Testbed Real-time Integrated Control and Evaluation Prototype System, is an expression that has been used to refer to several conceptualizations of the UCI Testbed. For the purposes of this document, however, **TRICEPS** refers to specific software components resident in the Testbed laboratories. The Testbed laboratories were envisioned to provide a platform for transportation systems researchers to conceive of, implement, test, and evaluate various tools for transportation systems management. Typically, these tools are software modules that perform specific tasks, ranging from monitoring and analysis to management and control.

Based on these concepts, we identified several requirements for the implementation and testing stages of transportation management applications. First, each module, whether it is an O/D estimation algorithm, an incident detection algorithm, or a control algorithm, generally depends upon other modules for data. Each individual module, however, is usually the product of focused individual research conducted in relative isolation and is therefore likely to be somewhat heterogeneous with respect to any existing modules in the system. Second, we recognized the need for a “simulation workbench,” a simulation to which we could connect the modules, test their behavior, and ultimately evaluate their performance. Several Testbed simulators were available for this purpose (e.g., Dynasmart and Paramics) and it was desirable to be able to interface a variety of research algorithms with them cleanly without requiring alterations to the simulator’s internal source code. Finally, we wanted to be able to make use of the connections we were developing to the adjacent real-world transportation systems in the Testbed to ultimately test and evaluate the modules in real-world system settings.

This remainder of this chapter describes the work on the **TRICEPS** platform that was conducted under MOU346. This work helped evolve the platform from an experimental product, to a stable development platform for ATMIS.

2.2 Design Concepts and Terminology

TRICEPS is a platform for developing, testing, and evaluating distributed transportation systems analysis and control applications in simulated and real-world settings. We term a particular distributed application that is implemented using **TRICEPS** to be a candidate Advanced Transportation Management and Information System (ATMIS). Thus, we might speak of a particular ATMIS

candidate, say ATMIS-x that we implemented using **TRICEPS**, and which we tested and evaluated using the simulated and real-world data that is accessible by **TRICEPS**. Furthermore, because some modules may not be directly transferable between urban areas, we may speak of an implementation of candidate ATMIS-x for a particular location, y, as ATMIS-x-y. In the broadest sense, an ATMIS must utilize sensor data from the transportation system to determine a set of control actions to meet some objective. A particular ATMIS candidate comprises a set of modules, with each module responsible for performing a particular set of tasks. There is a basic flow of data through the various modules in the system as shown in Figure 2.1. Low-level sensor data from the transportation system, from sources such as Inductance Loop Detectors (ILDs), Closed Circuit Television (CCTV), and probe vehicles, flow into a set of analysis modules. These analysis modules process the data into higher-level state measurements such as facility travel time, traffic density, and incident codes. These data subsequently feed a set of estimation and prediction modules, such as OD estimation and traffic assignment algorithms, which seek to anticipate future conditions in the system. These predictions, in turn, feed management modules that strategically guide a final set of algorithms that control the transportation system. The estimation, management, and control modules are connected via feedback loops. This allows, for instance, a management module to evaluate potential strategies using estimation modules and compare the predictions to the currently predicted (do-nothing) outcome.

The modules in an ATMIS are therefore interdependent; A researcher working on a module in one area may require other modules to exist in order to evaluate the research. One of the initial promises of the Testbed labs was to produce a platform that included a set of working algorithms with which researchers could interface their own particular work, thus alleviating the burden of reproducing that work themselves. **TRICEPS** is a significant step in this direction.

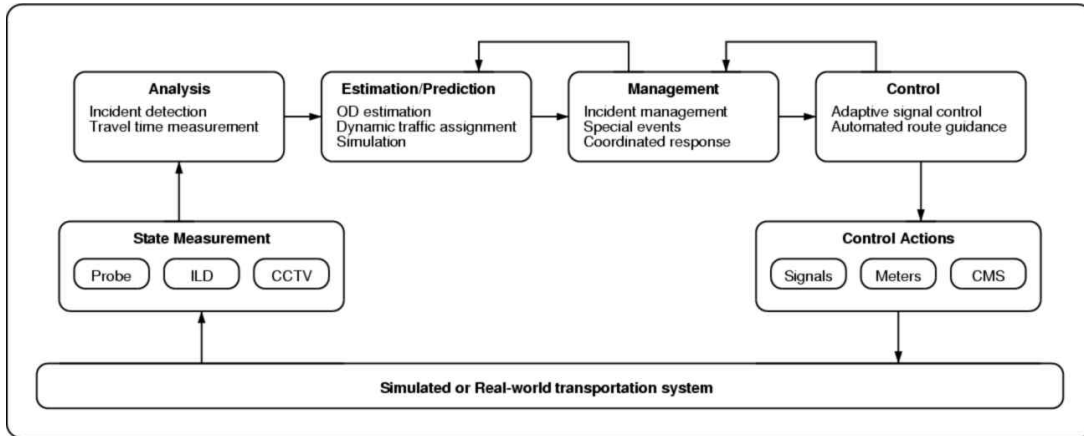


Figure 2.1: ATMIS components and data flow

2.3 Requirements

The previous section identified certain parameters that were considered necessary for **TRICEPS**. In this section, we formalize those parameters into a set of explicit requirements that defined the development of **TRICEPS**.

Research flexibility:

TRICEPS is designed around a research-centric philosophy. In this environment, researchers frequently work to solve specific sub-problems related to transportation system management. A major hurdle in such research efforts is evaluating the research as part of the complete system it is intended to augment. **TRICEPS** strives to allow researchers the ability to place their candidate algorithms in the management and control loop of a simulated or real-world transportation system.

Standardization:

Toward this end, **TRICEPS** defines a set of data structures and interfaces for representing and transportation system data and sharing it between modules.

Consistency:

To share data between modules, **TRICEPS** provides a common naming mechanism and utilizes standardized data structures to establish consistent interfaces to modules.

Portability

To allow researchers the most flexibility in their research, the system needed to be portable and support a range of hardware and operating system combinations. Selection of a non-portable platform might ultimately prevent future researchers from connecting their work to the system. Open standards and communications protocols were therefore considered mandatory.

Ease of use:

To facilitate the quick addition of components to the system, we felt that it should be easy to encapsulate software components for use with **TRICEPS**. Some of this ease of use depends on the portability of the platform, but also on the design of the system.

2.4 Design Specification

TRICEPS consists of three main components:

1. the Transportation Algorithm Interface Library (TAIL);
2. the **TRICEPS** Module Library (TML), and;
3. the **TRICEPS** Management Tool (TMT).

Figure 2.2 shows how these three components relate to each other, and to the real-world or simulated transportation system via the communications network infrastructure. The Transportation Algorithm Interface Library (TAIL) serves as a high-level interface to the underlying communications libraries, encapsulating data translation between global representations of system objects and representations internal to the module. In short, it is the layer between the communications interface and a module implementation that standardizes the way in which the module's objects map to global objects, and establishes dependencies to objects in the system (e.g., whether they use simulated or real object implementations). The Transportation Algorithm Interface Library (TAIL) is intended to facilitate the encapsulation of existing algorithms into the **TRICEPS** environment, releasing researchers from the burden of using particular data structures.

The **TRICEPS** Module Library (TML) consists of a bundle of **TRICEPS** compliant modules that respond to published interfaces to provide particular ATMIS services to other modules. The utility of **TRICEPS** as an implementation and testing platform is directly proportional to the size of the **TRICEPS** Module Library (TML). Obviously, the more modules that are available, the more candidate ATMIS configurations can be considered. At a minimum, the TML must contain modules for basic sensor data processing (e.g., loop detector data fusion), state estimation (e.g., travel demand estimation), congestion management (e.g., an incident management tool), and basic, parameterized traffic control algorithms (e.g., route guidance).

The **TRICEPS** Management Tool (TMT) is a set of programs for configuring **TRICEPS** to utilize a given set of modules as a particular candidate ATMIS. The TMT should provide an intuitive user interface to **TRICEPS** including:

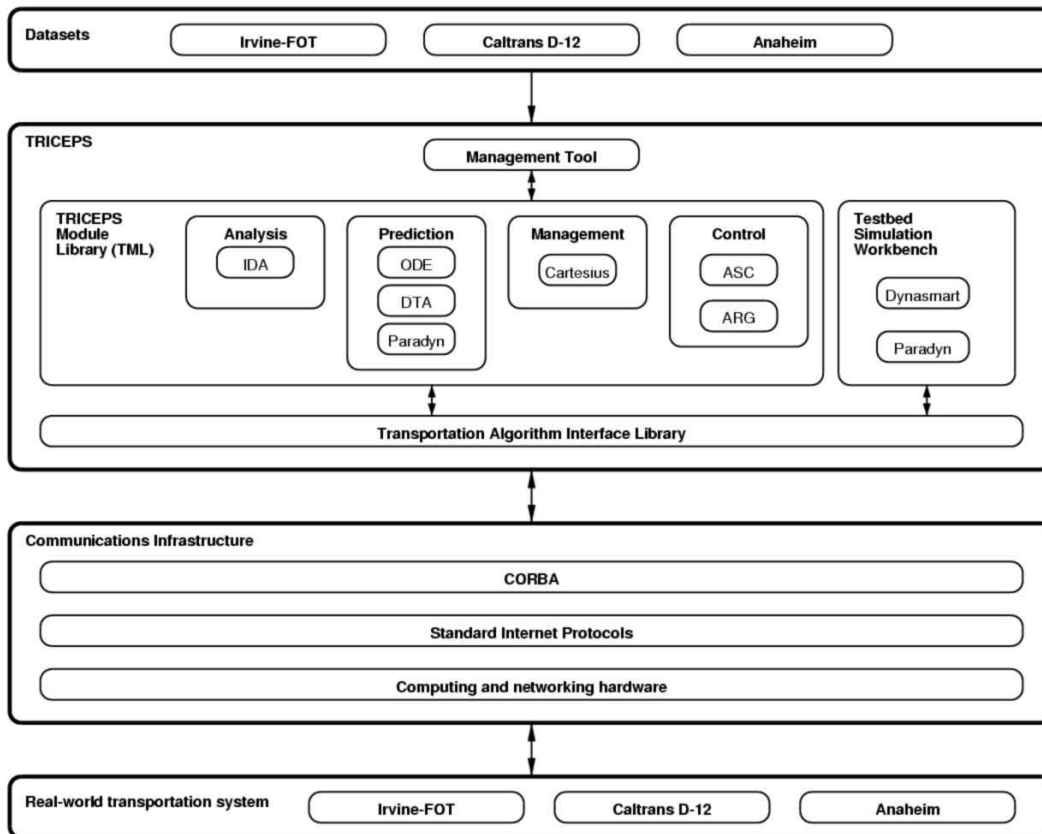


Figure 2.2: Relationships between TRICEPS Components

1. Various tools for editing module data sets, perhaps via a centralized database.
2. An ATMIS configuration capability allowing the specification of an ATMIS to evaluate and the data interdependencies between modules in the ATMIS and between modules and transportation system objects.
3. An interface for analyzing and visualizing the output from individual modules and the simulated or real-world transportation system.

Within this framework, evaluating a candidate ATMIS configuration (that is, a particular combination of transportation systems management and analysis modules) involves the following steps:

1. Identify the functional requirements for the ATMIS.
2. Identify the associated modules in the TML.
3. Encapsulate any additional algorithm(s) that are needed in the ATMIS, but not yet available in the TML, with TAIL library objects. This will allow the algorithm(s) to be used as **TRICEPS** modules.
4. Identify the data sets necessary to analyze the features of the transportation system of interest.
5. Configure each selected module ATMIS for use with the data set.

6. Execute simulation.
7. Analyze results based on selected performance criteria (e.g., total delay, worst delay, emissions, etc.)

2.5 Implementation

Given the specifications from section 4, we now turn to the details of the **TRICEPS** implementation for the Testbed laboratories. Because it is the core component of the platform, we begin with the TAIL, and then discuss the base modules included in the TML, and finally the management programs in the TMT.

2.5.1 TAIL

Earlier generations of **TRICEPS** relied upon distributed algorithm environments including UC Irvine Distributed Algorithm Testing Environment (*ELUCIDATE*) and Parallel Virtual Machine (PVM) that are more suitable for the design and optimization of specific distributed algorithms, than for the integration of a set of interacting, but relatively distinct, analysis modules. During the evolution of the **TRICEPS** platform, significant advances have been made in networked object technology. In particular, technologies such as CORBA, Distributed Component Object Model (DCOM), and a variety of web-based protocols have emerged as very usable tools for data sharing between software components. Of these, CORBA is arguably the most robust and complete network object architecture. Relative to the specifications discussed earlier, its features include a focus on interoperability and platform independence. Interoperability is achieved through two parts of the CORBA specification:

1. **Interface Definition Language (IDL)** provides the means for standardized object interfaces (whether the objects are simply data or algorithmic functions), and
2. **Interface Interoperability (IIOP)** defines a standard, vendor independent protocol for requesting data from a CORBA object.

CORBA's built-in features provide most of the general requirements identified in section 3. In particular, the architecture allows for portability, data and interface consistency, and provides the basic building blocks (via IDL) for standardizing transportation system data. These features jointly permit significant research flexibility. CORBA's only shortcoming with respect to these requirements is its complexity. Because it provides much of the necessary functionality, however, the remainder of the TAIL library can focus primarily on improving the ease of use. The TAIL consists of C++ helper classes that simplify the encapsulation of software modules and transportation system objects. The library consists of (a) Transportation system IDL object interfaces, (b) Server-side C++ helper classes, and (c) Client-side C++ helper classes.

Transportation System IDLs

A set of IDL files in the TAIL defines the interfaces to the basic objects that are either presently part of, or anticipated to be part of, all transportation management systems. Figure 2.3 shows these core transportation system interfaces, which are broken down into two basic classes: (a) Devices and (b) Device Factories. Devices are named objects that exist in and interact with the transportation system. There are two basic types of Device. Sensors, such as loop detectors (VDSs), take measurements from the transportation system while Controllers, such as Intersection Controller Units (ICUs), manipulate components of the transportation system. A particular device can be both a Sensor and a Controller (e.g., a Probe Vehicle Device (PVD)).

A Device Factory acts as a lightweight naming service (a standard CORBA service that associates names with object references), which a server implements to provide access to the device servant objects it implements. A server might be a real-world device (such as a central controller that coordinates multiple sensors in an urban area), a database that collects information from multiple sensors, a simulation representing the real world and the devices in it, or an algorithm in an ATMIS configuration. For instance, to encapsulate a simulation model for use with **TRICEPS**, we must define a Device Factory that “wraps” the simulation. This device factory wrapper must instantiate servants that implement the Device interfaces for each Detector Station (VDS), Ramp Meter Station (RMS), ICU, and Changeable Message Sign (CMS) in the simulated transportation system. The Device Factory interface provides methods for obtaining references to transportation devices objects so that their methods can be invoked to obtain data from or control them.

The TAIL includes client- and server-side object adapters generated from these IDLs. These automatically generated objects implement the CORBA layer of the TAIL, performing most of the heavy lifting that supports network communication. The details are available in the CORBA specification. Very simply, however, the framework allows client programs to obtain remote references to software objects sitting on servers on different machines across the network. The client-side reference is simply an object whose method implementations are remote procedure calls to the corresponding methods on the server-side implementation of the object. Thus, once the client has a reference to the Device, it can invoke that Device’s methods as if that Device was sitting on the local machine.

For instance, the **TRICEPS** simulator discussed in Chapter 4 implements, or is encapsulated by, four device factories, one for each of the four device types the simulation models: loop detectors (VDS), ramp meters (RMS), intersection controllers (ICU), and probe vehicles (PVD). When the simulation is started, the device factories are instantiated, creating servant objects for each device in the network being simulated. As CORBA servants, these factories are accessible from the CORBA object request broker by name. A module in the candidate ATMIS being evaluated using **TRICEPS**, such as the **CARTESIUS** module discussed

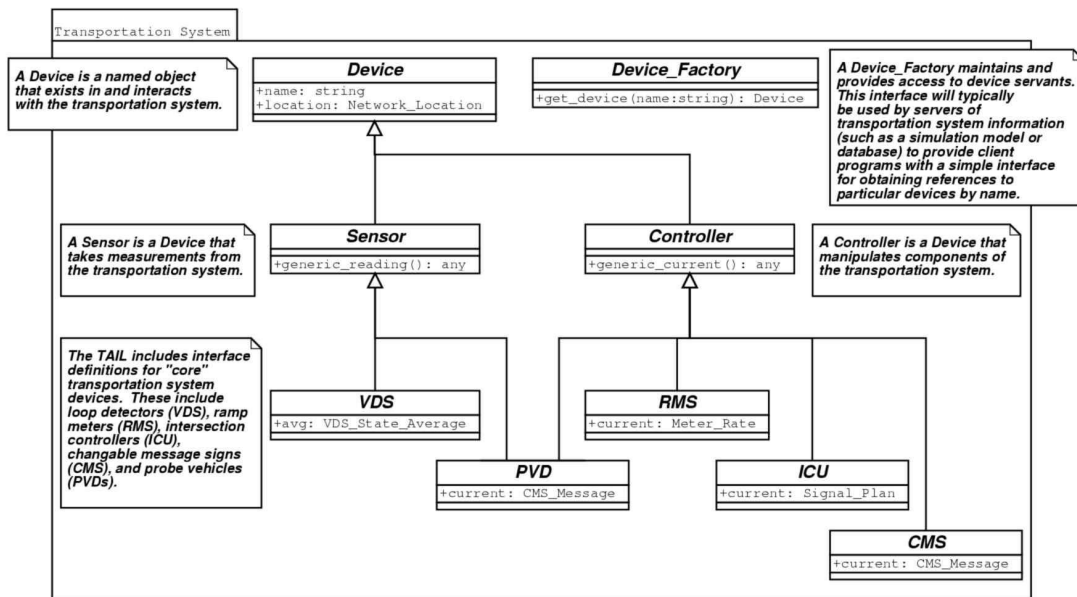


Figure 2.3: TAIL Interface Definitions for Transportation System Objects

in Chapter 3, can obtain a reference to a device servant, say a VDS, implemented by the simulation as follows. First, **CARTESIUS** ask the CORBA naming service for an object reference to the simulator's VDS device factory. Then, **CARTESIUS** must invoke the VDS factory's `get_device()` method to obtain an object reference to a particular VDS by name. Once **CARTESIUS** has this reference, it can invoke its methods, defined in the IDL interface definition, to obtain volume, occupancy, and speed at the station. In pseudo-code, the client side looks something like:

```

factory_ref := obtain factory reference from naming service
vds_ref = factory_ref->get_device("vds device name")
print vds_ref->volume(), vds_ref->speed()

```

The client and server side stubs generated from the IDL are generic. They simply provide an interface skeleton and the low-level machinery for remote method invocation. How the interface is implemented on the server side, and how it is used on the client side, is left unspecified. The remainder of the TAIL is intended to simplify and standardize these implementations.

Server-side C++ helper classes

The TAIL provides an abstract base class implementation for Device Factories that encapsulates and standardizes the operation of Device Factories in **TRICEPS**. This encapsulation automates lower-level CORBA library operations, including

the binding of servant objects with the naming service and associated error handling, within a simple high-level framework. This framework comprises two functions: one to load the set of servant objects managed by the device factory, and a second to destroy those objects. The `load_objects` function is abstract with different implementations for each module with which it is used. For instance in the startup process mentioned above for the simulation module, the `load_objects` method is implemented to loop over all system detectors in the simulation, creating VDS Device servant objects for each one via the `activate_object_with_name()` method provided by the server-side template. Other automated functionality includes:

1. *naming service functions* (for instance, the *Device Factory* binds all of its objects under a specific namespace associated with the factory)
2. *device list method* (for obtaining a list of all devices a particular server manages, and
3. Error handling.

In general, these service-side helper classes greatly simplify module encapsulation by hiding the CORBA-level details of implementing an object factory. The researcher need only implement the `load_objects()` method for his or her software module in order to instantiate servants for the objects the module provides.

Client-side C++ helper classes

Obtaining client-side references in CORBA is relatively straightforward. The process, however, involves a significant amount of duplicated code for connection establishment and related error handling. The TAIL therefore includes a C++ template client wrapper class that handles the CORBA level fundamentals and provides a simplified interface for obtaining and manipulating client reference to a servant object. Basic features include:

1. standard methods for obtaining a client reference from particular factory (and names-pace)
2. methods for dumping the state of an object (e.g., the current readings from a sensor) to an output stream(e.g., a file)
3. methods for updating the state of a controller (e.g., a new rate at a ramp meter) by reading from an input stream, and
4. direct access to the underlying object reference, narrowed to the appropriate type, so that methods specific to the object type can be accessed (e.g., obtaining the volume at a detector for use in a transportation management algorithm).

Because the client-side helper classes are implemented as templates, they can quickly be adapted to handle new data types that a particular research might add to the core TAIL interface definitions. For instance, a researcher may develop an OD estimation algorithm and a related interface (and data types) for obtaining

demand estimates. Once the researcher specifies an IDL for the interface, adds it to the TAIL IDL, and encapsulates his or her research software as a **TRICEPS** module using helper classes described above, other **TRICEPS** modules can obtain quick and easy access to the newly created objects with coding effort equivalent to the client-side pseudo code discussed above.

2.5.2 TRICEPS Module Library

The TML is a collection of modules already encapsulated for use with **TRICEPS** that implement portions of ATMIS functionality. The next sections discuss current modules in terms of the ATMIS components outlined in Figure 2.1.

State measurement and control

Recall from the preceding section that the TAIL defines a core set of IDLs for sharing transportation system data. A key source of data is obviously the transportation system that a candidate ATMIS will manage and be evaluated upon. Making that data available requires the specification of software server objects that implement TAIL-compliant interfaces. Currently, there are two potential sources for transportation system state measurement: (a) real-world transportation system data from Caltrans District 12, and (b) simulated transportation system data from the components described in Chapter 4.

The Caltrans-D12 implementation involves TAIL compliant, CORBA access to VDS, RMS, and CMS that are in the Testbed network. The VDS interface is limited to 30-second speed, volume, and occupancy readings, the RMS interface permits read and write access to ramp metering rates, and the CMS interface allows read and write access to displayed messages in the network. Note that the write access to RMS and CMS is limited to only tests approved by Caltrans. A prototype implementation of the PVD device has also been tested.

The Paradyn simulation workbench implementation allows for a more complete set of state measurements. In addition to unencumbered read/write access to VDS, RMS, and CMS devices in the simulated Testbed network, there are interfaces to all simulated ICU and PVD devices. This combination of 5 device types permits a candidate ATMIS complete access to all current, and some potential, state measurement and control devices available for transportation systems management.

Analysis

The main analysis feature provided by the TML is straightforward data fusion capabilities. This functionality is actually embedded in the TAIL client-side template classes of the TAIL and provides rudimentary capabilities for prioritizing the sensor sources. The helper class allows the client to specify a prioritized list of possible sources for a particular device. The helper class automatically selects binds to the best source available for the named object. The primary use for this data fusion algorithm is situations in which there are redundant object

representations. For example, one use in past Testbed research has been to augment real-world sensor data with simultaneously generated simulation data. This is useful to apply algorithms that require information about all facilities in the network to operate properly to networks that do not have complete sensorization. In this situation, the client algorithm specifies real-world data as the primary information source, and simulated data as a secondary information source. The data fusion capabilities in the helper class binds to real-world servant objects when they're available, and to the simulated servant objects otherwise. This approach can also support fault tolerance by, for example, specifying a historical database as a possible alternative to faulty sensor readings.

At present, no other analysis modules have been fully integrated as **TRICEPS** modules. Two incident detection algorithms, however, are candidates for inclusion, but the work has not yet been completed to date.

Estimation and prediction

Work is continuing on the development of *Paradyn* as a dynamic traffic assignment and OD-estimation tool. As this work matures, a new IDL will be added to the TAIL core that specifies the interface to the estimation results provided by *Paradyn*. The most likely implementation of this interface will use an intermediate central database that stores estimation results and serves them to other modules in the system by way of the estimation IDL.

Management

The TML contains the **CARTESIUS** transportation management tool that has been tested and evaluated for the Testbed network. **CARTESIUS** serves as the primary interface to candidate ATMIS solutions that use it. The encapsulation of **CARTESIUS** permits users of the program to implement control and management actions suggested by **CARTESIUS** to respond to the transportation system state it obtains from the system. The measurement and control interactions depend on the state measurement and control implementations discussed above. **CARTESIUS** is discussed in full detail in Chapter 3.

Control

At present, only statically optimized traffic responsive control is available in the TML. The control algorithm of the NMEA type 170 controller is actually implemented as a plug-in to the simulation tools described in Chapter 4. Candidate ATMIS solutions can manipulate the behavior of these control algorithms by way of the standard ICU interface defined in the TAIL IDL..

At present, no control modules meeting 2.5G or better functionality have been developed for use as **TRICEPS** modules, though candidate control algorithms have been developed in Testbed research. The primary reason for this is the lack of a satisfactory dynamic traffic assignment module in the TML, which provides the necessary inputs to the devised control algorithms.

2.5.3 TRICEPS Management Tool

The **TRICEPS Management Tool** is still a work in progress. Ultimately, the TMT will be a set of meta-applications for specifying and controlling ATMIS implemented using **TRICEPS**. The need for such a tool can be illustrated by way of example. Consider an ATMIS system that decomposes the managed network into a series of sub-networks that are each optimized by relatively independent subsystems (mini-ATMISs). These sub-networks may correspond to the different agencies responsible for portions of the transportation system, such as Caltrans and the various cities whose jurisdictions are overlaid by the freeway system (see Chapter 3 for a more complete discussion of the problems of institutional overlap in transportation management). To evaluate such a decomposed ATMIS application, researchers must define the mini-ATMISs, the portions of the network they are to manage, and the interactions they are to have with the ATMISs managed by neighboring jurisdictions. These mini-ATMISs consist of a bundle of software modules that will need to know what data they'll need to obtain and process as part of their role in the ATMIS. Specifying this information for each module, in each mini-ATMIS, quickly becomes a cumbersome task. The envisioned role of the TMT, therefore, is to streamline this process.

As envisioned, the TMT will rely upon a set of **TRICEPS** management interfaces defined as part of the TAIL. These interfaces will define a module interface layer that provides the hooks that the TMT will use to specify data interdependencies between modules. Thus, we can imagine the TMT as a user interface that allows researchers to graphically specify a set of **TRICEPS** modules to connect, and a set of data (e.g., portions of the network) upon which each of these modules should operate. In some scenarios, the TMT may simply define an ATMIS configuration with an unambiguous flow of data through it (as in Figure 2.1) and start it running. In others, the TMT may play an integral role in coordinating the execution of the algorithms in the ATMIS (as when the feedback loops shown in Figure 2.1 are used).

While a centralized meta-configuration application does not yet exist for **TRICEPS**, a library of scripts are available to automate ATMIS configuration (mostly written in Perl). As each module in a candidate ATMIS starts, it consults an input file to determine what data it will operate upon, and what data sources it should use to obtain that data. The TMT scripts allow rudimentary coordination of these various input files. As development progresses, this script/datafile-based system will be migrated to a GUI/database system that meets promise envisioned for the TMT and greatly improves **TRICEPS** use-ability.

2.6 Summary

TRICEPS is a fundamental component of the UCI Testbed that is designed to support the development, implementation, and testing of large-scale ATMIS solutions in both simulated and real-world settings. While it is still evolving,

TRICEPS has achieved its core goals, and is now capable of supporting transportation systems management research at UCI. The Transportation Algorithm Interface Library provides research flexibility and production-level stability in an easy-to-use framework. These requirements are met by encapsulating an industry standard CORBA implementation with a set of high-level interfaces that automate many of the common tasks associated with integrating interdependent software modules. The **TRICEPS** Module Library provides a core set of modules that are fundamental to any ATMIS configuration. The availability of these modules allows researchers working on algorithms that depend on data from such independent components to quickly place their research in to the simulation (or real-world) loop and evaluate their work. Finally, the **TRICEPS** Management Tool simplifies the configuration of candidate ATMIS using various combinations of the modules available in the TML.

In the following Chapters, we discuss the development of two key components of the TML. First, Chapter 3 discusses the **CARTESIUS** traffic management module that provides core ATMIS functionality to the TML. Then, Chapter 4 outlines the enhancements made to the Testbed simulation tools. Chapter 5 then describes the evaluation of **CARTESIUS** that was performed on the **TRICEPS** platform using the Testbed simulation tools.

Chapter 3

CARTESIUS

3.1 Overview

A central ATMIS capability is a timely and efficient response to non-recurring congestion. The complexity of traffic on urban networks requires substantial interaction between the various agencies that share responsibilities for its management. Coordinated response to congestion phenomena among these agencies avoids the implementation of operations that may be otherwise conflicting, and therefore counter-productive. At the same time, the spatial and administrative organization of such agencies often results in wide differences in policies and responsibilities. Thus a coordinated solution effort is required to satisfy all parties, preserve their own levels of authority, guarantee privileged control of their data, and in general reflect the inherent distribution of the decision-making power.

In response to the recognized need for integrated, area-wide transportation management infrastructures, several research and development projects are currently underway. Some of these attempts focus on the development and evaluation of area-wide control strategies, sometimes though, failing to recognize important institutional issues, such as the partition of responsibilities among several agencies. Some others try to address the complex problem of data exchange by implementing regional monitoring systems and data sharing links between several traffic operations centers. In most cases, though, regardless of the means of communication, only raw, unprocessed data, such as traffic parameters, or data describing the chosen control plan, are exchanged. Such a data-exchange protocol, in general, does not allow one unit to take advantage of, and complete, the partial data processing performed by other units. A more effective cooperation can be achieved by exchanging knowledge rather than data. In other words, processed information at intermediate steps during the decision-making process (such as the strategies selected to reach a certain goal, conditions to be satisfied by compatible control schemes, or benefits expected by certain actions) allows the cooperating units to direct their search for solutions toward a common direction. Even in existing "smart systems", the coordination of control strategies appears to be, in most cases, driven by pre-planned and previously agreed upon conditions.

3.2 CARTESIUS

CARTESIUS is a distributed architecture for real-time area-wide traffic incident response and management that provides cooperation among control modules, or agents, for the development of integrated, network-wide control in response to

incidents. The solution process provides interaction mechanisms that enable cooperative reasoning and conflict resolution by combining the desire of each independent system to preserve its autonomy and maintain the control of the facility under its jurisdiction with a willingness of all agents involved to cooperate and unify their problem-solving capabilities to achieve conflict-free, integrated responses. The architecture comprises modules that exchange high-level (i.e., highly processed) information for the identification of traffic congestion and the formulation of appropriate integrated response plans. This information may include partial and potentially incomplete results during the execution of problem-solving tasks and exploits inter-agent constraints to resolve inconsistencies that are due to the limitations of their information, in order to integrate local solutions into global, network-wide control plans.

As currently implemented, there are two agents which are decision-support systems for Traffic Management Center operators. One agent is responsible for the operation of a freeway sub-network and the other is responsible for the network of adjacent surface streets. The modularity of the architecture and the flexibility of the communication protocol provides for the accommodation of additional units, such as decision support for a transit operations agency, for local jurisdictions, or for police or emergency service.

3.3 CARTESIUS Architecture

The distributed architecture in **CARTESIUS** comprises interacting, real-time problem-solving agents that communicate with each other through a fast TCP/IP-based real-time protocol. The agents are able to perform cooperative reasoning and resolve potential conflicts for the analysis of non-recurring congestion and the formulation of system-wide ATMS/ATIS response strategies. As shown in Figure 1, the two agents are decision-support systems for a TMC operator. One agent supports incident management operations for a freeway sub-network and interacts with a human operator at the TMC of a freeway management agency. The other agent supports operations for the adjacent arterial network, and interacts with an operator at the local city TMC. Each module continuously receives real-time measurements from traffic detectors and a description of the current status of the control devices (signals, ramp meters, and CMS) under the jurisdiction of the corresponding agency. The modules provide the operators with traffic control and traveler information recommendations in response to the occurrence of incidents. These recommendations consist of a set of alternative, network-wide strategies, composed of suitable settings for signals, ramp meters and CMS. The agents provide an explanation of the reasons why each strategy is proposed and an estimation of the benefit it is expected to provide.

The uniqueness of the **CARTESIUS** approach lies in the efficient integration of existing techniques for real-time generation and assessment of appropriate control strategies, with emphasis on the coordination between multiple decision

makers in a multi-criteria environment. The analysis of the network state and the search for suitable control plans is based on a structured combination of heuristic approaches and well-established traffic control algorithms in a general distributed framework that provides the means for cooperation and conflict resolution.

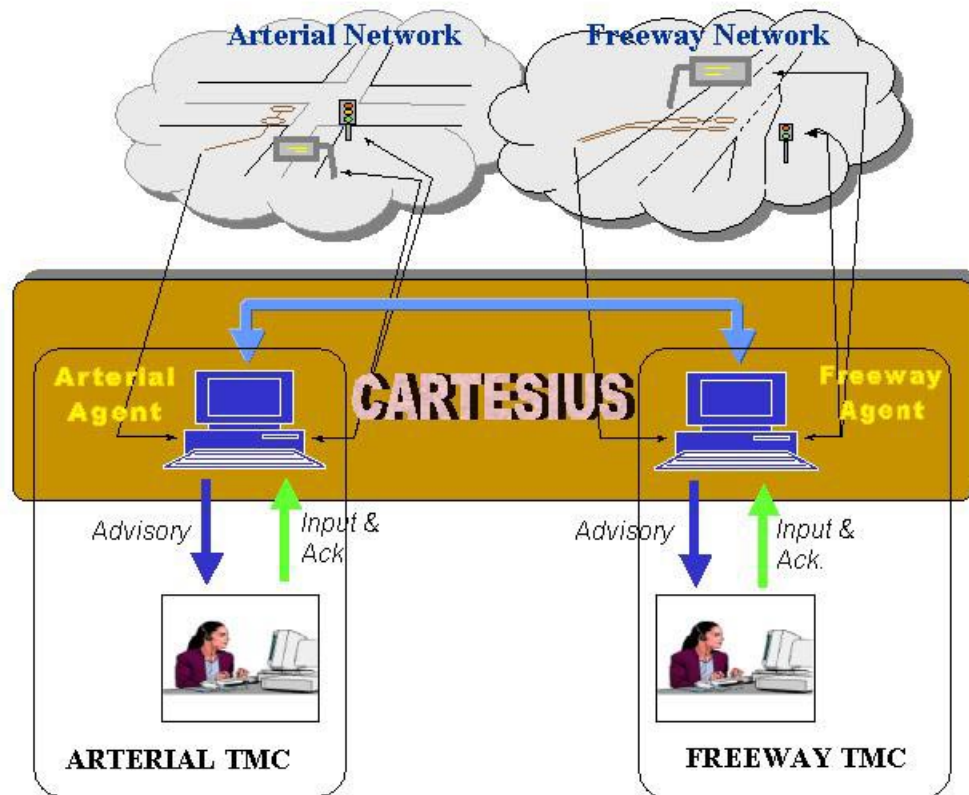


Figure 3.1 CARTESIUS Multi-agent Architecture

3.3.1 Agent Organization and Data Sharing

An organization based on interacting agents, as opposed to one using a central module with coordinating functions, was dictated by the following considerations. First, the need to have control decisions ratified by TMC operators and the lack, within the administrative organization of transportation management agencies, of an authority able to coordinate and potentially override control decisions made by either of the agents, limited the power of the coordinating module. Once the functions of the coordinating unit reduced to mere message passing and automatic (unmanned) decision-making, it was decided to eliminate the coordinating module, by allowing the agents to share some information and introducing a degree of computation redundancy. Thus the agents were provided with the ability to resolve inconsistencies through the definition and

verification of inter-agent constraints, and to decide when it is necessary or convenient to interact with another agent and what type of information should be exchanged.

The issue of data sharing is closely tied to the agent organization. A centralized database, accessible to both agents would require extensive data communications and originate potential access delays and maintenance complications. An organization in which copies of the same database were made available to both agents would call for complex mechanisms to guarantee consistency and at the same time would severely limit the system adaptability. These solutions have the advantage that, directly or indirectly, each unit has access to complete and exact data, thus making the problem solving process easier to deal with. At the same time, though, they would preclude the interacting agencies to have reserved data access, thus interfering with their desire of relative autonomy and exclusive control of their jurisdictions.

A more suitable option involves the adoption of a partitioned database, such that each agent has exclusive access to a portion of the data (the one local to its own jurisdiction) and provides the other modules with abstractions of the data that are considered relevant for the accomplishment of its tasks. Such an option allows each agent to preserve dedicated control over its portion of data, by controlling the amount and the quality of the information that is made available to the other agent. Another important advantage of this option involves the reduction of data processing that can be achieved, by having one agent process its data locally, and then making intermediate or final results of such processing available to the other agents. Given the lack of completely specified and globally accessible information, such an approach requires providing the agents with mechanisms for satisfying constraints and resolving inconsistencies, to develop a globally compatible and efficient solution

Thus, input data describing the status of the network is made available to each agent through access to detector data on road sections that are part of the sub-network controlled by the corresponding agency. A small redundancy was introduced for the agents to assess the status of the network at the boundaries between the freeway and the arterial network. The status of traffic controllers (signals and ramp meters) is partitioned in such a way that each agent has access to and can set only the controllers under the jurisdiction of the corresponding agency.

Data related to CMS are treated in a slightly different way: in order to guarantee consistent traveler information, predefined combinations of CMS messages are used, that include settings for CMS both on the freeway and on surface streets. Groups of messages that initiate traffic diversion from the freeway are part of the knowledge of the freeway agent, while those that initiate traffic diversion from surface streets are part of the knowledge of the arterial agent. This is consistent with real-life scenarios, for example in California, where often Caltrans, the

agency responsible for freeway operation is aware of the possible messages that the local City TMC can use, and vice versa. Nonetheless, each agency has exclusive authority over the CMS within the network under its jurisdiction.

3.3.2 The Interaction Mechanism

The structure of the distributed processing and the interaction between the agents is based on the Functionally Accurate, Cooperative (FA/C) paradigm that was introduced by Lesser and Corkill (1981). This paradigm has been applied to the development of distributed systems in several fields (Carver and Lesser, 1995; Lesser, 1991; Carver et al, 1991). According to the FA/C approach, agents cooperate by generating and exchanging partial results at various levels of abstraction, obtained during the problem-solving process. These results, which may be incomplete or inconsistent, are based on the agents' limited local view of the problem and of the solution domain. By providing the cooperating modules with the ability to determine a local solution even in the absence of complete and current information and then using processed data coming from other agents to determine a consistent global solution, the agents may reduce their communication bandwidth and synchronization time delays.

The FA/C approach is particularly suited to applications such as distributed sensor networks and distributed control, where there is a natural spatial distribution of information but where each agent has insufficient knowledge to completely and accurately solve the global problem. In the context of traffic management in metropolitan networks it is often impossible or too expensive to decompose the problem in such a way to ensure a perfect match between the location of information and data processing expertise, and the computational requirements for problem solving. On one hand, the impracticability of sharing expertise and decision-making power in a real-time context often limits the flexibility of transportation management systems, by requiring the adoption of predefined, previously established cooperation plans. On the other hand, the maintenance of accurate, complete, and up-to-date information requires too heavy and frequent communication of intermediate processing results, thus burdening the agents with high communication and synchronization delays that are not practical in real-time applications.

Therefore, the FA/C paradigm fits well problems that can be solved through a search process, requiring the examination of many alternative partial results in order to arrive at a complete solution. Each agent must be able to detect inconsistencies between its tentative partial results and those received from the other agent, and integrate into its local data base those portions of the results coming from the other agent that are consistent with its own.

The FA/C problem-solving approach allows agents to cooperatively solve tasks, using only limited and uncertain knowledge of the processing performed and the results obtained by other agents. According to the FA/C approach, **CARTESIUS**

agents cooperate by generating and exchanging partial results at various levels of abstraction, obtained during the problem-solving process. These results, which may be incomplete or inconsistent, are based on the agents' limited local view of the problem and of the solution domain. The ability to determine a local solution even in the absence of completely specified and up-to-date information and to use remotely processed data for the selection of a consistent global solution, allow the agents to reduce their communication synchronization delays.

3.4 TCM: Traffic Congestion Manager

The agents in CARTESIUS are based on TCM (Traffic Congestion Manager), a centralized decision support system for incident response and traffic control management, described in more detail in Logi (1995). In the distributed architecture, each agent is an enhanced instance of TCM, suitably modified to provide inter-agent communication, cooperative reasoning and conflict resolution capabilities.

TCM comprises a structured collection of knowledge modules which are used for the detection of critical traffic conditions, for the analysis of their causes, and for the selection of suitable plans for traffic control and traveler information. These plans are presented to the operator with an explanation of how they were generated and what benefits they are expected to provide. The operator is notified of the occurrence and the characteristics of critical situations and is presented with a sorted list of possible solutions which are expressed as sets of suitable signal plans, freeway ramp metering rates, and CMS messages.

Input data from an external incident detection algorithm, from traffic sensors (volume and occupancy) and, if needed, from the operator, are used to determine the characteristics of the detected problem (type, location, and anticipated capacity reduction) as well a path-based assessment of the traffic volume currently affected by the problem. , This allows for the estimation of the impact that the detected problem is expected to have on traffic circulation.

After the current problems have been analyzed, and their impact on traffic has been forecast, the system starts a process for the search of suitable solutions. Given the impracticality of formally defining an optimal solution, and the need for a real-time operational environment, the search for solutions is oriented towards selecting a set of *satisficing* alternatives. Each solution is composed of a combination of signal plans, CMS messages and ramp-metering rates, selected from a pre-stored database. TCM uses a heuristic technique based on an iterative step-wise procedure that performs an efficient search of the solution space, given by all possible combinations of control device settings, and selects those solutions that are expected to produce an improvement in the traffic conditions. Each combination of compatible control settings is ranked according to the extent of the expected improvement and a sorted list is proposed to the

operator. The process starts from a description of the problems that must be solved. At each step, a set of problems is analyzed, by first determining a set of suitable goals to be pursued, and then expanding each of those goals into a set of actions on the control devices, expected to achieve - partially or entirely - those goals. At the end of each step, an estimation of the expected effect of the partial control solution, allows the original problem to be modified into one of smaller scale, and the procedure continues on the problems that have not been reduced to an acceptable level.

3.5. The Distributed Algorithm

Two instances of individual TCM modules, enhanced and suitably modified to support inter-agent cooperation, are currently incorporated in CARTESIUS. The two agents communicate through a CORBA-based real-time protocol that allows them to exchange information in the form of complex objects, within an object-oriented paradigm. The main goal of the distributed algorithm is to determine a satisficing and consistent global solution, requiring the least amount of synchronization and communication possible, so as to reduce delay.

The following terminology is used:

- a *problem* is an object describing the characteristics of an operational problem;
- a *problem description* is a list of problems;
- a *network load* is a time-dependent, path-based estimation of the traffic distribution across the network;
- a *problem state* is a couple given by a problem description and a network load;
-
- a *control action* is a list of control device settings, such as plans for signals (minimum and maximum green, unit extensions, and phase sequence), messages for CMS, or metering rates for freeway ramp meters;
- a *strategy* corresponds to an intermediate goal in the problem solving decomposition, and describes a high-level (device-independent) way to achieve such a goal. Strategies include reducing the flow upstream of the congested location, via traffic diversion or metering, increasing the capacity downstream of the congested location, through signal and ramp metering control, or adjusting the capacity along alternative routes in case of traffic diversion. Each strategy has an index for each type of problem, that provides a way to select the next strategy to be expanded, during the search process;

- a *condition* is a requirement that has to be verified by a certain control setting. Normally it is triggered by the selection of a control action, and in such a case, a description of the effects of that control action is part of the information associated with the condition;
- a *candidate solution* describes a step during the search process. It is composed of the strategies and control actions selected so far, the problem state modified by the effect of the control actions (and the conditions that they satisfy), and a heuristic value that provides a measure of its value.

The algorithm executed by each agent comprises three main phases.

3.5.1 Initialization

In this phase, the agent exchanges information with the other agent, describing the problems that have been detected and analyzed by both agents, for which a solution has to be determined. This step allows the agent to become aware of problems detected by the other agent, that may in general affect its search for a solution, and that the agent can help solve.

3.5.2 Search for Local Solutions

This step involves determining a set of solutions composed of control actions local to the agent. As in the centralized algorithm, a search process is performed, that can be visualized as the traversal of a search tree. The root node corresponds to the initial problem state and an empty set of control actions. At each step, an open node, selected based on the heuristic value of the corresponding solution, is expanded into a set of children nodes, one for each alternative feasible strategy. One of these strategy nodes is expanded into a set of children nodes, one for each of the alternative control actions that implement the corresponding strategy. The selection of the strategy node to be expanded is based on the strategy's index. A strategy can not be selected if one with a lower index has already been used within the same search. This prevents multiple searches composed of different sequences of the same set of strategies to be selected, thus avoiding redundant searches. The search terminates when none of the problems can be reduced any further, either because the control applied to them is expected to reduce them to problems of sufficiently low magnitude, or because all meaningful combinations of control actions have been considered.

To guarantee compatibility among a set of solutions, the selection of a control action may require other solutions from both agents to satisfy a certain condition. The condition requirements are then transmitted to the other agent, and both agents transform them into a strategy, which is translated into control actions local to the agent.

At the end of the search the agent has a list of local solutions. Each solution stores information describing the list of conditions that it satisfies, partitioned into the set of those whose satisfaction was requested {em to} the other agent, at some point in the search, and those that were requested {em by} the other agent. This information is used by the third step of the procedure, for the integration of the partial solutions.

3.5.3 Search for Global Solutions

This step involves the construction of consistent and efficient global solutions. These are obtained by:

1. selecting pairs of solutions, one from each agent, that are compatible and mutually consistent;
2. sorting them according to their expected delay reduction;
3. selecting a limited number of the most promising ones;
4. running short, faster-than-real-time simulations of the near-future network performance to verify which of the solutions yields the best improvement, including among the simulated scenarios, the one with the current (no-change) control.

Two solutions, one (SOL_1) determined by the freeway agent and one (SOL_2) determined by the arterial agent, are mutually consistent if:

1. each is internally consistent (this is verified at the local level during the second phase of the algorithm);
2. their control actions are compatible, and;
3. they satisfy the same set of conditions.

The following condition-compatibility rule is applied. Letting $TO(SOL_1)$ and $BY(SOL_1)$ be the lists of the conditions satisfied by SOL_1 , requested respectively to and by the freeway agent, and $TO(SOL_2)$ and $BY(SOL_2)$ be the lists of the conditions satisfied by SOL_2 , requested respectively to and by the arterial agent, then SOL_1 and SOL_2 are compatible if:

$$TO(SOL_1) = BY(SOL_2) \text{ and } TO(SOL_2) = BY(SOL_1).$$

If two solutions are found to be mutually consistent, they are assembled into a unique global solution, by combining their control actions. The corresponding expected delay reduction is computed considering the effect that the global control action is expected to produce. The creation of compatible solutions, their sorting, and the selection of a small number of the most promising ones provides

a significant reduction in the number of feasible alternative control plans, and therefore a reduction in the number of simulation processes to perform. The ultimate sorting is based on the results of the simulation, and the resulting list of plans is then presented to the operator.

3.6 CARTESIUS Summary

The initial formulation of **CARTESIUS**, which was based on TCM, was completed prior to the development of **TRICEPS**. Nevertheless, significant enhancements have been made to both **CARTESIUS** and to its interface with **TRICEPS**. Details of the development of **CARTESIUS** are provided by Logi (1999).

Chapter 4

TRICEPS Simulation Components

4.1 Overview of Simulation Model Components

A variety of tools have been used over the years to simulate transportation system performance. From the earliest days, the trade off between the efficiency and abstraction of macroscopic models, and the detailed precision of microscopic models has been a factor. Much of the earliest work on the Testbed utilized the FHWA's microscopic simulation tools: the INTRAS freeway simulation (later Freesim) and the NETSIM arterial network simulator to implement and evaluate ATMIS algorithms. Use of these tools required the researcher to modify the simulation's (very dated) FORTRAN source code to embed the algorithm of interest into the running program directly. Eventually, the overhead associated with this work led to a line of development that produced an interactive version of INTRAS to which control directives could be sent using remote procedure calls (RPCs). INTRAS's restriction to modeling a single freeway segment, however, led to further development. In particular, was the desire to model wide scale network routing behavior, and the associated response to traveler information. The mesoscopic DYNASMART (DYnamic Network Assignment Simulation Model for Advanced Road Telematics) simulator had also been used for a variety of research projects on the Testbed, and provided validated models of routing behavior, but lacked the microscopic detail to model all forms of traffic control effectively. The merging of DYNASMART, INTRAS, and NETSIM to create a hybrid traffic simulator seemed an ideal solution. Research continued and a Hybrid DYNASMART/INTRAS simulator was the result. This simulation tool was the first Testbed simulation workbench, and provided scalable simulation capability, with an interactive simulation shell that allowed for modular expansion of the simulation, including the interfacing of remote modules (such as traffic control algorithms) by way of the same RPC mechanism that was developed for the INTRAS simulator.

Despite these improvements, the dated INTRAS code proved cumbersome to maintain and modify, and the INTRAS component (and planned NETSIM component) were abandoned in favor of a new configuration that addressed the needs of ATMIS simulation more directly. This new configuration couples the **Paramics** (PARAllel MICroscopic Simulation) microscopic simulation model developed in Scotland (Duncan, 1995; Quadstone Ltd., 1999) with the mesoscopic DYNASMART (DYnamic Network Assignment Simulation Model for Advanced Road Telematics, Jayakrishnan *et al.*, 1994) model. The result is a hybrid simulator capable of modeling operation and effects of nearly every ATMIS technology available or imagined.

4.1.1 Paramics

Microscopic simulation using **Paramics** provides many flexible, advanced and useful features, perhaps more than most other existing microscopic simulations, certain limitations exist in the modeling and evaluation of ATMS. Because **Paramics** represents traffic flow from the standpoint of the individual driver, traffic engineers are able to distinguish between minor sub-optimal design variations without resorting to deterministic proxies. All known components likely to significantly effect traffic flow are represented, across the full range of road network types. **Paramics** can currently simulate the traffic impact of signals, ramp meters, loop detectors linked to variable speed signs, VMS signing strategies, in-vehicle network state display devices, and in-vehicle messages advising of network problems and re-routing suggestions.

The primary difficulty with microscopic simulations is the inability to handle path dynamics in large networks. **Paramics** allows vehicle routing according to routing tables and feedback capturing information supply, but does not allow storage of sufficient path trees and storage of individual vehicle's routes, which are essential requirements for the simulation of driver response towards information supply and the resulting route choice. The difficulty arises from the detailed network descriptions used in such microscopic simulation models. The node and link representations for microscopic simulations are often such that any point on a physical link with a change in geometry or other characteristics results in an extra node in the representation. This results in an order of magnitude more nodes and links in networks used in such simulation, than needed to model the path dynamics which requires only the network made up of the true *decision nodes*, which are the nodes that are of significance in the drivers' route decisions. **Paramics'** scalability permits vehicle simulation of very large networks with additional processors, but if detailed driver response modeling and path processing are to be incorporated, such microscopic models can only be used to simulate small to medium-sized urban areas. This is because many network algorithms show nonlinear increase in storage and computational requirements as network sizes increase.

One of the nice features in **Paramics** is that many features of the underlying simulation model can be customized. Access is available through a functional interface or application programming interface (API). This API allows additional functionality by adding more external modeling routines. The hybridization of **Paramics** and DYNASMART, as well as the encapsulation of the hybrid simulation as a TRICEPS module are both implemented via the **Paramics** API.

4.1.2 DYNASMART

Jayakrishnan pointed out the two primary deficiencies of existing simulation models when introducing DYNASMART (Jayakrishnan *et al.*, 1994): (a) the lack of modeling of path-based traffic dynamics and (b) the lack of explicit

representation of driver decisions such as route-choice under information. Even though microscopic simulation models are more often used due to their capability of representing realistic vehicle movements, it is even harder to remedy the two deficiencies that Jayakrishnan pointed out. The problem arises from the sophisticated network representation required for describing vehicle maneuvers in microscopic simulation and in the memory requirements for storing individual vehicles' paths. Certainly, it is a large burden to store individual vehicles' paths; however, importance of realistic route choice behavior modeling should not be overlooked in microscopic evaluation simulation models for ATMIS.

DYNASMART was developed to address these issues, specifically as they related to studying the effectiveness of alternative information-supplying strategies, as well as alternative information/control system configurations. The model represents each vehicle individually (as in a microsimulation model) but simulates individual vehicle movements according to a macroscopic flow model. In this manner, the drivers' path selection behavior in response information can be explicitly modeled. The path-processing component is designed for efficient application of the framework to large and realistic networks. While DYNASMART does model individual vehicles, their movement are on idealized network links, and the number of nodes in the network model may not be significantly higher than the decision nodes in the actual network. Its ability to model certain microscopic details of traffic movement may be limited, but it has the ability to model network level traffic details such as path travel times effectively.

In DYNASMART, path dynamics are modeled based on the route or routes that drivers have in their minds, and the routes provided by ATIS. A simple behavior mechanism used often is a comparison between the current route and the best alternate. Thus, the routes in the minds of individual drivers are stored a separate lists for the comparison. The flexibility for modeling various driver response mechanisms and information supply strategies comes from the ability to find and store multiple paths efficiently, using networks of reasonable sizes. DYNASMART was used as a simulation tool to find dynamic assignment solutions (Mahmassani *et al.*, 1994) and was extended to multi-user class real-time assignment (Peeta and Mahmassani, 1995).

4.2 Development of a Hybrid Simulation Framework

4.2.1 Overall Model Structure

The ***Paramics***' capabilities are enhanced with additional routines added though the application programming interface (API). Since various control strategies are tested and evaluated in the Orange County ATMIS Testbed, flexibility and data interface are essential part of the simulation model. In addition to the main ***Paramics*** module, the hybrid simulation framework consists of five additional modules: (1) Monitoring, (2) Adaptive Traffic Control, (3) Data Communication,

(4) Route Information, and (5) Route Decision. Each module consists of one or more APIs. The individual APIs have their own functions interfacing with other modules within the simulation framework. The flexible nature of the model framework allows easy incorporation of new technologies and algorithms into the model framework. Figure 4.1 shows the overall interrelation between modules.

4.2.2 TRICEPS APIs

Paramics provides a framework that allows the user to customize many features of underlying simulation model. The capability to access and modify the underlying simulation model through Application Programming Interfaces (APIs) is essential for research. These APIs have a dual role: first, allowing users to override the simulator's default models (such as car following, lane changing, or route choice), and second, allowing users to interface complementary modules to the simulator. Complementary modules could be any of a variety of ITS applications, such as signal optimization, adaptive ramp metering, or incident management. Paramics could evolve as a simulation shell with actual simulation components loaded as customized API plug-and-play tools.

In the **TRICEPS** project, some APIs related to traffic control have been developed and implemented to allow Paramics to better represent traffic infrastructure and flow scenarios typical of California and thus needed in ATMIS research. These APIs include:

1. Fully-actuated signal control

The standard eight-phase, dual-ring, concurrent controller logic has been implemented in this API. Each of eight phases accommodates one of the through or left turning movements. The right turns are omitted and assumed to proceed with the through movements. In full-actuated signal control, all phases at an intersection are actuated, then the length of each phase, and consequently the cycle length, will vary with each cycle. Some phases may be skipped if there is no vehicle actuation. To simulate the real controller better, the order and sequence of phases can also be altered.

2. Actuated signal coordination

Coordination is a mode of signal operation designed to allow platoons of traffic to form and "progress" through several signals with minimum stops and delay. Where signals are closely spaced and traffic volumes are high, coordination of signals is necessary to avoid excessive delay and stops. The actuated signal coordination API inherits most parts of full-actuated signal API, with additional force-off logic to maintain the background cycle length, and form green band for a particular phase (sync phase).

3. Time-based actuated ramp control

This API implements a fixed rate, time-of-day basis ramp control. The ramp-control signal, mounted close to driver level, generally provides two indications, green and red only. Vehicles are released from ramp to the mainline traffic at a fixed ramp-metering rate during the certain time period. Every vehicle has to stop before the stop lane, waiting for the green signal. The detector for sensing the presence of a vehicle allows the signal to rest in red, avoiding potential confusion to a driver approaching the signal, due to the short greens.

Although significant effort has already been expended in enhancing the capabilities of the model through the development of APIs, continued effort is required in this regard, particularly as it relates to the model's ability to interface with dynamic routing choices, and dynamic O-D estimation.

4.3 Encapsulation of the Hybrid Simulator as a TRICEPS Module

Recall from Chapter 2 that for a particular software component to be used as a module in the **TRICEPS** framework, it must be encapsulated by a set of CORBA servants that implement the interfaces of transportation system objects, or ATMIS modules, as defined in the TAIL library. The details of implementing a **TRICEPS** module have already been discussed in Chapter 2. We just briefly here mention the steps taken encapsulate the simulation for use in **TRICEPS**.

The Testbed simulation component, as a proxy for the real world, must implement servant objects for the transportation system infrastructure it models. This infrastructure includes the range of ATMIS measurement and control devices including: loop detectors, ramp meters, signal controllers, variable message signs and probe vehicles. The implementation of the object servants for each of these devices is accomplished using the Paramics APIs. Upon initialization, a particular servant object, say for a loop detector, is instantiated with the index or pointer to the simulated device. When an external client module, such as the **CARTESIUS** model described in Chapter 3, obtains a reference to this servant and invokes a method on the reference (say, to obtain the current volume measurement at the detector), the CORBA infrastructure invokes the associated method implemented by the servant object. The implementations for devices represented by the simulator has been written to call the appropriate Paramics API, using the pointer that the servant object is associated with as a parameter to the API. The result of the API is returned (via CORBA) to the calling module. The **TRICEPS** architecture automatically makes the devices implemented by the simulator available as soon as servant objects are instantiated. Thus, as soon as the simulation starts and the instantiation is complete, all of the devices in the simulated network are available to any module in **TRICEPS**.

Chapter 5

TRICEPS / CARTESIUS Evaluation

5.1 Introduction

This chapter describes the initial testing and evaluation the key components of the Testbed Real-time Integrated Control and Evaluation Prototype System (**TRICEPS**). **TRICEPS** is a software platform that facilitates the implementation and evaluation of a wide range of algorithms for traffic control and Advanced Transportation Management Systems (ATMS). **TRICEPS** supports research activities by providing consistent interfaces for transportation management modules to both simulated and real-world environments. One of the key components of the **TRICEPS** platform is **CARTESIUS**, a distributed architecture for the provision of real-time decision support to Transportation Management Center (TMC) operators that provides a set of core transportation management applications for multi-jurisdictional traffic control and incident management on freeway and arterial networks. The **TRICEPS** architecture hosts algorithms for the estimation of current traffic conditions, the analysis of incident characteristics and the formulation of multi-decision-maker traffic control plans, using advanced methodologies for cooperation and conflict-resolution. The process of evaluation of such methodologies using the **TRICEPS** platform, while aimed at demonstrating the effectiveness of the cooperative approach, also provides a demonstration of platform functionality for range of related applications.

5.2 Evaluation of System Performance

The evaluation of the ATMS/ATIS strategies proposed by **CARTESIUS** involved the analysis of network performance under different traffic conditions, determined by the occurrence of several types of incidents, and a comparison between network-wide measures of effectiveness (MOE), with and without the implementation of those strategies. Three real-time ATMS/ATIS response strategies were concurrently applied:

1. Adaptive system-wide ramp metering,
2. Adaptive arterial traffic signal control, and
3. Traffic diversion based on traveler response to CMS information.

The MOEs considered included:

1. An assessment of the network travel time reduction obtained by the implementation of incident response plans suggested by optimal deployment of the three ATMS/ATIS response strategies.
2. The system's response time.

3. The impact of the integration between the various control components, by comparing the effect of fully integrated control plans (freeway traffic diversion and arterial signal control) to incomplete control plans that use exclusively traffic diversion schemes.

After a description of the site for which the analysis was developed, the quantitative and qualitative results of the evaluation process are presented.

5.3 The Test Site

The test site selected for the evaluation is a highly congested corridor in the city of Irvine, in Orange County, California. This network (shown in Figure 5.1), which is a sub-network of the full ATMS Testbed network, is a slightly larger version of the network used in the federally-funded Irvine Adaptive Control Field Operation Test. The network includes 4-mile intersecting sections of the Interstate 5 and 405 Freeways, the connecting SR-133 Freeway, and the adjacent sub-network of surface streets. Land use in the area is dominated by the Spectrum, a rapidly growing hi-tech employment and entertainment/retail center. Immediately north and south of the corridor are developed and developing residential areas of Irvine and Lake Forest. The network selected allows for testing the feasibility of diversion strategies that included both freeway-to-arterial and freeway-to-freeway diversion.

The City of Irvine Traffic Management Center (ITRAC) is responsible for traffic operations on the arterial network, with a computer-aided traffic system that controls over 240 signalized intersections, 32 of which are within the test network, and 5 arterial CMS. Signal control is fully actuated, and signal control parameters (minimum and maximum green, phase recall, etc.) are set according to a time-of-day basis. ITRAC also has control over 80 CCTV cameras located at major intersections and connected to the TMC through a fiber-optic network. Caltrans District 12's ATMS uses state-of-the-art computer, software, and communication systems to manage the flow of traffic on the county freeway network. Vital elements at the core of the system's operations include 30 CCTV cameras, 34 CMS, Highway Advisory Radio, 278 metered on-ramps, 1,098 incident call boxes and 258 directional miles of loop detectors. Within the sub-network for which this evaluation was conducted, Caltrans controls 3 CMS, located at the three major freeway entry points to the area. Ramp metering control is performed on all 18 freeway on-ramps within the sub-network.

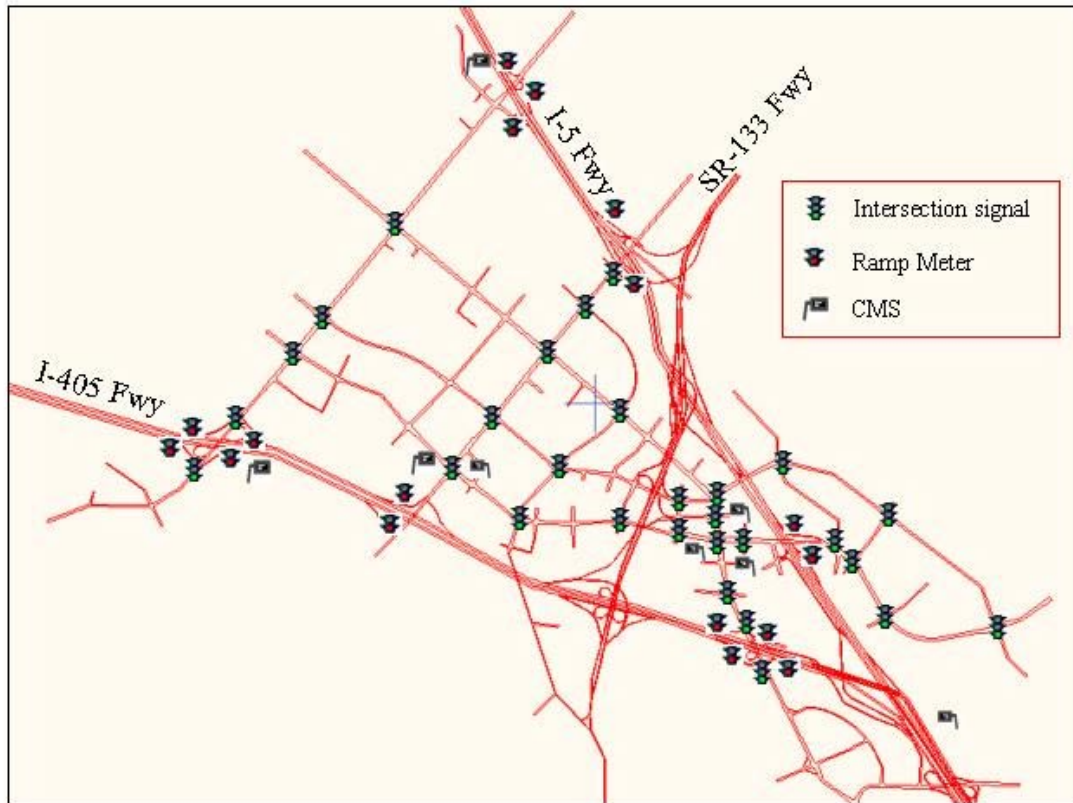


FIGURE 5.1 The Irvine Test Site

5.4 Evaluation Approach

The multi-agent, real-time ATMS/ATIS traffic management expert system, **CARTESIUS**, was used in this effort to identify “optimal” ATMS/ATIS responses to degradations in system performance. The two agents within **CARTESIUS** (one representing the City of Irvine and the other representing Caltrans District 12) are able to receive real-time traffic data from the Caltrans District 12 ATMS, through the ATMS Testbed communication network, a wide-area communications network backbone linking the City of Irvine’s traffic management center to the Caltrans’s District 12 TMC and to the ATMS Research Laboratories at the Institute of Transportation Studies, University of California, Irvine. In particular, one client was developed for interaction with the freeway agent in **CARTESIUS**, to query the Caltrans ATMS data server for real time traffic data, and transfer such data to the agent, through an object-oriented, TCP/IP-based real-time connection. The communication with Caltrans District 12 ATMS provides the freeway agent with loop detector data (volume and occupancy), with CMS and ramp metering data, and with the ability to transmit, subject to Caltrans’s approval, ramp metering rate control. A real-time connection between the arterial

agent and ITRAC is currently under development and could not be used in this research. It will allow the arterial agent to receive traffic and control status data from the arterial system, and to transmit, subject to ITRAC approval, alternative signal timing and CMS setting plans.

The agents in **CARTESIUS** were developed using G2 Version 4.1 (Gensym, 1995), a real-time knowledge-based system shell designed for the development and execution of complex applications that require intelligent monitoring and control in real-time. G2 is an object-oriented tool that allows one to define knowledge bases using several knowledge representation and inference techniques, including rule bases, procedures, and frame matching. It also provide means to develop remote real-time communication between several knowledge bases.

5.5 The Simulated Environment

The Testbed Simulation Workbench provided input data for the evaluation, using enhanced versions of the traffic simulators DYNASMART (Jayakrishnan *et al.*, 1994) and PARAMICS (Quadstone, 1999), both components of the **TRICEPS** platform. The **TRICEPS** architecture provides interfaces that simulate the functions of the Caltrans and City of Irvine traffic data servers. Traffic and control device data (detector, CMS, signal, and ramp metering data) are exchanged between the agents and the server using exactly the same interface on the client side.

A time-varying OD matrix was estimated for input to the simulation based on data from the Irvine Traffic Analysis Model (JHK & Associates, 1993) using the TRANPLAN (UAG, 1995) and CONTRAM (Leonard *et al.*, 1989) software packages. ITAM is a model package that implements the traditional four-step process of trip generation, distribution, mode choice and assignment. ITAM comprises a regional model, which encompasses the whole Southern California network (Los Angeles, Orange, Riverside, San Bernardino, and Ventura counties), and of a local model, which is a subsection of the regional component that covers the City of Irvine and adjacent areas (consisting of approximately 600 zones). The network under analysis within the TRICEPS project is a subset of the local component; only data related to the local component were used for the generation of the OD matrix.

The trip generation process used 1991 person-trip rates for identified land uses and a data file corresponding to 1995 land use distribution, to produce updated person trip ends for the various zones. The trip distribution process was based on a calibrated gravity model, that used travel time as impedance, and resulted in person trip tables for the morning (AM), evening (PM), and off-peak hours. Based on the ITAM model, trips were then distributed among various modes (drive-alone, HOV, and transit, with a vast majority assigned to the drive-alone mode). Trips were then incrementally assigned to the network. The generated

volumes, both on freeways and surface streets, were then checked against ITAM screenlines and Caltrans volume counts from 1991, and were found to be within tolerable limits (for details, see Kulkarni and McNally, 1997).

The sub-area for the network under analysis, containing 320 nodes (including 106 zones, 85 internal and 21 external), and 679 links (including centroid connectors) was then extracted from the ITAM local network, and a new traffic assignment process was performed to load the sub-area network. Volumes on arterials and freeways were found to closely match those obtained from the assignment on the local ITAM network and thus were considered appropriate for further use.

The estimation of a dynamic OD matrix for a 3-hour PM-peak, specifying a path-based demand distribution aggregated in 15-minute intervals, was performed using a standard iterative process that involves, at each step, distributing the OD demand obtained from the previous step across the network, using observed vehicle counts previously collected for a subset of the network links, along major arterial and freeway sections (75 one-way links, in this case). This process uses the transportation software packages CONTRAM and COMEST.

CONTRAM assigns the OD demand to paths in the network in packets of vehicles. COMEST uses those data and observed volumes (from detector counts) to provide updated 15-minute assignments that attempt to match the observed data. The updated assignment is then input again into CONTRAM and the process is repeated until convergence (small changes in the output produced by COMEST in consecutive iterations) is reached. The average difference between observed link counts and those obtained by the assignment was 16 percent and was considered acceptable.

The time-varying path distribution resulting from this process was provided as input to the two agents, for the creation of the path-based, backbone network demand. The default PM-peak settings for signal control were obtained from ITRAC, and encoded in the arterial agent. When knowledge was elicited from ITRAC, signal control was based on time-of-day, traffic-actuated control, according to which signal control parameters (minimum and maximum green, unit extension, etc.) vary according to the time of the day. Analogously, default PM-peak ramp metering rates were elicited from the Caltrans District 12 TMC, and embedded in the knowledge of the freeway agent.

5.6 Evaluation Results

At the core of the evaluation process was the assessment of the system's ability to provide traffic control plans in real-time response to the occurrence of incidents. Total and average travel time and traveled distance, were considered

suitable MOEs, both because they provide an indication of the network level of service and because they are easily measurable using a simulator.

5.6.1 Network Performance

A set of 18 test scenarios was created, by running simulations of 90-minute peak periods and artificially injecting incidents (temporary reductions in the capacity of a link), by varying such characteristics as the incident location, the associated loss of capacity and the duration of the capacity reduction. For each scenario, the MOEs provided by the simulator (average and total travel time and traveled distance), were collected. For each test case, two simulations were executed: one, the *before* case, using the default control (no CMS message and the default, time-of-day signal and ramp meter timing plan), and one, the *after* case, implementing the integrated ATMS/ATIS control suggested by the agents, in response to the notification of the occurrence of congestion. The comparison of the network performance, through the implementation of the two forms of control, provided a measure of the performance increase that can be expected when the default control is substituted with the control plans based on ATMS/ATIS responses suggested by **CARTESIUS**.

A list of the major characteristics of the test scenarios is described in Table 5.1. In scenario 0 no incident was injected, in order to obtain estimations of the *basic* network average and total travel time and distance. Such estimations are used to assess the performance deterioration caused by the occurrence of incidents. Given the disparity between traffic volumes on the freeway and the arterial subsystems, **CARTESIUS** is primarily concerned with management of incidents occurring on freeways, thus test scenarios involved mainly incidents occurring on both directions of the two major freeways, Interstates 5 and 405. Scenarios 1, 2, and 3 correspond to the occurrence of an incident on the northbound 405 freeway, with different durations and capacity reductions, thus having different impacts on traffic. Scenarios 4, 5, and 6 correspond to the occurrence of an incident on the southbound 405 freeway. Scenarios 7, 8, and 9 describe an incident occurring on the northbound 5 freeway, while scenarios 10, 11, and 12 are related to an incident on the southbound 5 freeway. Scenarios 13 and 14 describe an incident occurring on an arterial (Alton Parkway), and Scenario 15 on the northbound 405 on-ramp at Irvine Center Drive. Scenarios 16 and 17 describe cases for which two incidents were injected simultaneously. In both scenarios, the first incident was simulated on the freeway (405 north and southbound), while the second was injected along one of the major paths that had been chosen by the system as a bypass for the first incident. The purpose of these latter tests was to analyze the system's response to multiple incidents with interacting effect on each other.

Tables 5.2 and 5.3 show a summary of the network performance for the 18 test scenarios. For each scenario, average (per vehicle) and total travel time and traveled distance are reported. Each scenario includes data pertaining to the use

of default control (Before), and to the use of integrated ATMS/ATIS control directives suggested by **CARTESIUS** (After). Table 5.2 reports the measured total and average travel time in each scenario, for the before and after case. Table 5.3 reports the total and average traveled distance, for the before and after cases.

Table 5.1: Incident characteristics in the test scenarios

Scenario #	Location	Lanes Blocked (Total)	Duration (minutes)
0	NO INCIDENT		
1		1(4)	20
2	405 NB	2(4)	20
3		2(4)	25
4		1(4)	20
5	405 SB	1(3)	20
6		2(4)	20
7		2(5)	15
8	5 NB	3(5)	15
9		3(5)	20
10		1(5)	15
11	5 SB	2(5)	20
12		3(5)	30
13	Alton EB	2(3)	20
14	Alton WB	2(3)	20
15	NB 405 on-ramp	1(2)	15
16	405 NB Alton WB	2(4) 1(2)	20 15
17	405 SB on-ramp	2(4) 1(2)	20 10

Table 5.2: Simulated average and total travel time

Scenario #	Incident Characteristics			Average Travel Time (min/veh)		Total Travel Time (*10 ⁴ veh-hrs)	
	Location	Lanes	Duration (minutes)	Before	After	Before	After
0	No incident			5.24	5.24	0.380	0.380
1	405 NB	1(4)	20	5.87	5.87	0.429	0.429
2		2(4)	20	6.76	6.32	0.496	0.463
3		2(4)	25	7.20	6.70	0.528	0.486
4	405 SB	1(4)	20	7.47	6.54	0.546	0.479
5		1(3)	20	6.86	5.85	0.502	0.428
6		2(4)	20	7.98	7.21	0.584	0.528
7	5 NB	2(5)	15	5.30	5.30	0.387	0.387
8		3(5)	15	5.55	5.26	0.400	0.383
9		3(5)	20	5.81	5.33	0.426	0.391
10	5 SB	1(5)	15	5.24	5.24	0.385	0.385
11		2(5)	20	7.05	5.97	0.510	0.438
12		3(5)	30	8.69	8.38	0.634	0.615
13	Alton EB	2(3)	20	5.26	5.26	0.381	0.381
14	AltonWB	2(3)	20	5.26	5.26	0.382	0.382
15	on-ramp	1(2)	15	5.26	5.24	0.383	0.381
16	405 NB	2(4)	20	6.78	6.35	0.496	0.459
	Alton WB	1(2)	15				
17	405 SB	2(4)	20	8.09	7.57	0.601	0.560
	on-ramp	1(2)	10				

Table 5.3: Simulated average and total traveled distances

Scenario #	Incident Characteristics			Average Travel Dist. (veh-miles)		Total Travel Dist. (*10 ⁶ veh-miles)	
	Location	Lanes	Duration (minutes)	Before	After	Before	After
0	No incident			4.48	4.48	0.197	0.197
1	405 NB	1(4)	20	4.51	4.51	0.198	0.198
2		2(4)	20	4.52	4.55	0.199	0.200
3		2(4)	25	4.53	4.54	0.199	0.199
4	405 SB	1(4)	20	4.51	4.53	0.198	0.199
5		1(3)	20	4.49	4.52	0.198	0.199
6		2(4)	20	4.50	4.52	0.198	0.198
7	5 NB	2(5)	15	4.48	4.48	0.197	0.197
8		3(5)	15	4.48	4.49	0.197	0.197
9		3(5)	20	4.48	4.48	0.197	0.197
10	5 SB	1(5)	15	4.49	4.49	0.198	0.198
11		2(5)	20	4.50	4.50	0.195	0.198
12		3(5)	30	4.50	4.52	0.196	0.199
13	Alton EB	2(3)	20	4.48	4.48	0.197	0.197
14	Alton WB	2(3)	20	4.48	4.48	0.198	0.198
15	on-ramp	1(2)	15	4.48	4.48	0.198	0.198
16	405 NB	2(4)	20	4.52	4.52	0.199	0.199
	Alton WB	1(2)	15				
17	405 SB	2(4)	20	4.51	4.53	0.201	0.201
	on-ramp	1(2)	10				

Data reported in Tables 5.2 and 5.3 show that the implementation of the control plans suggested by **CARTESIUS** determines, in general, a reduction in the average (and total) network-wide travel time. The average (and total) traveled distance is not significantly affected by the alternative control, even though the control plans, in all test scenarios, included the use of CMS messages. This result should be perhaps attributed to the limited size of the network.

In some scenarios (scenarios 1,7, 10, and 14), the injected capacity reduction was not high enough to cause congestion, thus the agents did not receive any alarm, and no alternative plan was proposed. Data for these cases are reported for completeness, but they are not used for the computation of average measures of performance.

5.6.2 Travel Time Reduction

The quality of the improvement, in general, varies according to the availability of alternative paths, the amount of spare capacity on those paths, and the demand/capacity ratio on paths affected by congestion. Tables 5.4 through 5.9 present summaries of the percentage travel time reduction for the various test scenarios partitioned according to the location of the incident (e.g., NB 405). In order to obtain a normalized measure of the travel time reduction across the scenarios, the tables also report, for each case, the average travel time increase with respect to scenario 0, which corresponds to a *no incident* situation, and can thus be considered as the base case. For each case, the difference in per-vehicle travel time corresponding to the two forms of system response (*non ATMS/ATIS* versus *ATMS/ATIS*), is reported. Also, the mean percentage travel time reduction, among all the scenarios for which **CARTESIUS** was notified of the occurrence of congestion is reported.

Table 5.4: Average travel time for the 405 NB scenarios

#	Incident Characteristics		Average Travel Time			Comparison with Scenario 0		
	Lanes	Durat. (min)	Before	After	Change (%)	Before (%)	After (%)	Change (%)
0	no incident		5.24	5.24	0.00	0.00	0.00	0.00
1	1(4)	20	5.87	**	**	12.02	**	**
2	2(4)	20	6.76	6.32	-6.51	29.00	20.61	-8.39
3	2(4)	25	7.20	6.70	-6.94	37.40	27.46	-9.94
Mean			-6.73%			-9.17%		

Table 5.5: Average travel time for the 405 SB scenarios

#	Incident Characteristics		Average Travel Time			Comparison with Scenario 0		
	Lanes	Dur. (min)	Before	After	Change (%)	Before (%)	After (%)	Change (%)
0	no incident		5.24	5.24	0.00	0.00	0.00	0.00
4	1(4)	20	7.47	6.54	-12.45	42.56	24.81	-17.75
5	1(3)	20	6.86	5.85	-14.72	30.92	11.64	-19.28
6	2(4)	20	7.98	7.21	-9.65	52.29	37.60	-14.69
Mean			-12.27%			-17.24%		

Table 5.6: Average travel time for the 5 NB scenarios

#	Incident Characteristics		Average Travel Time			Comparison with Scenario 0		
	Lanes	Dur. (min)	Before	After	Change (%)	Before (%)	After (%)	Change (%)
0	no incident		5.24	5.24	0.00	0.00	0.00	0.00
7	2(5)	15	5.30	**	**	1.15	**	**
8	3(5)	15	5.55	5.26	-5.23	5.92	0.38	-5.54
9	3(5)	20	5.81	5.33	-8.26	10.88	1.72	-9.16
Mean			-6.74%			-7.35%		

Table 5.7: Average travel time for the 5 SB scenarios

#	Incident Characteristics		Average Travel Time			Comparison with Scenario 0		
	Lanes	Dur. (min)	Before	After	Change (%)	Before (%)	After (%)	Change (%)
0	no incident		5.24	5.24	0.00	0.00	0.00	0.00
10	1(5)	15	5.24	**	**	0.00	**	**
11	2(5)	20	7.05	5.97	-15.32	34.54	13.93	-20.61
12	3(5)	30	8.69	8.38	-3.57	65.83	59.92	-5.91
Mean			-9.45%			-13.26%		

Table 5.8 reports the scenarios related to incidents on arterials and at a freeway on-ramp. As mentioned above, only scenarios 13 and 15 are used for the computation of the mean travel time reduction.

Table 5.8: Average travel time for Alton (WB and EB) and on-ramp scenarios

#	Incident Characteristics		Average Travel Time			Comparison with Scenario 0		
	Lanes	Dur. (min)	Before	After	Change (%)	Before (%)	After (%)	Change (%)
0	no incident		5.24	5.24	0.00	0.00	0.00	0.00
13	2(3)	20	5.26	5.26	0.00	0.38	0.38	0.00
14	2(3)	20	5.26	**	**	0.38	**	**
15	1(2)	15	5.26	5.24	-0.38	0.38	0.00	-0.38
Mean			-0.19%			-0.19%		

The set of scenarios includes cases where the control plan established in response to the first incident must be re-evaluated and altered at the occurrence of the second incident (scenarios 16 and 17). Table 5.9 reports on the scenarios related to those cases.

Table 5.9: Average travel time for multiple incidents scenarios

#	Incident Characteristics		Average Travel Time			Comparison with Scenario 0		
	Lanes	Dur. (min)	Before	After	Change (%)	Before (%)	After (%)	Change (%)
0	no incident		5.24	5.24	0.00	0.00	0.00	0.00
16	2(4) 1(2)	20 15	6.78	6.35	-6.34	29.38	21.18	-8.20
17	2(4) 1(2)	20 10	8.09	7.57	-6.43	54.39	44.47	-9.92
Mean			-6.39%			-9.06%		

It is not a simple task to draw general and absolute conclusions beyond the simple observation that in all tested scenarios the implementation of the ATMS/ATIS control plans suggested by **CARTESIUS** results in an improvement of network-wide traffic conditions (except for scenario 13, where no alternative

solution is determined; it must be noted, however, that in this case, the occurrence of an incident determines a very low (0.38%) deterioration of average travel time). The mean percentage reduction of average (per vehicle) travel time across all scenarios (including scenario 13) is 7.4 percent. Compared to the *no incident* base case (scenario 0), the mean percentage increase of average travel time caused by the occurrence of incidents is reduced from 31.2 to 20.3 percent.

The impact of incident and of incident response measures varies extensively, depending on a variety of factors, that include the capacity reduction caused by the incident, its duration, the location of the incident, the availability of feasible alternative routes, and the demand/capacity ratio before, during, and after the occurrence of the incident, both on the routes directly affected by the incidents and on the remaining sections of the network that are directly or indirectly affected by incident response strategies.

As noted, the scenarios can be classified according to the location of the incident. From the results presented in Tables 5.4 through 5.9 it can be observed that, on average, the highest performance improvement is experienced in the scenarios related to incidents along the I-405 freeway, southbound, with a mean reduction in average travel time of 12.27 percent, and a mean difference in the travel time increase caused by the incident, of 17.24 percent (compared to scenario 0, see Table 5.5). The lowest improvements are observed for incidents occurring on surface streets. In those cases, as shown in Table 5.8, the effect of incidents is marginal; thus, the corresponding improvement of traffic conditions is also very small. Performance does not deteriorate for the cases of multiple incidents (Table 5.9), which seems to demonstrate the effectiveness of combined ATMS/ATIS strategies in dealing with multiple, concurrent sources of congestion.

5.6.3 Response Time

Given the real-time nature of the problem that the ATMS/ATIS management system is intended to address, it was important to provide a measure of the system response time, i.e., the time required by the agents to determine a list of control plans, once they have been provided by the operator with all the necessary input. For each simulated scenario, the system's maximum response time was collected and reported. The response time is partially dependent on the number of alternative solutions that must be composed and evaluated, which, in turn, depends on the availability of alternative paths. Since each agent must compose pairs of local solutions into compatible global solutions, and assess the corresponding delay reduction, the response time also depends on the number of compatible global solutions found by the agents. The problem-solving process requires, at various times, the agents to synchronize and exchange data before the final solutions are presented to the operator.

Table 5.10 shows the system response time. In all tested scenarios, the system response time is below 22 seconds, with an average of 15.3 seconds. This indicates, in practical terms, that combined system optimal ATMS/ATIS strategies can indeed be implemented in something close to real-time response.

The system response time, in all tested scenarios, is distributed fairly homogeneously around a mean value of 15.3 seconds, with a minimum of 11 seconds for scenarios 6 and 15, and maximum values of 21 and 22 seconds, for scenarios 2 and 3. This difference can be explained by the difference in the number of available alternative solutions. The vast majority of traffic that takes advantage of diversion strategies is freeway traffic. Northbound traffic on the I-405 freeway can be diverted using several alternative routes, depending on the location of the incident, that always include the arterial Alton Parkway, a divided arterial, with three lanes per direction and two protected left-turn lanes at most intersections. Traffic can potentially be diverted through the I-5 freeway (leaving the freeway at the Alton off-ramp), or through the arterial Irvine Center Drive, that has relatively low cross traffic until the intersection with Alton Parkway (see Figure 1). Furthermore, the available CMS messages on Alton WB can be used to direct traffic back to the I-405 using either Irvine Center Drive, Sand Canyon Avenue, or Jeffrey Road.

When looking for solutions, the agents must evaluate all feasible combinations of these alternatives. Because of the location of CMS, southbound traffic on the I-405 and both north and southbound traffic on the I-5 have slightly fewer diversion options. Thus, in general, the agents must analyze fewer possible alternatives, and the system response time for these cases is lower. The location of the CMS on the I-405 southbound, downstream of Jeffrey Road, reduces the number of freeway off-ramps that can be effectively used for diversion to two, thus limiting the number of potential alternative routing solutions. An analogous situation occurs for incidents occurring on the I-5 freeway, where, because of the limited number of CMS, southbound traffic can be diverted by **CARTESIUS** only through the SR-133 freeway, to the I-405, and northbound traffic, depending on the incident location can only use either the I-405 or the off-ramp at Alton Parkway.

It is observed that the system response time does not vary significantly for scenarios that involve the occurrence of multiple incidents (scenarios 16 and 17), compared to the cases of a single incident. This can probably be explained by considering that, in general, a higher number of problems on one hand increases the number of possible solutions that must be analyzed, but on the other, reduces the number of possible alternatives, thus reducing the number of feasible solutions for which the corresponding delay must be computed.

Table 5.10: ATMS/ATIS Management System: Maximum Response Time

Scenario #	Location	Lanes Blocked (Total)	Response Time (seconds)
1		1(4)	**
2	405 NB	2(4)	22
3		2(4)	21
4		1(4)	15
5	405 SB	1(3)	15
6		2(4)	11
7		2(5)	**
8	5 NB	3(5)	15
9		3(5)	15
10		1(5)	**
11	5 SB	2(5)	16
12		3(5)	15
13	Alton EB	2(3)	15
14	Alton WB	2(3)	**
15	NB 405 on-ramp	1(2)	11
16	405 NB Alton WB	2(4) 1(2)	15
17	405 SB on-ramp	2(4) 1(2)	13

Note: Measures were obtained using a SUN Ultra 30 Workstation, with an Ultra Sparc 2 processor

5.6.4 Effect of Signal Control Plans

As part of the analysis of system performance, a quantitative assessment of the effect of signalization was performed within the integrated ATMS/ATIS control plans proposed by **CARTESIUS**. For three of the scenarios for which diversion through the arterial system is recommended (scenarios 2, 5, and 9, with incidents respectively on the I-405 northbound, I-405 southbound, and I-5 southbound), an additional simulation was performed, in which the plans for signals and ramp meters suggested by **CARTESIUS** were **not** transmitted to the traffic simulator. In each case, the network performance, for the modified scenarios (2', 5' and 9') was compared to that of the corresponding basic scenarios (2, 5, and 9), in which

the complete control directives were transmitted to the simulator. These tests were aimed at estimating the synergetic effect of integrated response control plans, by estimating the reduction in the network performance caused by the lack of integration between traffic diversion control and signal and meter control.

Table 5.11 describes the results for test scenario 2, with an incident on the Interstate 405 freeway, northbound.

Table 5.11: Travel time with and without signal control for an incident on 405 NB

Signal Coord	Incident Characteristics			Average Travel Time			Comparison with Scenario 0		
	#	Lanes	Dur. (min)	Before	After	Change (%)	Before (%)	After (%)	Change (%)
	0	no incident		5.24	5.24	0.00	0.00	0.00	0.00
Yes	2	2(4)	20	6.76	6.32	-6.51	29.00	20.61	-8.39
No	2'			6.76	6.87	1.63	29.00	31.11	+2.11
1.1.1	Difference			-8.14%			-10.05%		

Table 5.12 describes the comparison for test scenario 5, with an incident on the Interstate 405 freeway, southbound.

Table 5.12: Travel time with and without signal control for 405 SB incident

Signal Coord	Incident Characteristics			Average Travel Time			Comparison with Scenario 0		
	#	Lanes	Dur. (min)	Before	After	Change (%)	Before (%)	After (%)	Change (%)
	0	no incident		5.24	5.24	0.00	0.00	0.00	0.00
Yes	5	1(3)	20	6.86	5.85	-14.72	30.92	11.64	-19.28
No	5'			6.86	6.46	-5.83	30.92	23.28	-7.64
1.1.2	Difference			-8.89%			-11.64%		

Table 5.13 describes the comparison for test scenario 9, with an incident on the Interstate 5 freeway, northbound.

Table 5.13: Travel time with and without signal control for an incident on 5 NB

Signal Coord	Incident Characteristics			Average Travel Time			Comparison with Scenario 0		
	#	Lanes	Dur. (min)	Before	After	Change (%)	Before (%)	After (%)	Change (%)
	0	No incident		5.24	5.24	0.00	0.00	0.00	0.00
Yes	9	3(5)	20	5.81	5.33	-8.26	10.88	1.72	-9.16
No	9'			5.81	5.35	-7.92	10.88	2.10	-8.78
Difference				-0.34%			-0.38%		

The two scenarios that involve incidents on the I-405 freeway are characterized by a clear superiority of the integrated control compared to the partial one. The scenario that involves diversion from the I-5 freeway does not show such a noticeable performance gain. This difference can be explained perhaps by considering the characteristics of the major alternative routes for the freeway traffic in the three cases. In the first case (scenario 2), freeway traffic is diverted through Irvine Center Drive, Alton Parkway, and Sand Canyon Avenue, and it crosses nine signalized intersections and one metered freeway on-ramp. In the second scenario (scenario 5), traffic is directed through Sand Canyon Avenue, Alton Parkway, to the I-5 freeway, crossing six intersections and one metered ramp. Also, in both cases, as a consequence of congestion, arterial traffic directed to the freeway (coming from Alton Parkway) is advised to remain on the surface street, modifying the original traffic distribution even further. In the last case (scenario 9), the alternative route crosses only five signalized intersections, and traffic directed from Alton to the freeway is not affected by the diversion because no CMS is available on Alton Parkway westbound, upstream of the freeway on-ramp.

5.7 Evaluation Summary

A set of 18 test scenarios was created, by running simulations of 90-minute peak periods and artificially injecting incidents (temporary reductions in the capacity of a link), by varying such characteristics as the incident location, the associated loss of capacity and the duration of the capacity reduction. In scenario 0, no incident was included, in order to obtain estimates of the *basic* network average and total travel time and distance. Such estimates were used to assess the relative performance deterioration caused by the occurrence of incidents, across

the different scenarios. Thus, for each case, the average travel time increase with respect to scenario 0 was measured. Two scenarios were also included that describe cases for which *two* incidents were injected simultaneously. In both scenarios, the first incident was simulated on the freeway, while the second was injected along one of the major paths that had been chosen by the system as a bypass for the first incident. The purpose of the latter tests was to analyze the system's response to multiple incidents with interacting effect on each other.

For each scenario, the MOEs provided by the simulator were collected. For each test case, two simulations were executed: one, the *before* case, using the default control (no CMS message and the default, time-of-day signal and ramp meter timing plan), and one, the *after* case, implementing the integrated ATMS/ATIS control suggested by the agents, in response to the notification of the occurrence of congestion. The comparison of the network performance, through the implementation of the two forms of control (non-ATMS/ATIS versus ATMS/ATIS), provided a measure of the performance increase that can be expected when the default control is substituted with the control plans based on the responses suggested by **CARTESIUS**.

The analysis shows that the implementation of the control plans proposed by **CARTESIUS** results, in general, in a reduction in the average and total network-wide travel time. The average and total traveled distance is not significantly affected by the alternative control, even though the control plans, in all test scenarios, included the use of CMS messages. This result should be perhaps attributed to the limited size of the network.

The quality of the improvement, in general, varies according to the availability of alternative routes, the amount of spare capacity on those routes, and the demand/capacity ratio on routes affected by congestion. Thus, the scenarios, described in Table 5.1, are partitioned according to the location of the incident. For each incident, its characteristics are: the location, the associated capacity reduction (number of lanes closed versus total number of lanes), and the duration of the capacity reduction. Table 1 shows the total and average (per vehicle) travel time resulting from the simulated scenarios. The percentage travel time difference between the *before* and the *after* case is shown. Furthermore, in order to obtain a normalized measure of the travel time reduction across the scenarios, the last two columns of the Table 5.1 show, for each case, the average travel time increase with respect to scenario 0, which corresponds to a *no incident* situation, and thus represents a common base case. For each case, the percent difference in per-vehicle travel time corresponding to the two forms of control (*non ATMS/ATIS* vs. *ATMS/ATIS*) is reported.

It is not a simple task to draw general and absolute conclusions beyond the simple observation that in almost all tested scenarios the implementation of the ATMS/ATIS control plans suggested by **CARTESIUS** results in an improvement of network-wide traffic conditions. In scenarios 1, 7, 10, and 14, the small capacity

reduction did not have noticeable effects on traffic flow, thus no incident notification was received by **CARTESIUS**. Scenario 13 is the only one, among those for which **CARTESIUS** was used, in which no travel time reduction is observed; this is due to the fact that no alternative solution to the default control was found. It must be noted, however, that in this case, the occurrence of an incident results in a very low deterioration of average travel time, with respect to scenario 0 (0.4%). The mean percentage reduction of average (per vehicle) travel time across the 13 scenarios for which **CARTESIUS** received an incident notification is 7.4 percent. Compared to scenario 0 (the *no incident* case), the mean percentage increase of average travel time, caused by the occurrence of incidents, is reduced from 32.6 to 20.3 percent.

The reduction in average travel time ranges between 0.0 and 15.3 percent. The variation is due both to the different duration and capacity reduction of the incident and, perhaps more importantly, to the characteristics of its location, such as the flow to capacity ratio and the availability of alternative routes.

On average, higher performance improvements are experienced in the scenarios related to incidents occurring on freeway sections (the first 12 scenarios), in particular on the I-405 freeway southbound (scenarios 4-6). The lowest improvements are observed for incidents occurring on surface streets (scenarios 13-15). In those cases, the effect of incidents is marginal (0.4% travel time increase), thus the improvement of traffic conditions is also very small. The network performance is improved by **CARTESIUS** also for the cases of multiple incidents, which seems to demonstrate the effectiveness of the approach in dealing with multiple, concurrent sources of congestion.

TABLE 5.14: Travel Time Summary for the Simulated Scenarios

#	Incident Characteristics			Total Travel Time (x10 ⁴ veh-hrs)		Average Travel Time (min./veh)				
	Location	Lanes	Length (min.)	Before	After	Before	After	Change (%)	% Change from Scenario 0 ¹	
									Before	After
0	No incident			0.380	0.380	5.24	5.24	0.0	0.0	0.0
1	405 N	1(4)	20	0.429	0.429	5.87	*5.87	**	12.0	12.0
2		2(4)	20	0.496	0.463	6.76	6.32	-6.5	29.0	20.6
3		2(4)	25	0.528	0.486	7.20	6.70	-6.9	37.4	27.5
4	405 S	1(4)	20	0.546	0.479	7.47	6.54	-12.4	42.6	24.8
5		1(3)	20	0.502	0.428	6.86	5.85	-14.7	30.9	11.6
6		2(4)	20	0.584	0.528	7.98	7.21	-9.6	52.3	37.6
7	5 N	2(5)	15	0.387	0.387	5.30	*5.30	**	1.1	1.1
8		3(5)	15	0.400	0.383	5.55	5.26	-5.2	5.9	0.4
9		3(5)	20	0.426	0.391	5.81	5.33	-8.3	10.9	1.7
10	5 S	1(5)	15	0.385	0.385	5.24	*5.24	**	0.0	0.0
11		2(5)	20	0.510	0.438	7.05	5.97	-15.3	34.5	13.9
12		3(5)	30	0.634	0.615	8.69	8.38	-3.6	65.8	59.9
13	Alton E	2(3)	20	0.381	0.381	5.26	5.26	0.0	0.4	0.4
14	Alton W	2(3)	20	0.382	0.382	5.26	*5.26	**	0.4	0.4
15	on-ramp	1(2)	15	0.383	0.381	5.26	5.24	-0.4	0.4	0.0
16	405 N	2(4)	20	0.496	0.459	6.78	6.35	-6.3	29.4	21.2
6	Alton W	1(2)	15							
17	405 S	2(4)	20	0.601	0.560	8.09	7.57	-6.4	54.4	44.5
7	on-ramp	1(2)	10							
Average:								-7.4	32.6	20.3

¹ These two columns describe the percentage difference in average travel time when each scenario is compared to scenario 0, corresponding to the *no incident* case.

* No incident notification to CARTESIUS.

As part of the analysis of system performance, a quantitative assessment of the synergistic effect of coordination between signal control and traffic diversion was performed within the integrated ATMS/ATIS strategies proposed by **CARTESIUS**. For three of the scenarios for which diversion through the arterial system was recommended due to freeway incidents (scenarios 2,5,9), an additional simulation was performed (scenarios 2b, 5b, 9b), in which the adjustment to plans for signals and ramp meters suggested by **CARTESIUS** were *not* transmitted to the traffic simulator. In each case, network performance for the modified simulations was compared to that of the corresponding scenarios, in which the complete control directives were transmitted to the simulator. These tests were aimed at estimating the synergistic effect of integrated response control plans, by computing the reduction in the network performance caused by the lack of integration between traffic diversion control and signal and meter control. Table 5.15 shows the result of these tests.

TABLE 5.15: Travel Time Summary for Freeway Incident

Signal Coord.	Scenario #	Incident Characteristics		Average Travel Time (min./veh)			Comparison with Scenario 0	
		Lanes	Duration (min.)	Before	After	Change (%)	Before (%)	After (%)
	0	No incident		5.24	5.24	0.0	0.0	0.0
yes	2	2(4)	20	6.76	6.32	-6.5	29.0	20.6
no	2b				6.87	+1.6		31.1
yes	5	1(3)	20	6.86	5.85	-14.7	30.9	11.6
no	5b				6.46	-5.8		23.3
yes	9	3(5)	20	5.81	5.33	-8.3	10.9	1.7
no	9b				5.35	-7.9		2.1

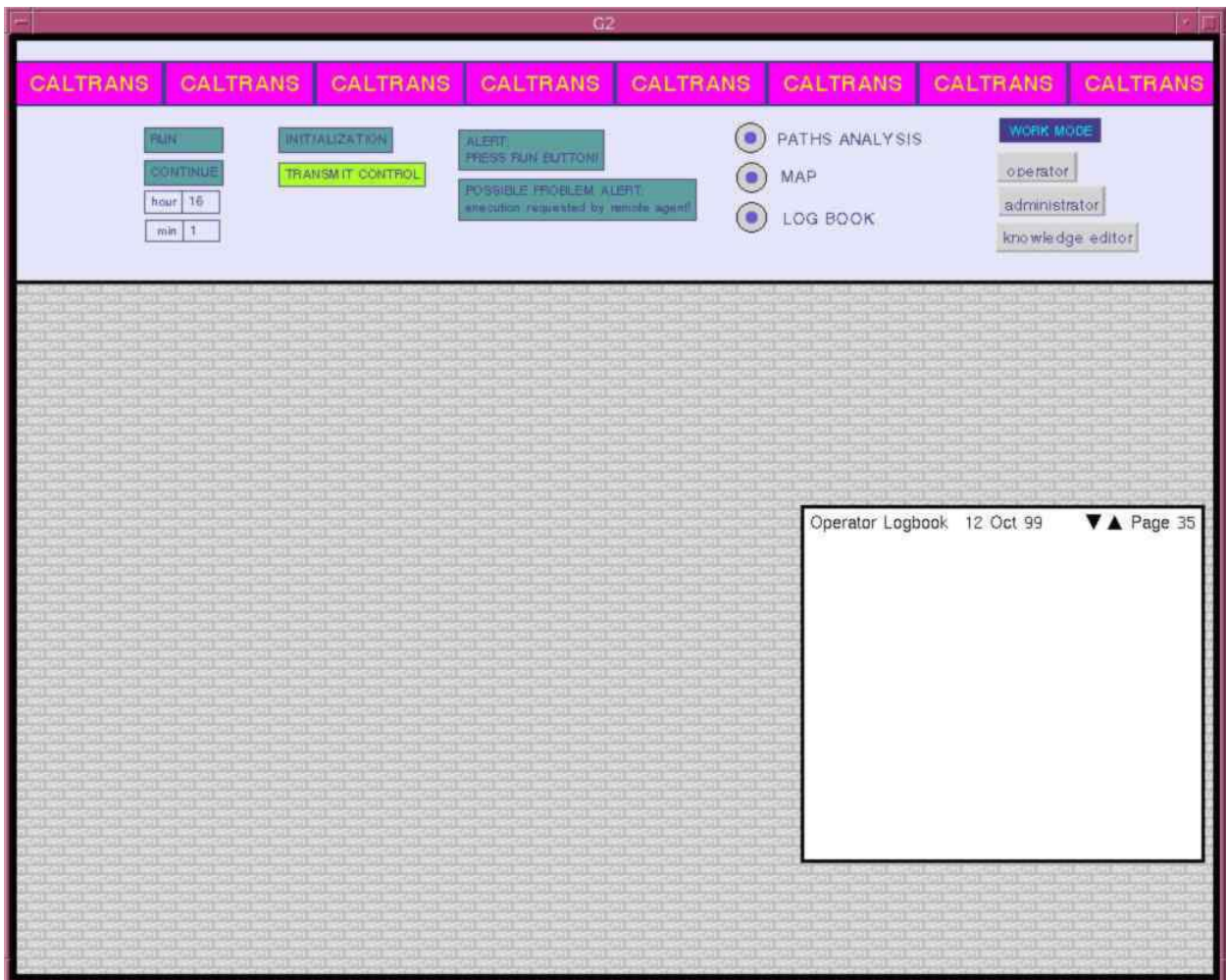
Chapter 6

Sample Application of TRICEPS

While only an on-line demonstration of **TRICEPS** can fully display the research and operational capabilities of this ATMIS, an illustrative example is presented via captured screen shots from a live application of **TRICEPS** using simulated data.

Response phases include:

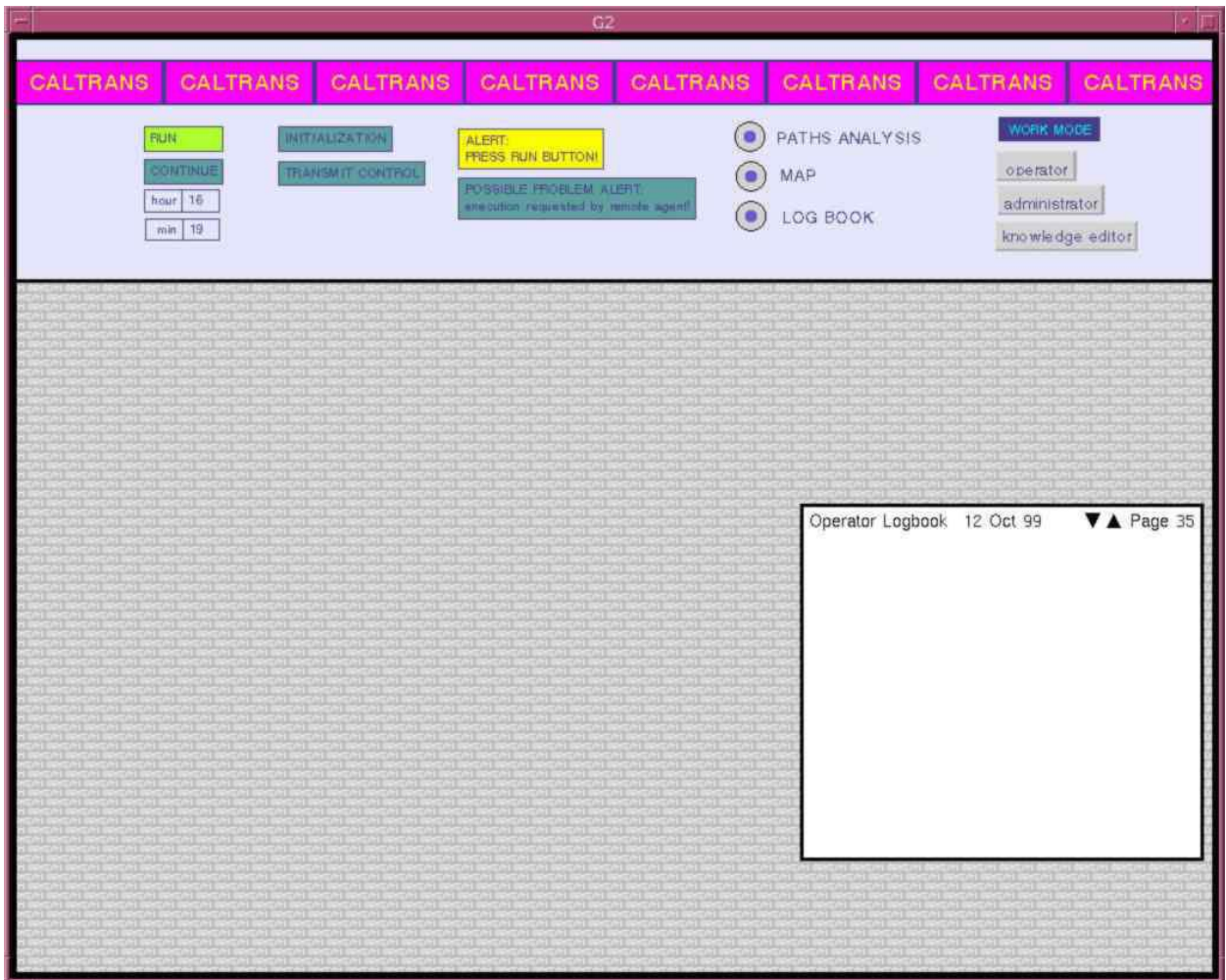
- 6.1 System Monitoring and Incident Detection (slides 1-3)
- 6.2 Initial Response and Inter-agent Communication (slides 4-7)
- 6.3 Generation of Response Strategies (slides 8-13)
- 6.4 Strategy Negotiation (slides 14-18)
- 6.5 Strategy Selection and Implementation (slides 19-23)
- 6.6 Incident Recovery and System Monitoring (slides 24-28)



6.1 System Monitoring and Incident Detection (slides 1-3)

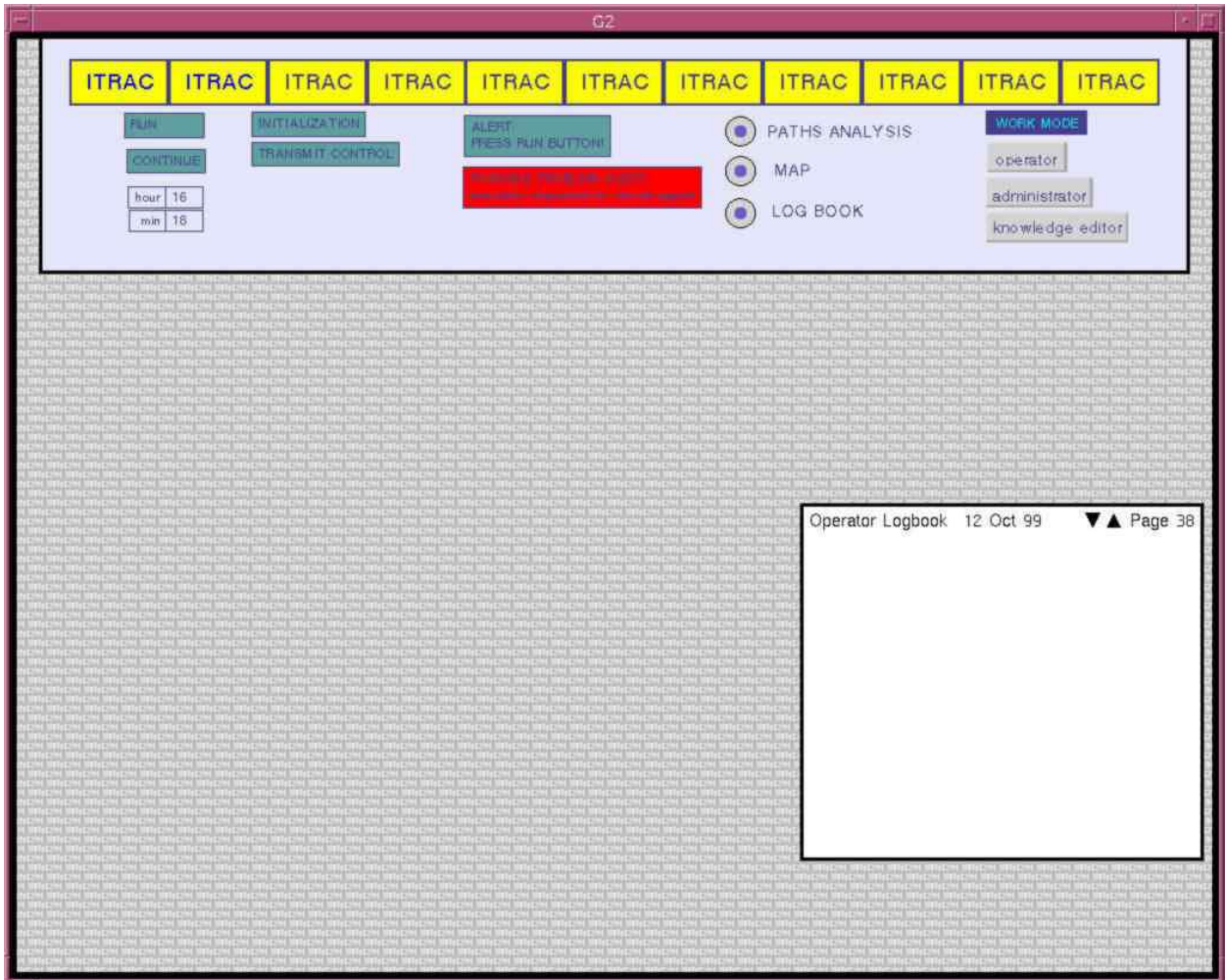
Screen 1

The initial **CARTESIUS** display for the freeway agent, Caltrans' District 12 TMC (note the purple color bar across the top of the screen; has a yellow bar), shows the various TMC operator options, including graphic displays of networks and sensor data streams, as well as communication options with the field and other options. The initial display for the arterial agent, Irvine's ITRAC, is identical except for the yellow agent identifier bar at the top of the screen (this screen is not shown).



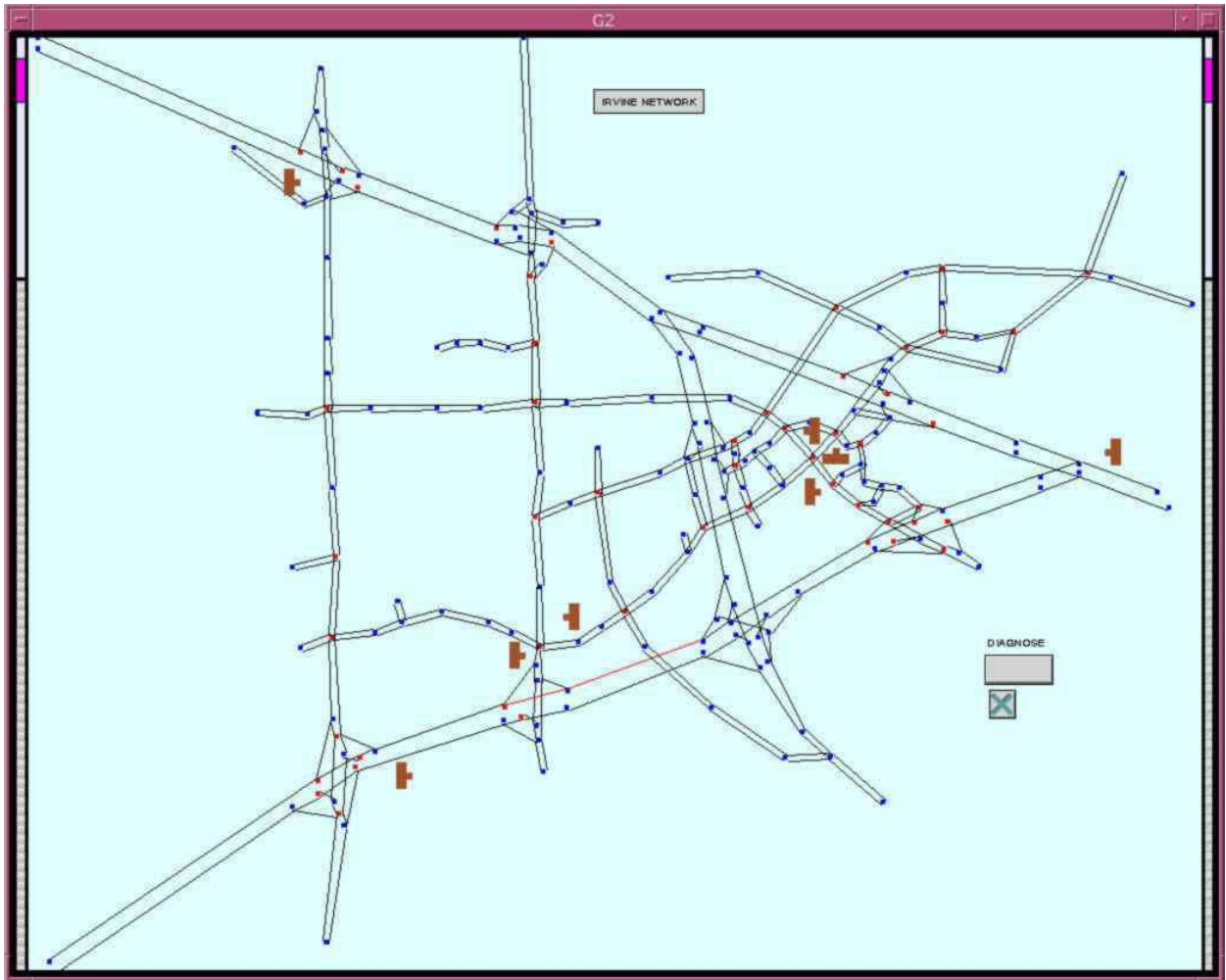
Screen 2

In this and the following **CARTESIUS** screens, first the freeway operator (henceforth, D12) is notified of a potential incident on a freeway link and then conveys this information to the arterial operator (henceforth ITRAC). The incident detection process may use internal and/or external algorithms and then notifies the operator for verification.



Screen 3

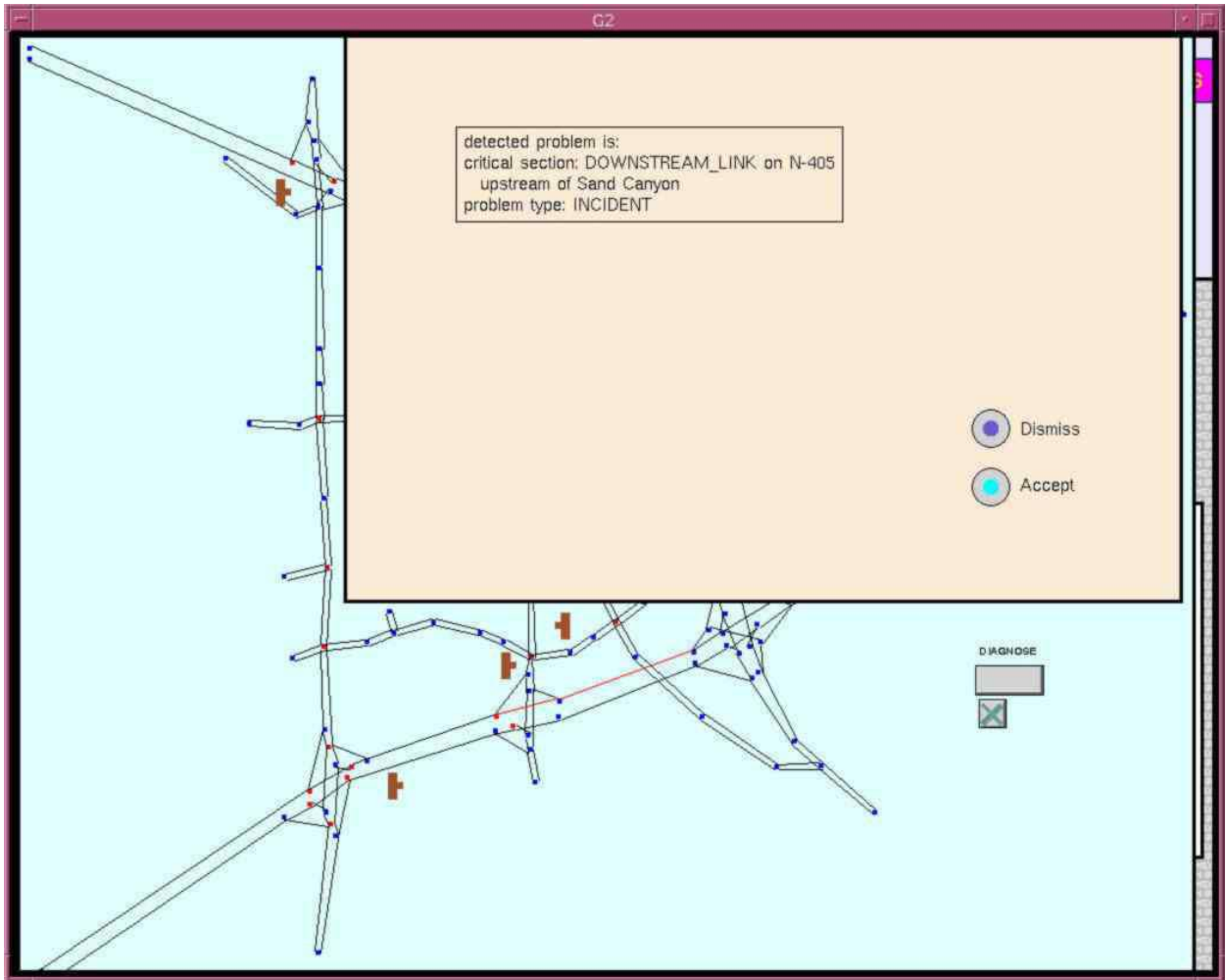
ITRAC: Notice of the freeway incident alert is received by ITRAC agent who then surveys the ITRAC sub-network for possible problems.



6.2 Initial Response and Inter-agent Communication (slides 4-7)

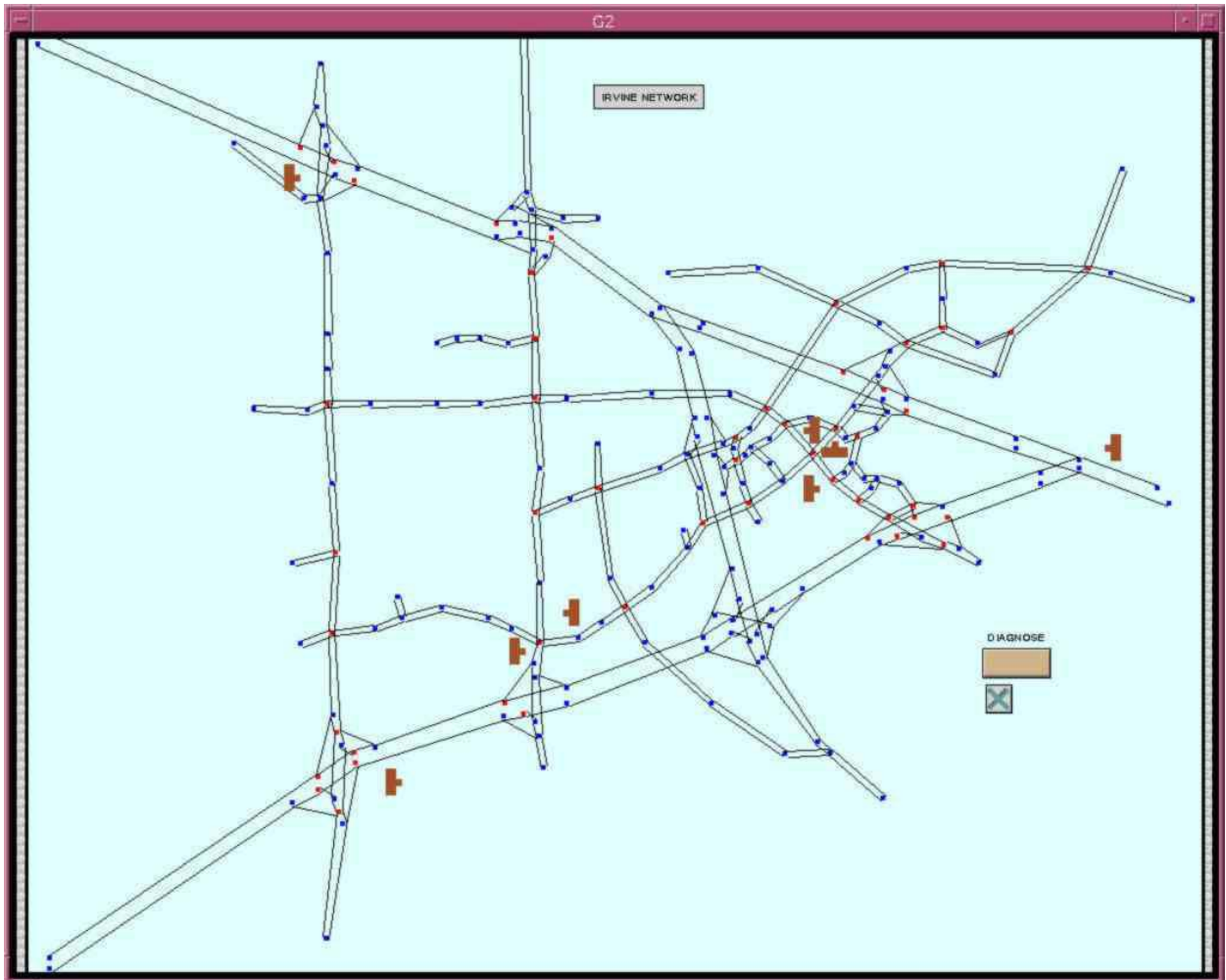
Screen 4

D12: **CARTESIUS** displays suspected incident link in red (I-405 northbound between the second and third interchanges from the left) and requests D12 operator verification (see next screen). At this point the operator may use a variety of verification techniques, including direct observation via CCTV cameras or indirect verification via highway patrol reports, or 9-1-1 calls.



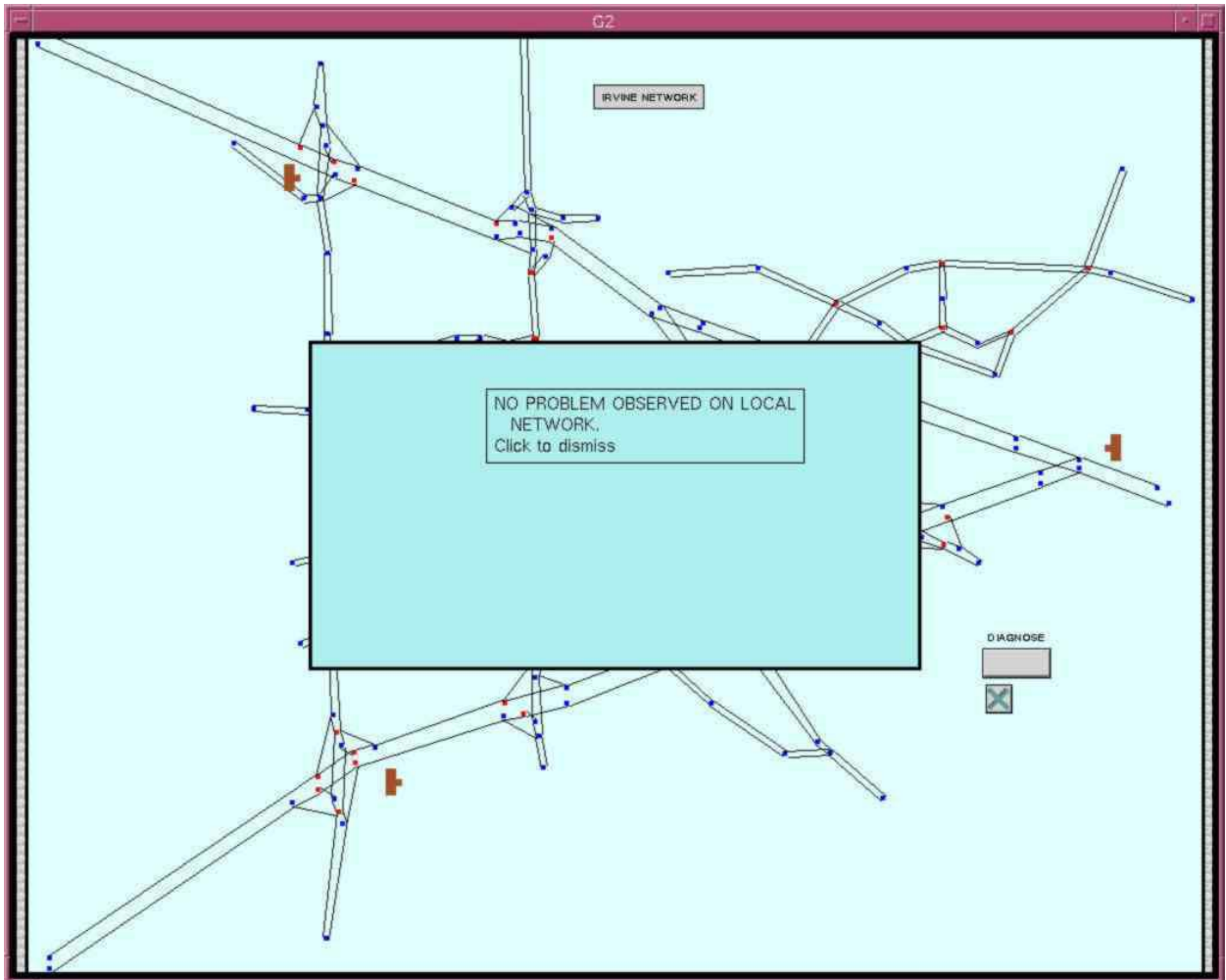
Screen 5

D12: Freeway agent diagnosis is completed with operator confirmation (or rejection) of the now verified incident.



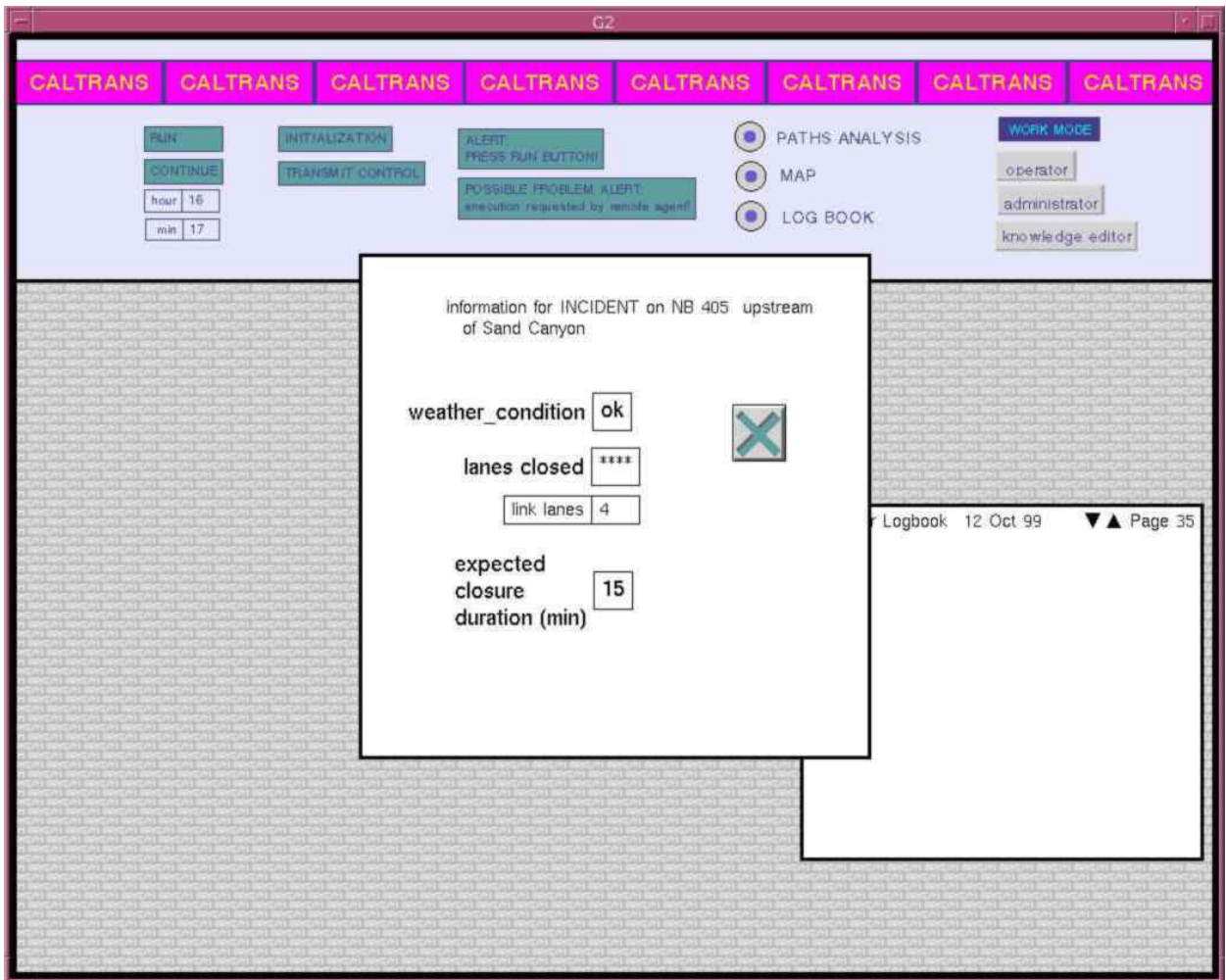
Screen 6

ITRAC: The arterial operator, after receiving the freeway incident advisory, performs an independent system diagnosis on the local sub-network.



Screen 7

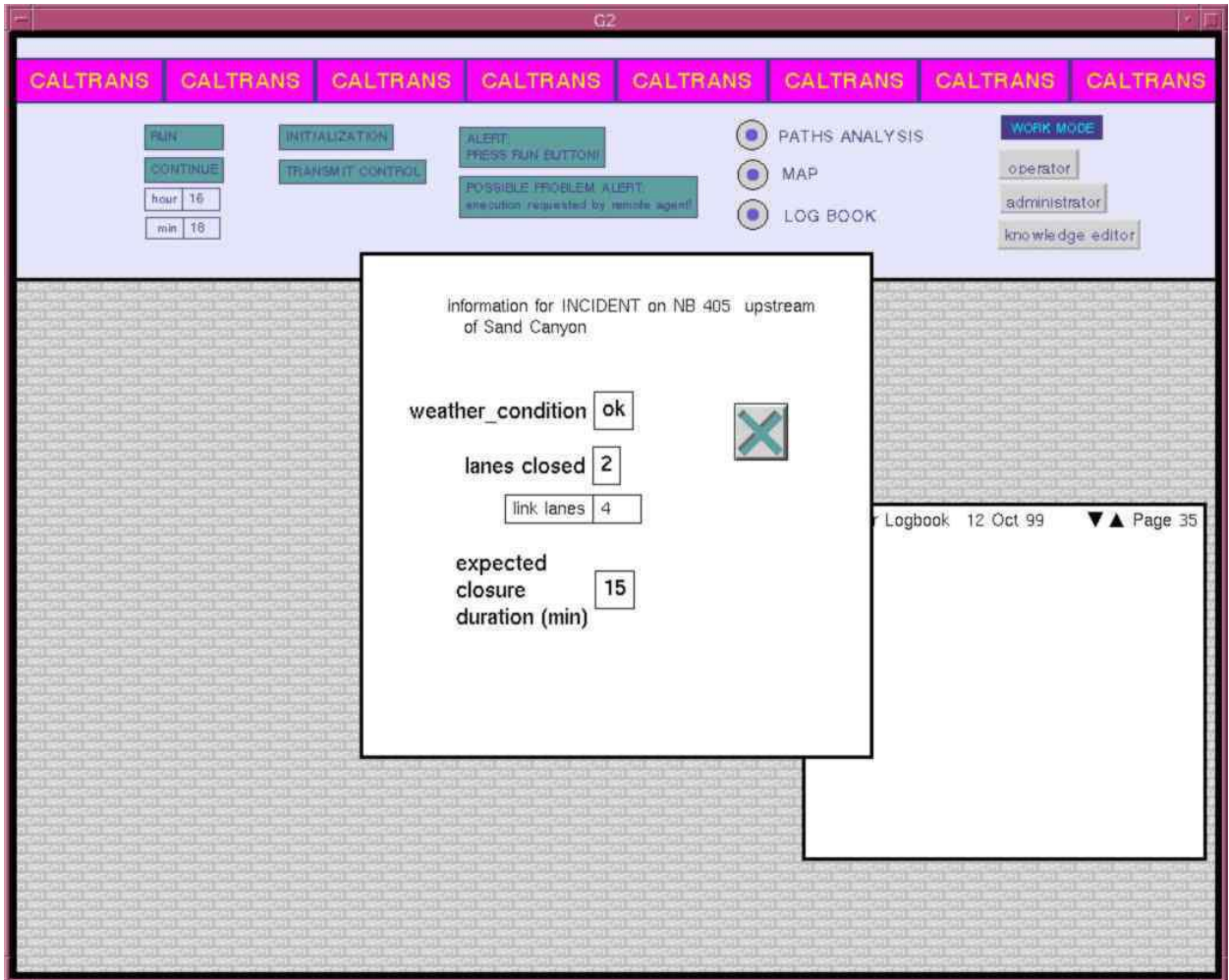
ITRAC: The **CARTESIUS** arterial agent detects no local problems. This information is shared with the D12 freeway agent.



6.3 Generate Response Strategies (slides 8-13)

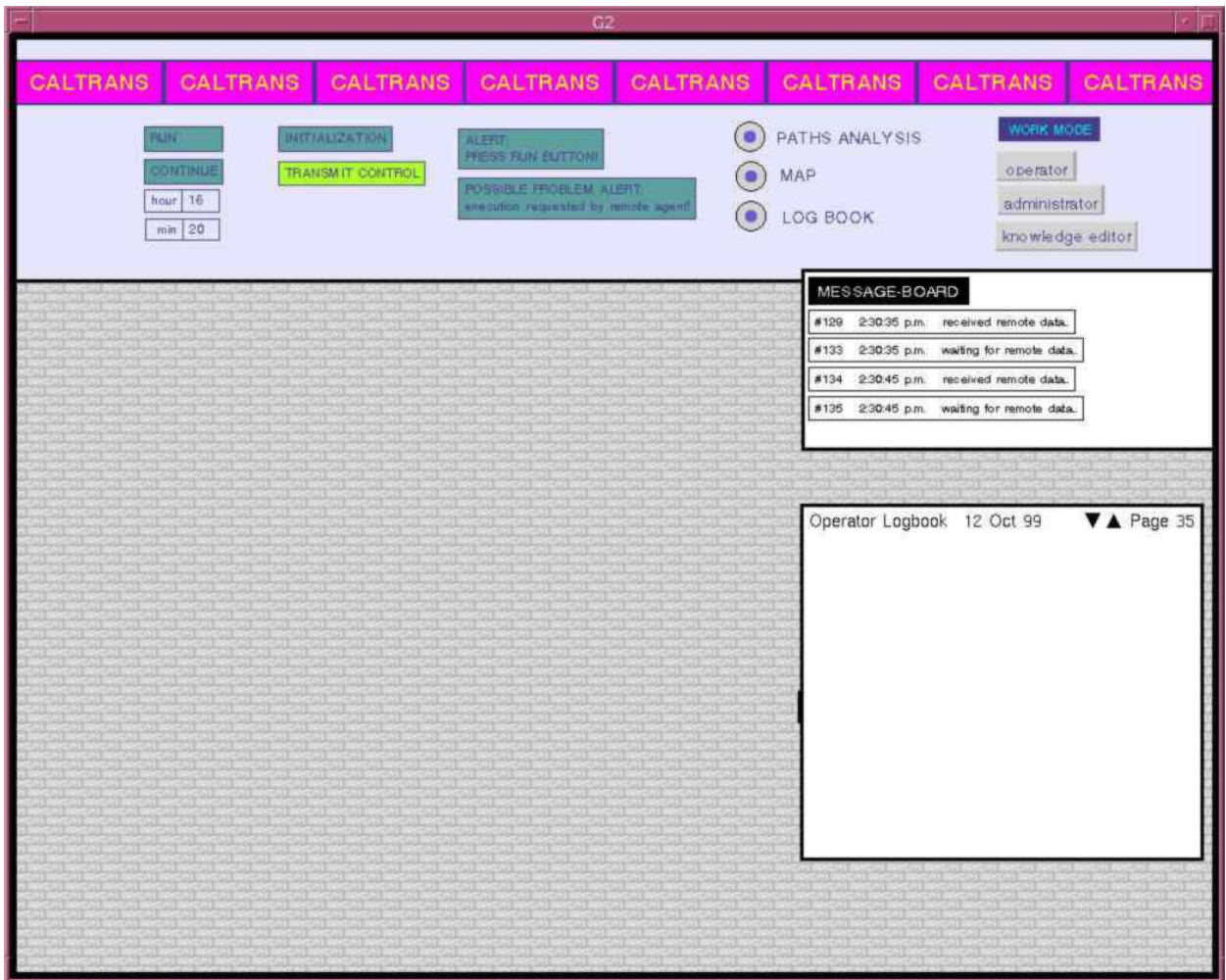
Screen 8

D12: CARTESIUS prompts for basic parameters for the verified incident.



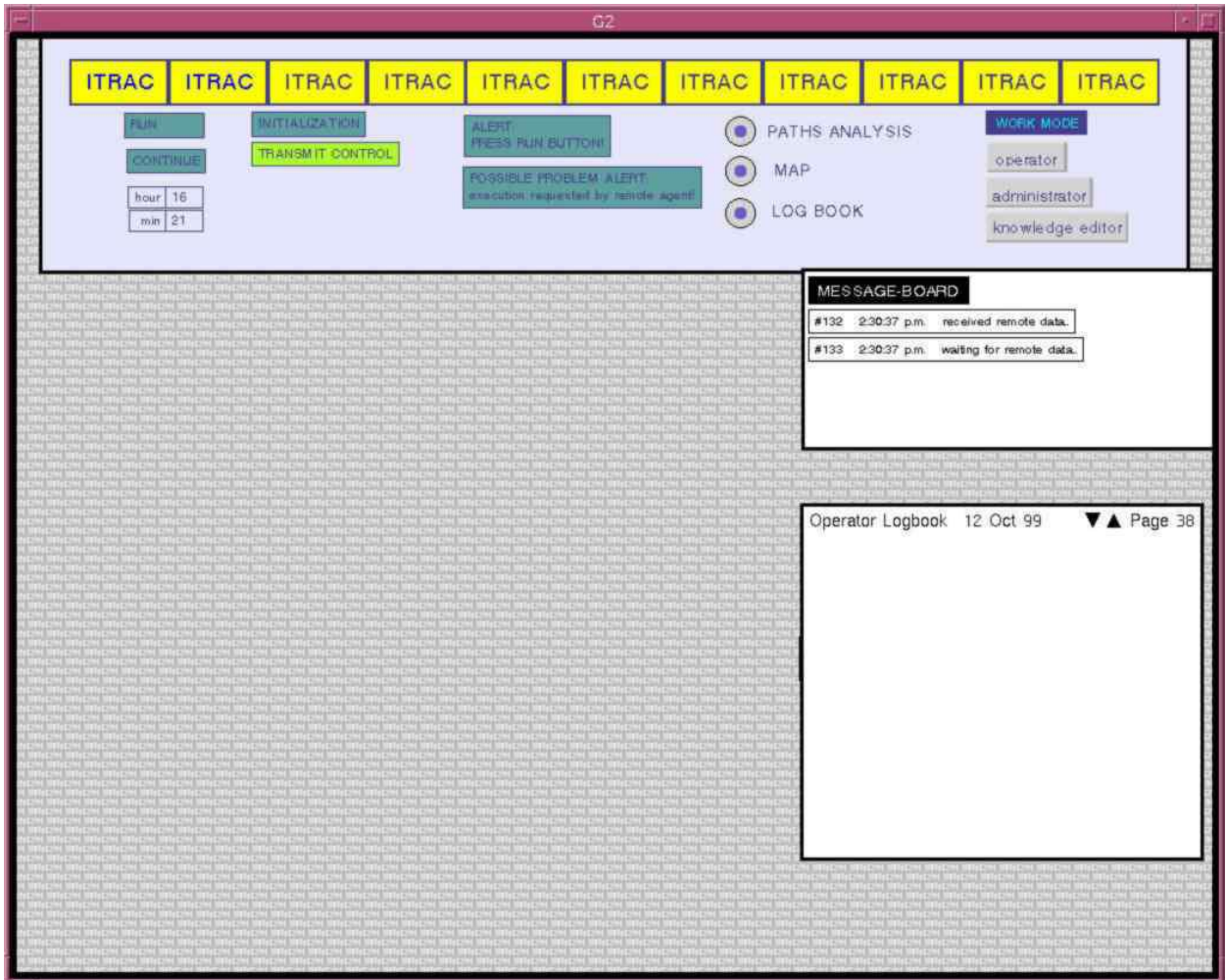
Screen 9

D12: Freeway agent completes initial incident assessment, including weather conditions, number of lanes affected, and expected closure duration.



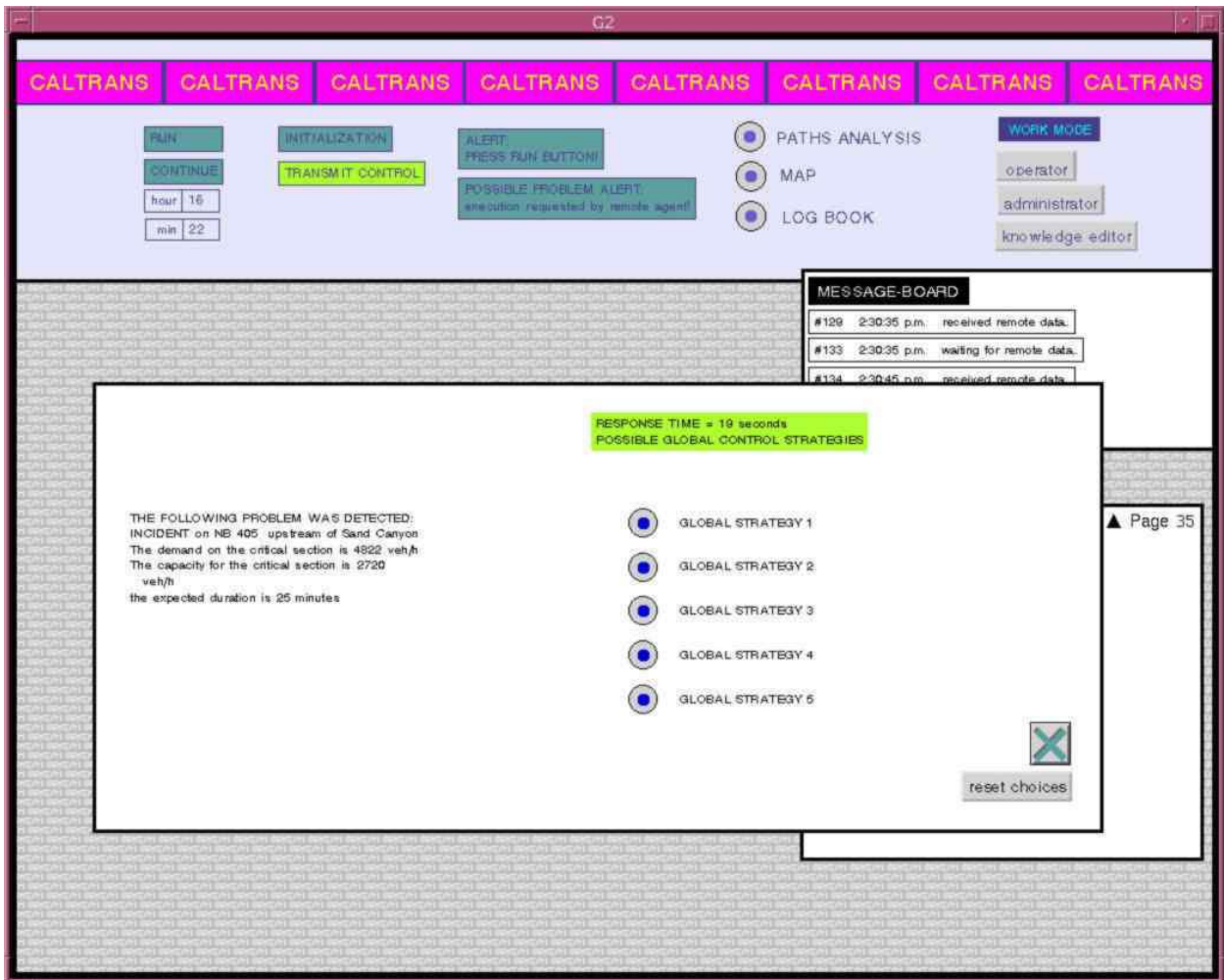
Screen 10

D12: The **CARTESIUS** freeway agent begins analysis of the congestion problem and the process of generating global solutions for sharing with other agents (note interaction with the arterial agent on message board).



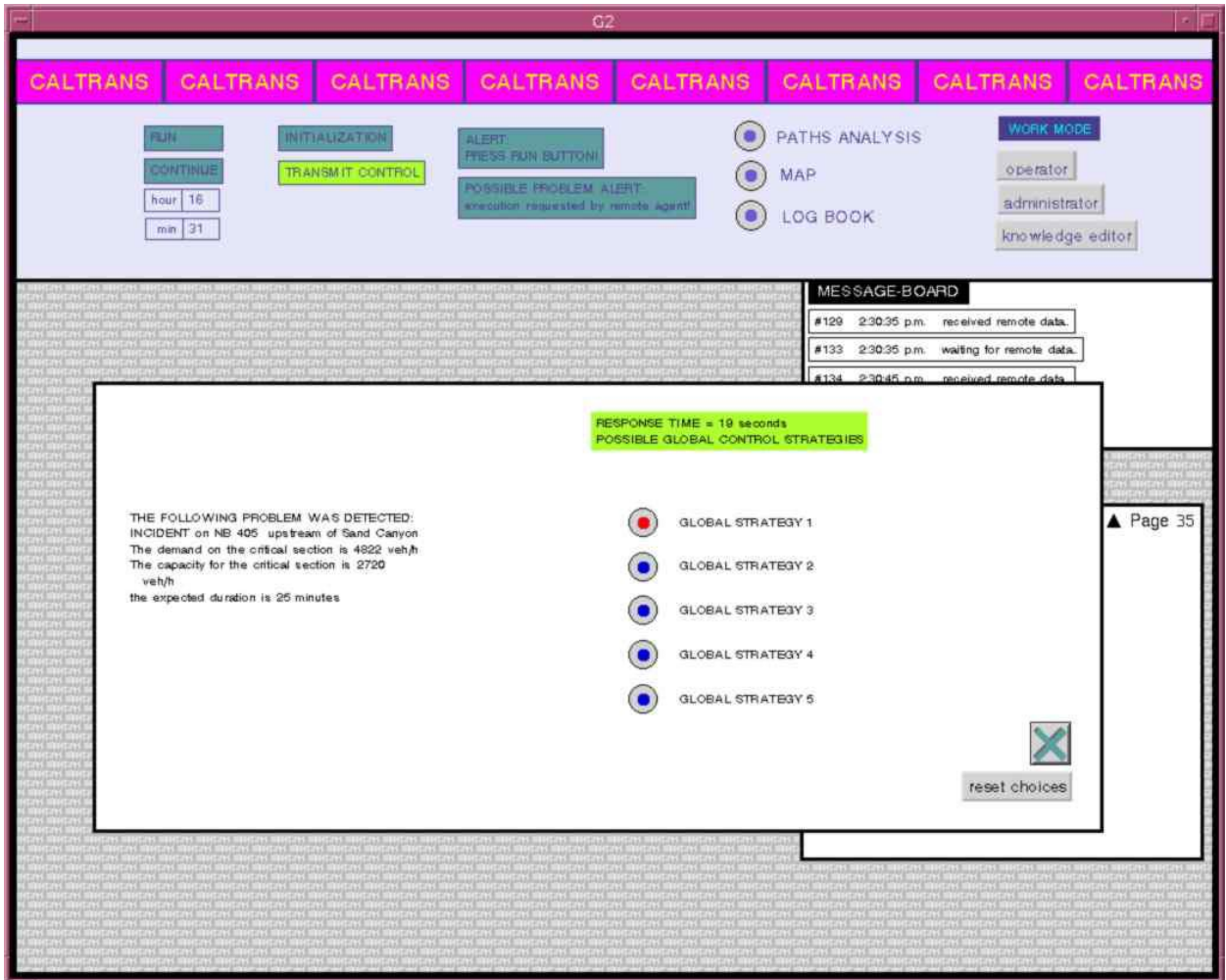
Screen 11

ITRAC: Arterial agent solving problem (note interaction with freeway agent on the message board). Note that only the freeway agent is formally addressing an identified problem but simultaneous events, recurrent congestion could place the arterial agent into active problem solving. At this point, the arterial agent is waiting for input from the freeway agent.



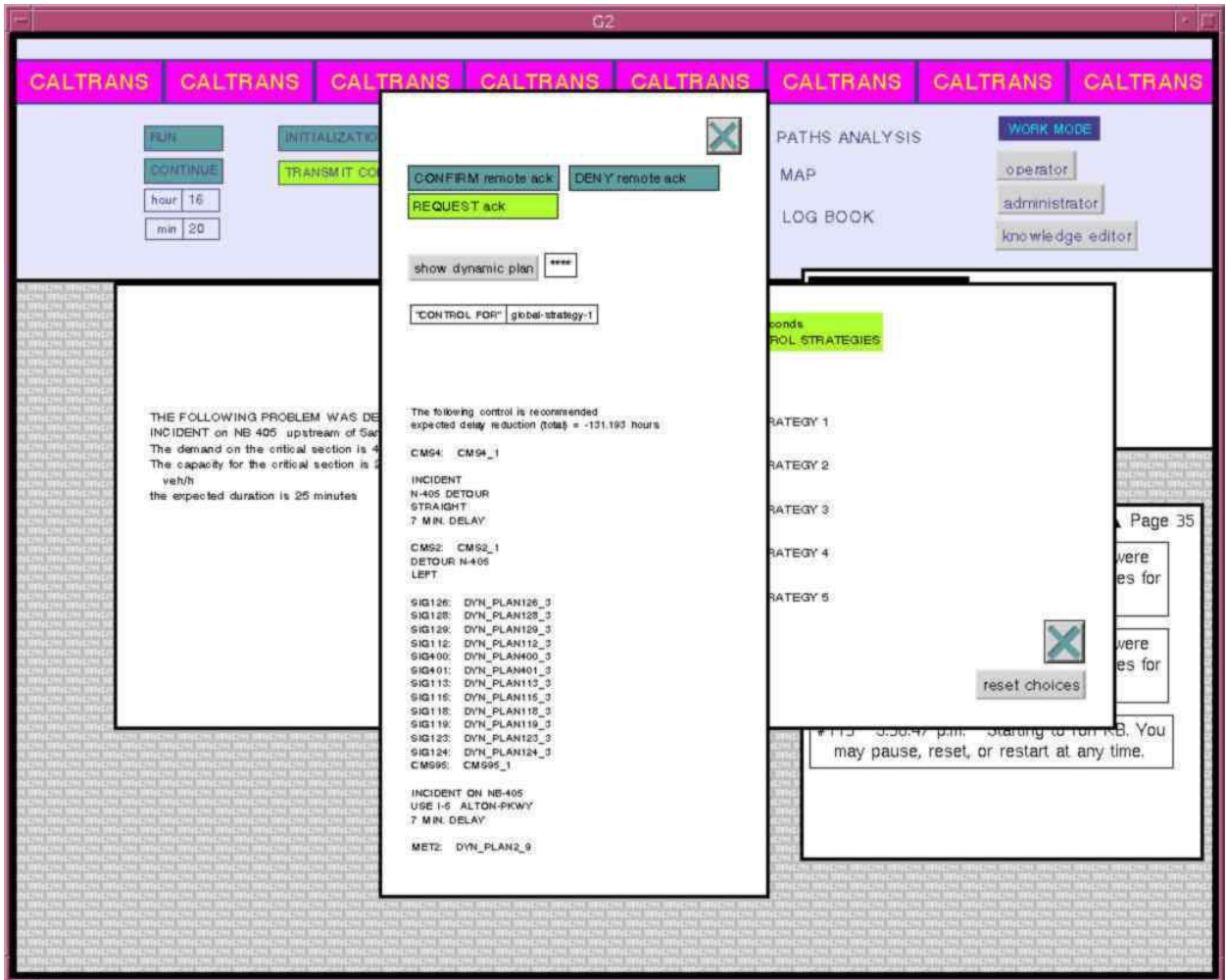
Screen 12

D12: The **CARTESIUS** freeway agent has generated five (current maximum) potential solutions, ranked in order of estimated reduction in global (system-wide) delay. Since this agent has the only current problem, the operator must select a preferred strategy to forward to other agents to begin negotiation. Note that each agent is always free to act unilaterally within their own jurisdictions but typically the participation of at least one other agent is required to fully implement the global strategy's management components to achieve the global solution.



Screen 13

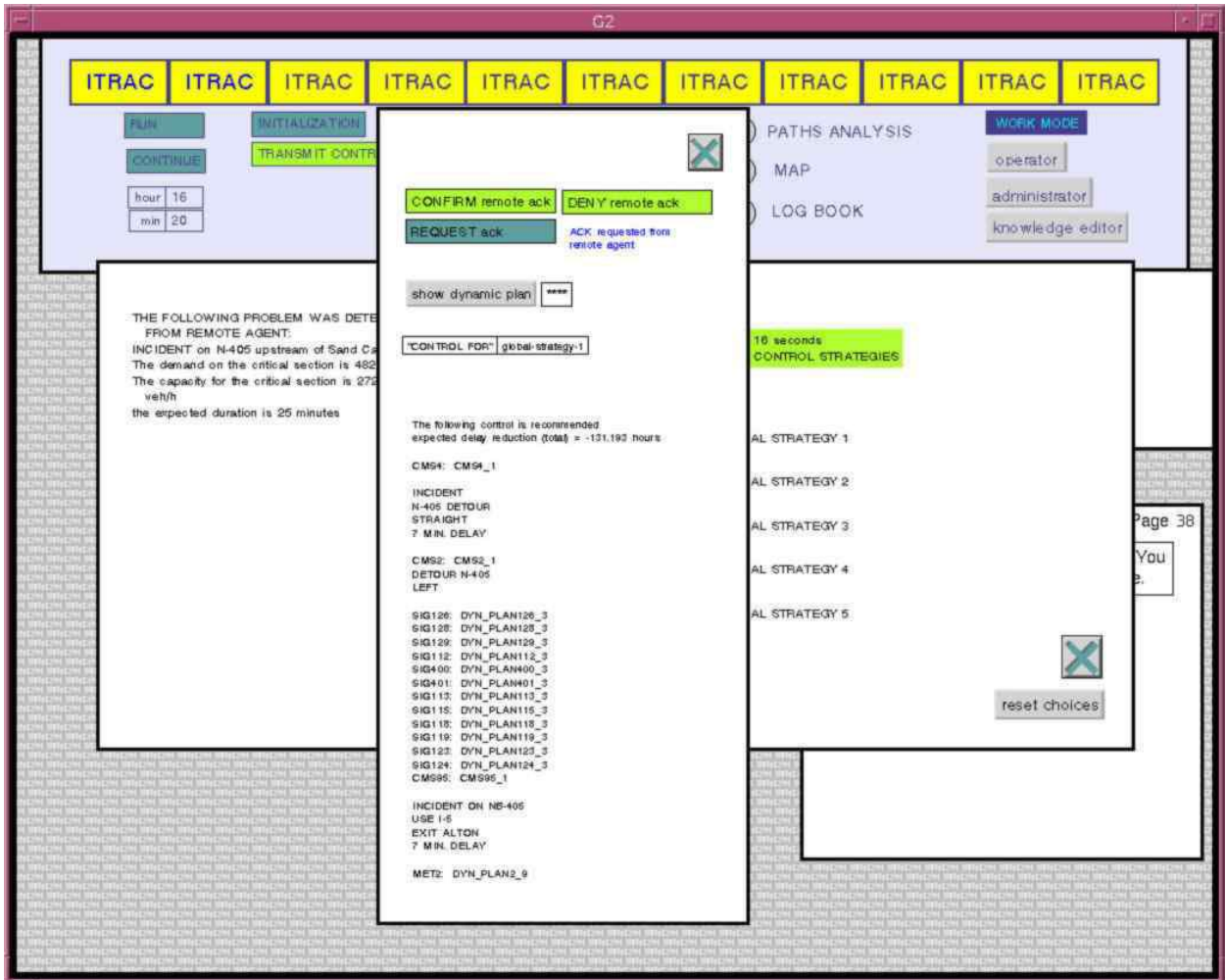
D12: The **CARTESIUS** freeway agent selects Global Strategy 1 to review as a potential preferred response plan to begin negotiations.



6.4 Strategy Negotiations (slides 14-18)

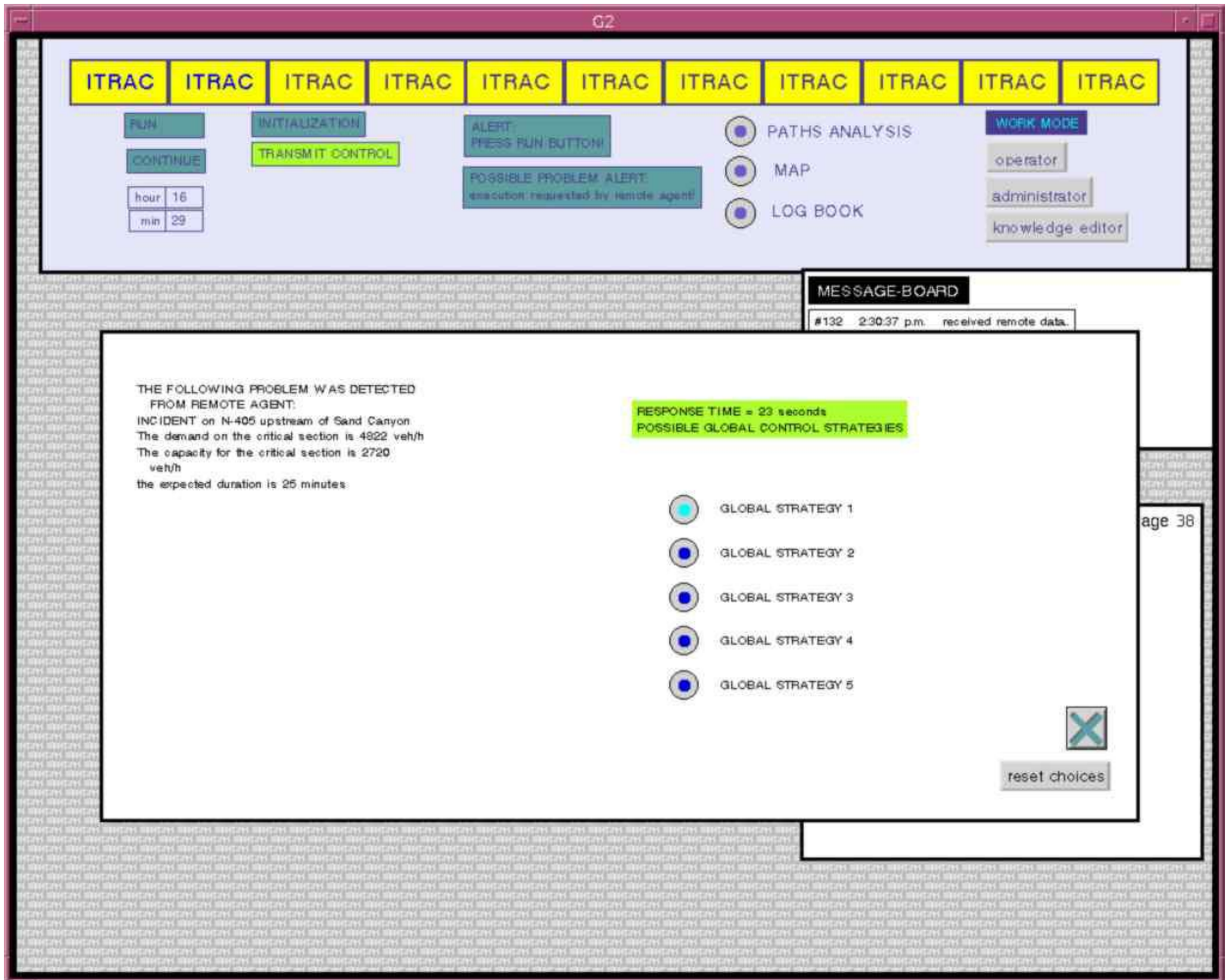
Screen 14

D12: The **CARTESIUS** freeway operator examines the response plan for Global Strategy 1 (GS1). This plan includes plan components for ramp metering rates and CMS on the freeway, and the corresponding plans for arterial signal control and CMS. The freeway operator selects this plan as the preferred plan and requests acknowledgement of this preference from other agents (“REQUEST ack” option is selected), initiating negotiations.



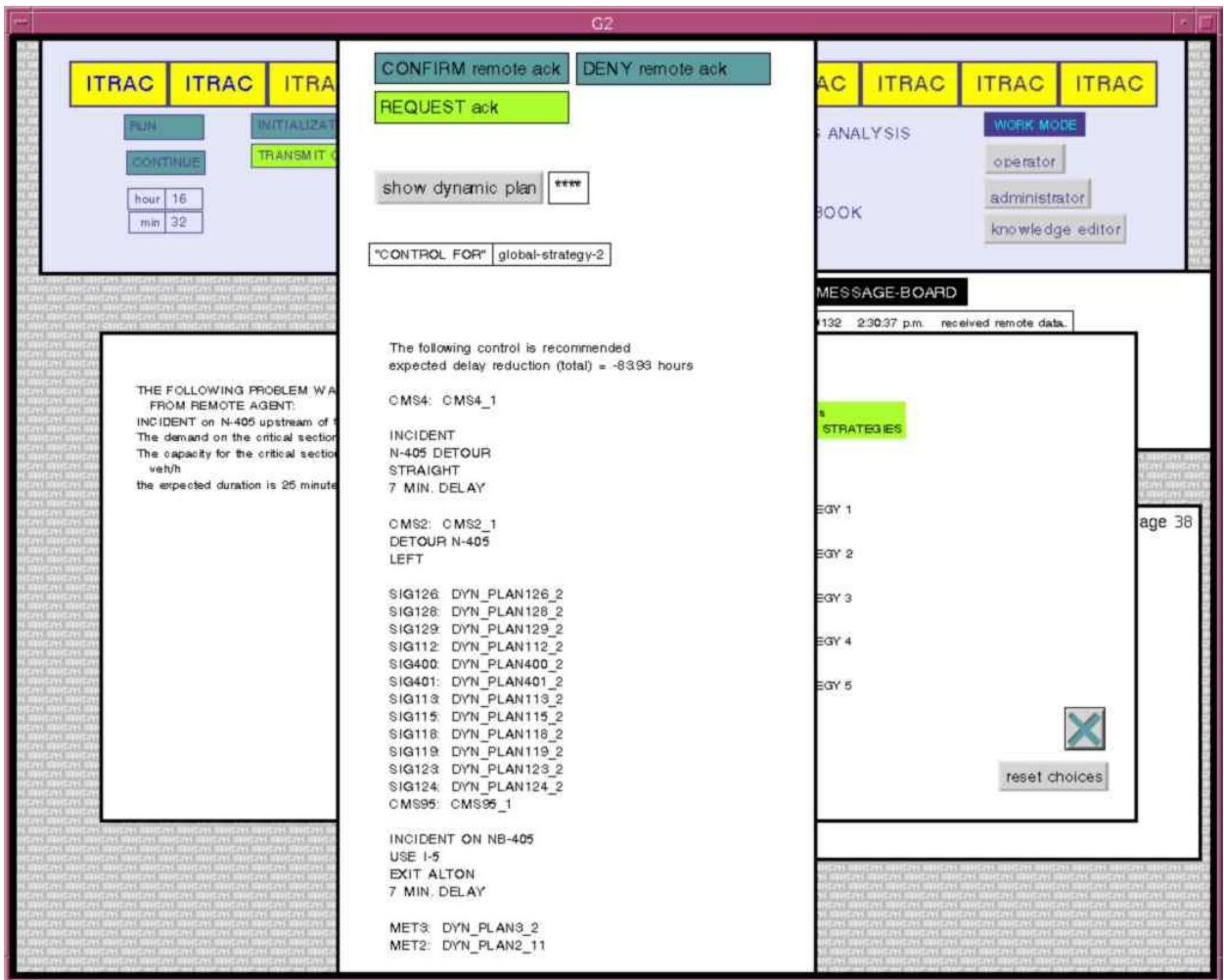
Screen 15

ITRAC: The **CARTESIUS** arterial agent receives an acknowledgement request from the D12 freeway agent and examines the D12-recommended response plan for Global Strategy 1 (GS1).



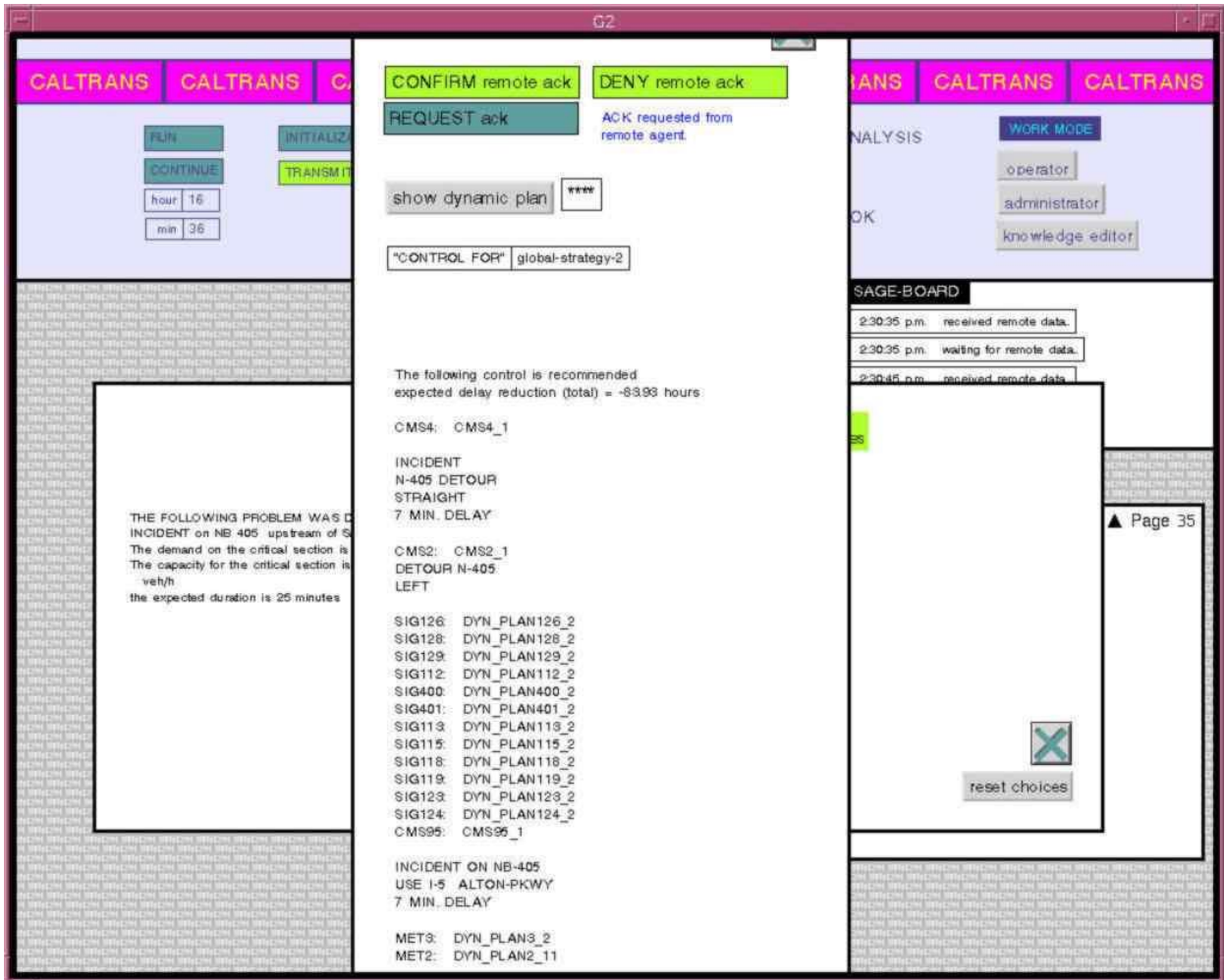
Screen 16

ITRAC: The **CARTESIUS** arterial agent rejects the request from the D12 freeway agent for Global Strategy 1 (GS1). A variety of jurisdiction specific reasons might exist, including expected changes in background demand during the response period or anticipated impacts on local streets.



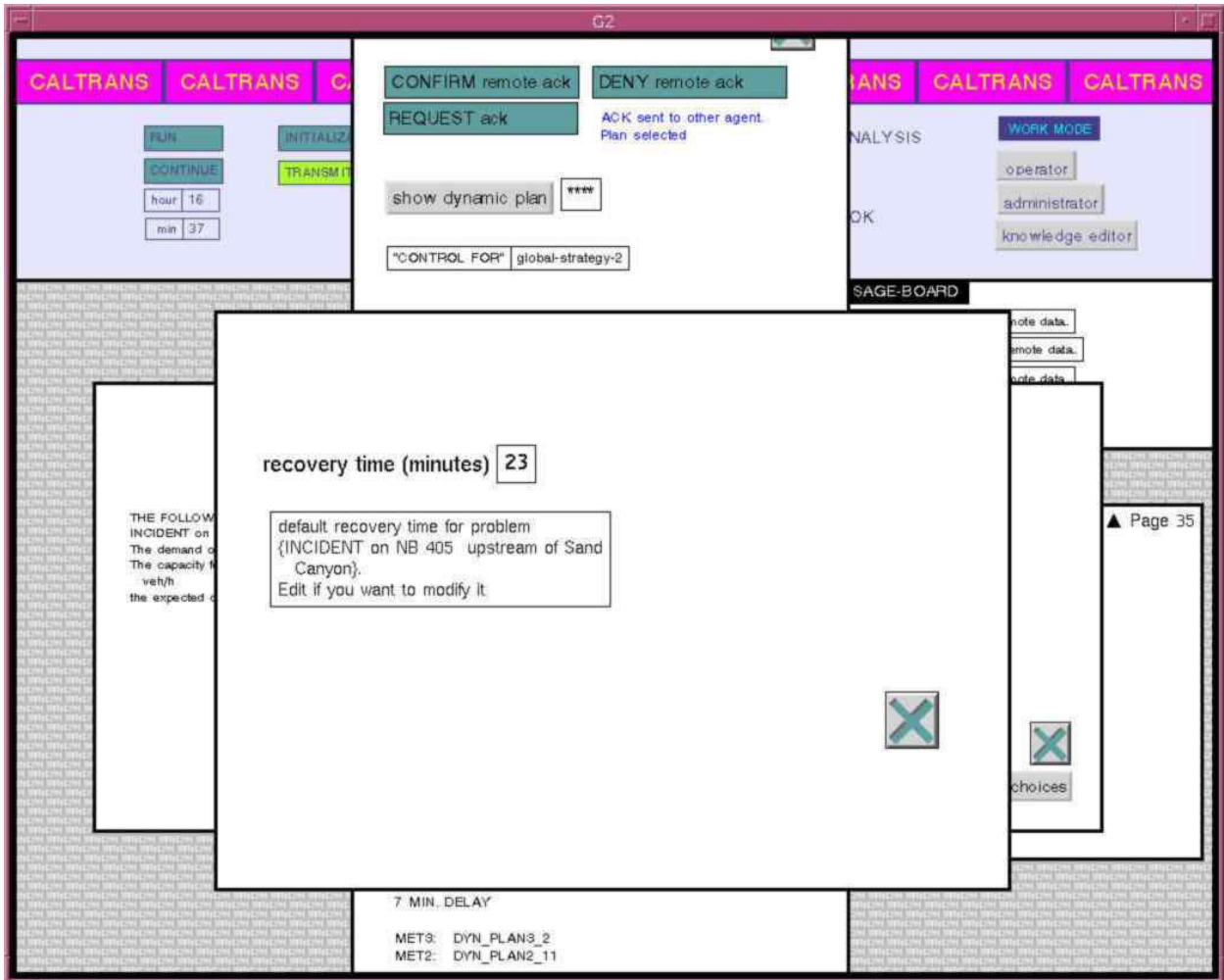
Screen 17

ITRAC: The **CARTESIUS** arterial operator reviews Global Strategy 2 (GS2) as an alternative response plan and requests acknowledgement from the freeway agent.



Screen 18

D12: The **CARTESIUS** freeway agent receives the acknowledge requests from the arterial agent with the Global Strategy 2 preference. GS2 is reviewed as a potential response plan as negotiations continue.

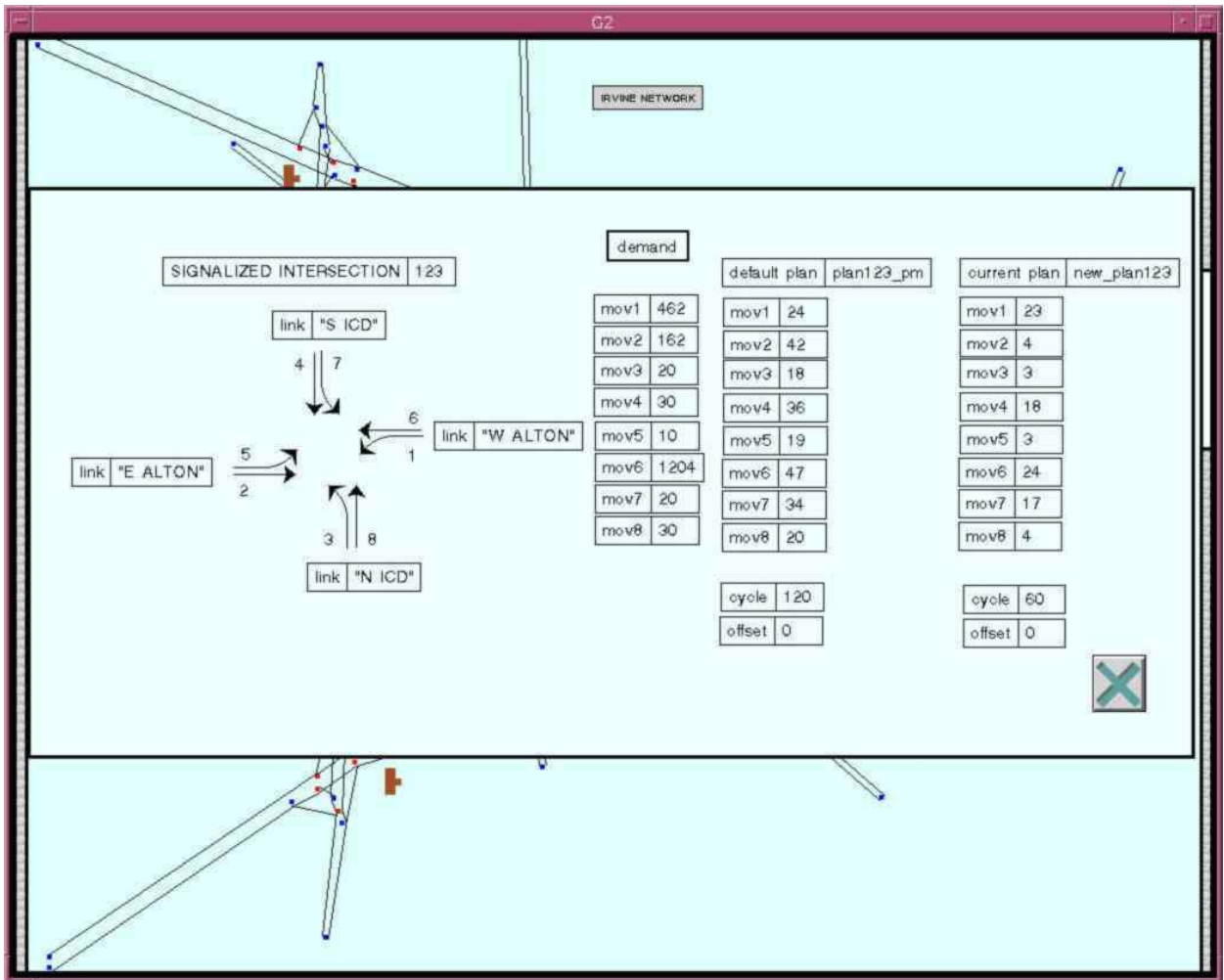


6.5 Strategy Selection and Implementation (slides 19-23)

Screen 19

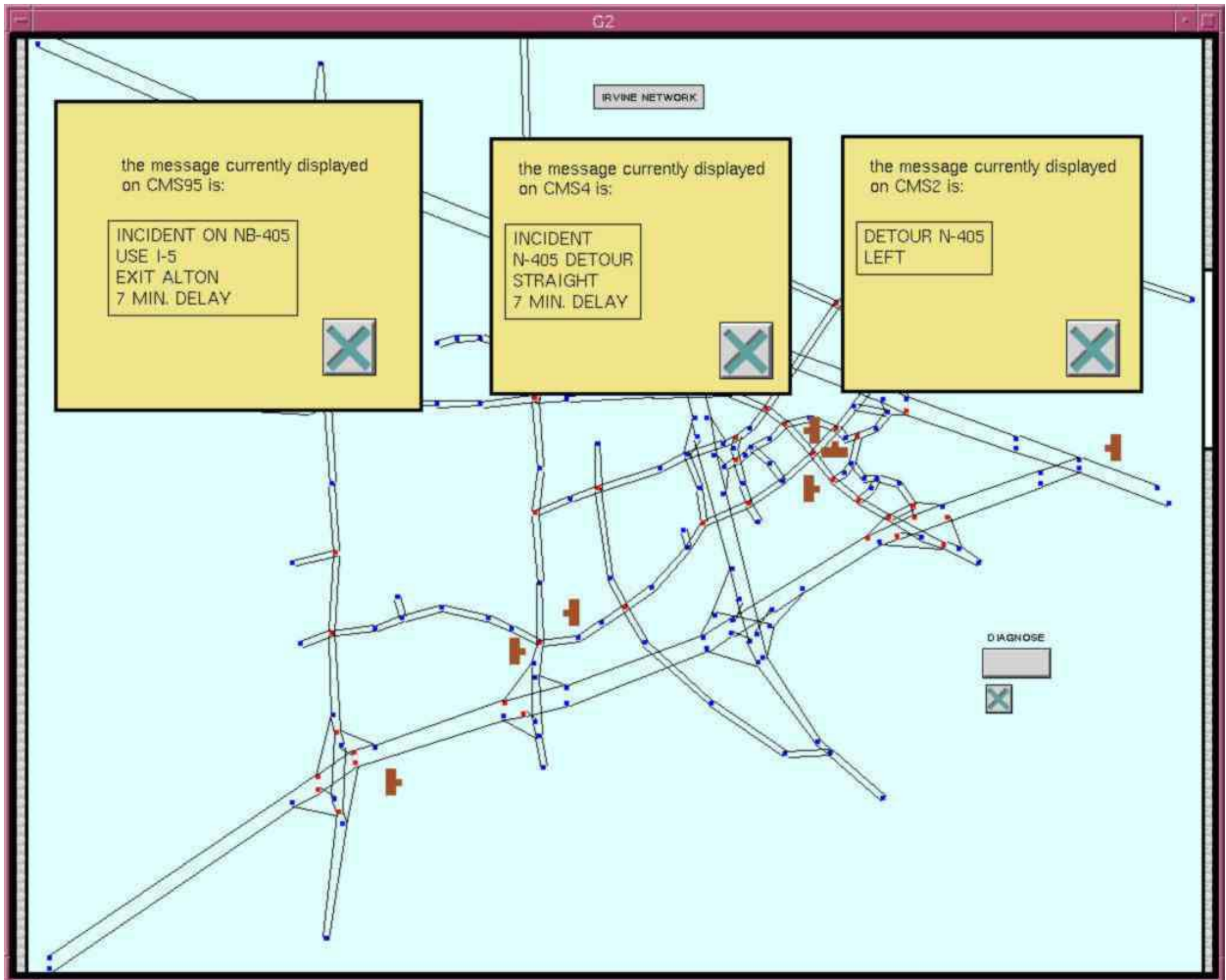
D12: The **CARTESIUS** freeway agent acknowledges the arterial agent, accepts Global Strategy 2, and implements GS2 as the response plan, ending the negotiation. The freeway agent implements freeway plan components while the arterial operator implements arterial responses. **CARTESIUS** can be set to select and implement plans if no response is received from other agents within some set response time, however, the actual plan implementation might be limited by jurisdiction policy regarding unilateral plan implementation..

The freeway agent requests the operator to provide an estimate of recovery time. This estimate will be used as an upper bound when **CARTESIUS** will again notify the operator of a (continued) problem situation. It is expected, of course, that all operators will continue to monitor the situation.



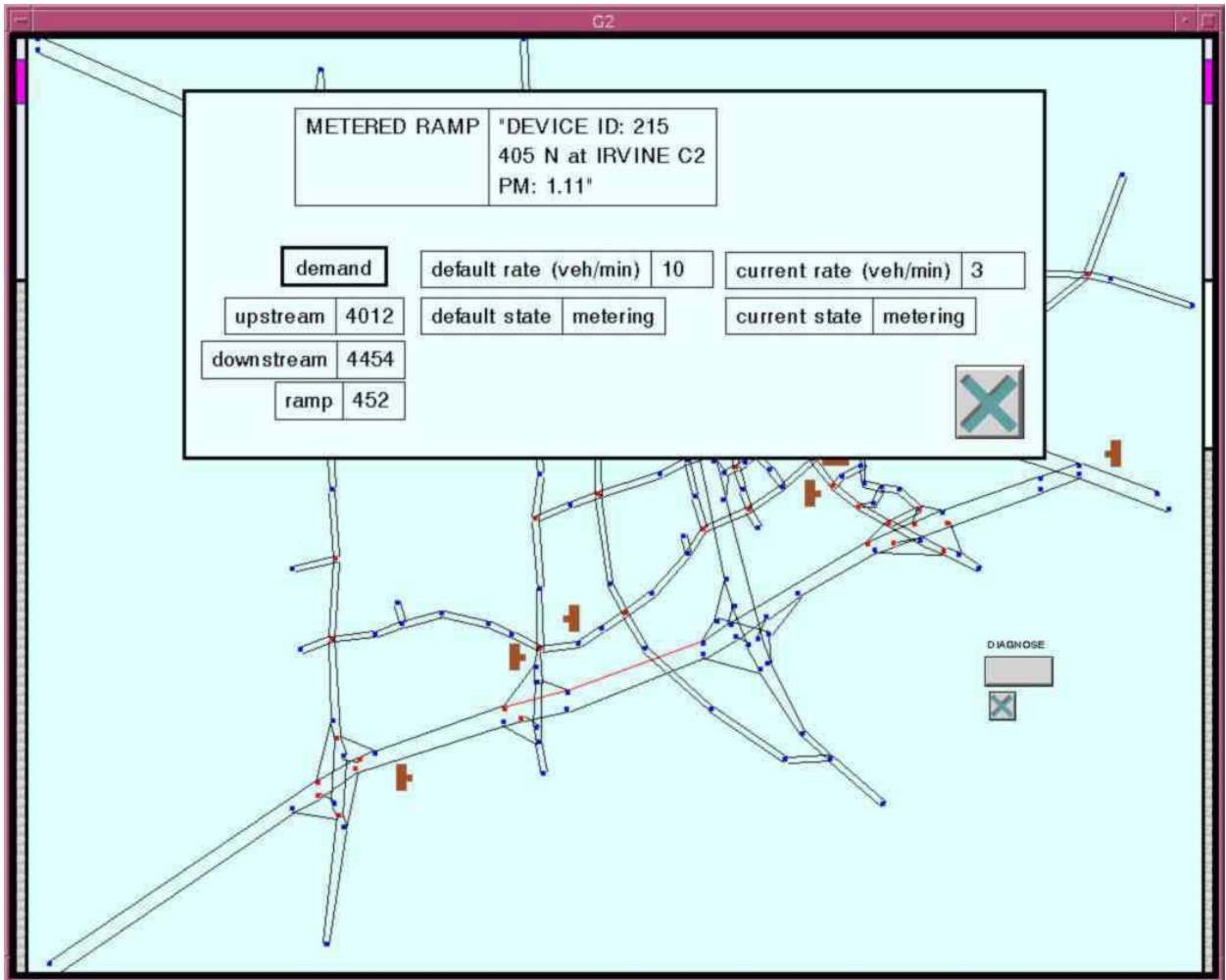
Screen 20

ITRAC: The **CARTESIUS** arterial agent reviews and implements GS2 signal plan changes (here, for the intersection of Irvine Center Drive and Alton).



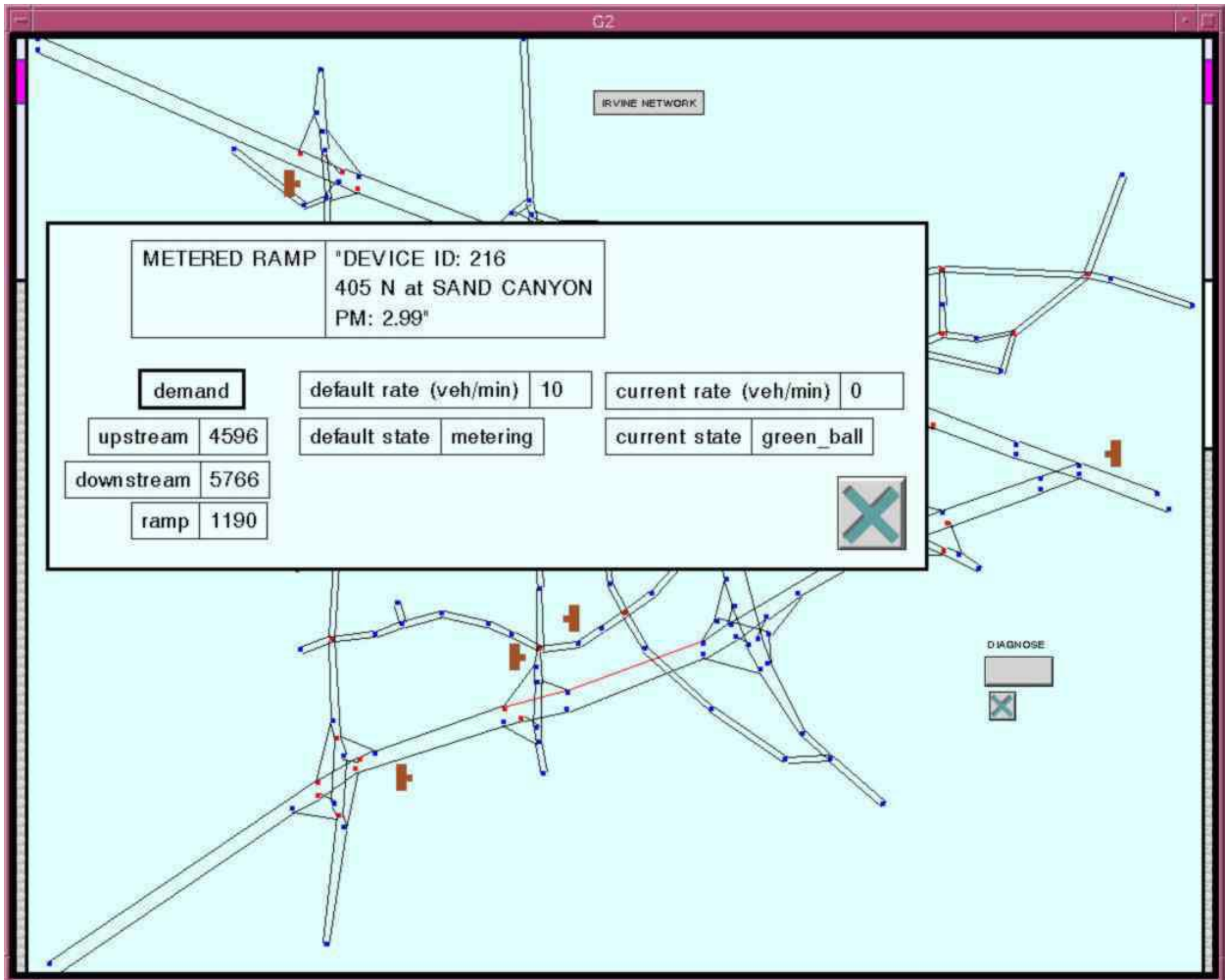
Screen 21

ITRAC: The **CARTESIUS** arterial agent reviews and implements GS2 plan changes for the CMS located on the Alton Pkwy diversion route. The GS2 plan included this diversion route to intercept northbound traffic from I-5N (coming from the lower right) heading for the I-405N, delay their exit until Alton Parkway (the next I-5N exit). The next two CMS provide arterial directions to continue on Alton then to detour back to the I-405 at Sand Canyon Boulevard, immediately north (to the left) of the incident link.



Screen 22

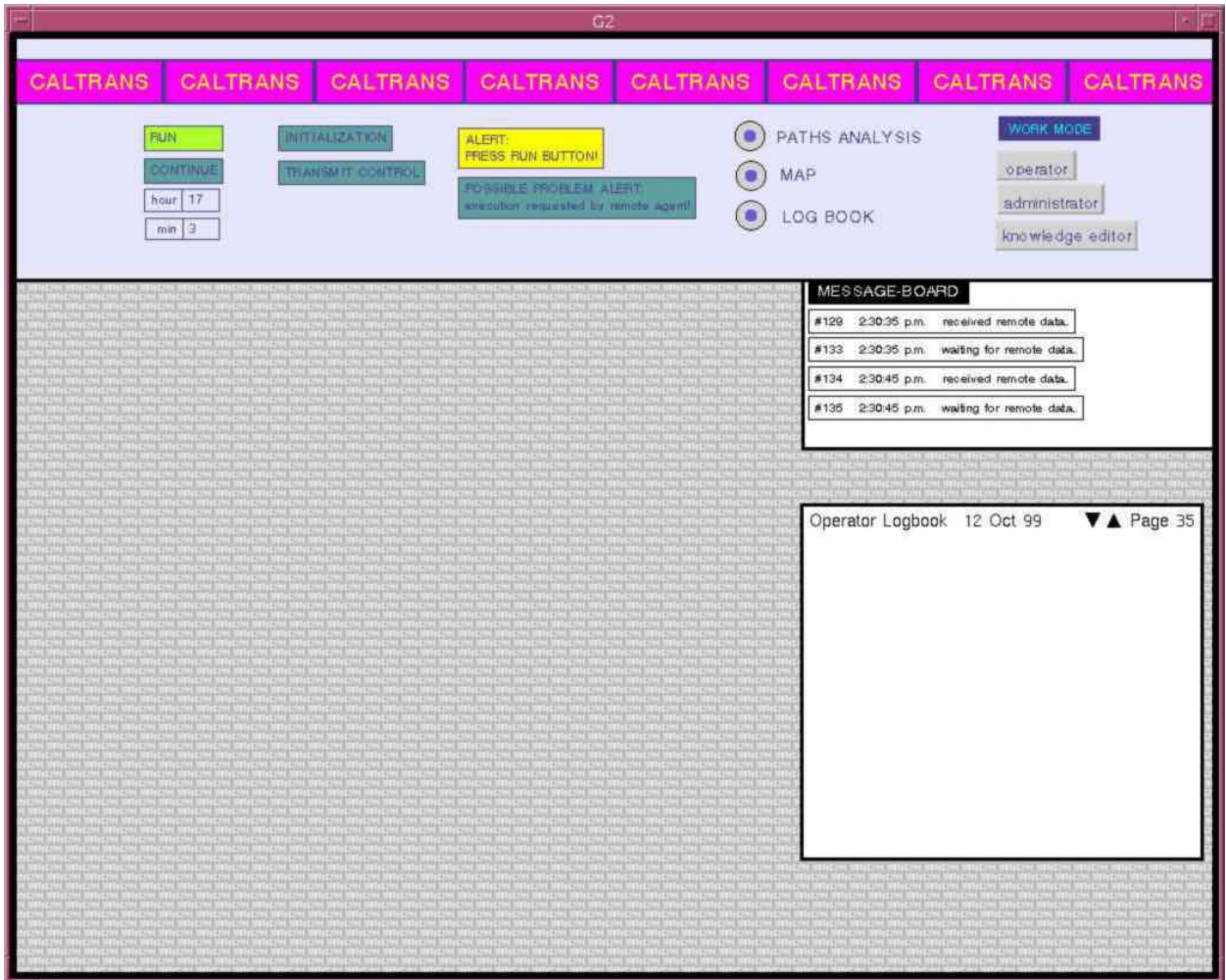
D12: The **CARTESIUS** freeway agent reviews and implements the GS2 plans for I-405 ramp meters immediately *upstream* of the identified incident (here, at Irvine Center Drive). Note the 3 veh/min rate will severely restrict traffic entering I-405 N upstream from the incident.



Screen 23

D12: The **CARTESIUS** freeway agent reviews and implements the GS2 plans for I-405 ramp meters immediately *downstream* of the identified incident (here, at Sand Canyon). Note that the meter has been set to a green ball to allow traffic from the diversion route to enter I-405 N unimpeded upstream from the incident.

At this point, **CARTESIUS** has responded by implementing the generated plans in either the real world (real-time mode), the simulated environment (simulated mode), or both (integrated mode). In this example, **CARTESIUS** continues to monitor the simulation as traffic returns to pre-incident conditions.

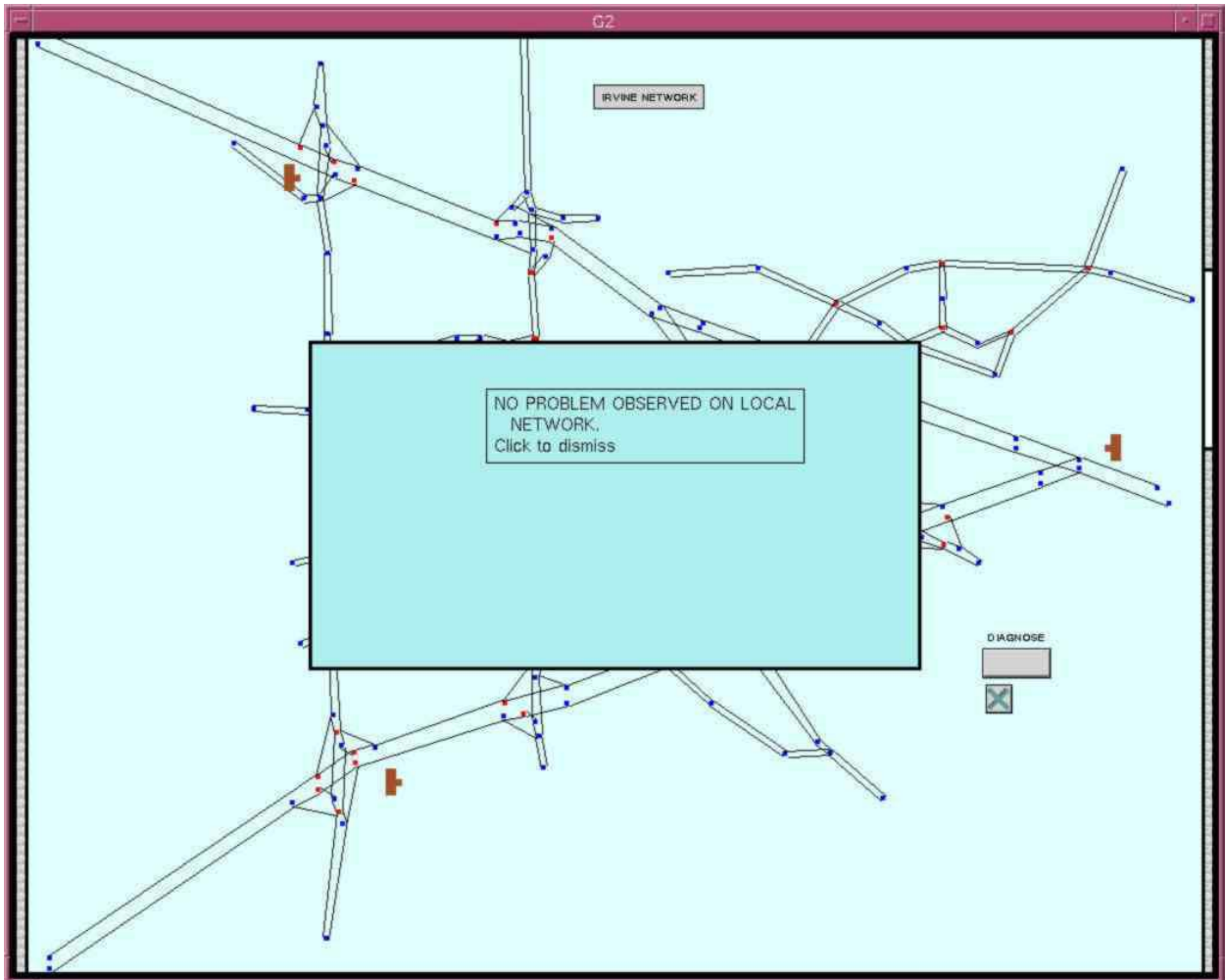


6.6 Incident Recovery and System Monitoring (slides 24-28)

Screen 24

D12: The **CARTESIUS** freeway agent receives a second alert, this being generated when the agent “detects” that there is no longer a problem. This alert will also be generated at the end of the operator-specified recovery period. Other incidents may occur in the interim, and these are addressed reflecting current conditions system-wide (including current response plan conditions).

ITRAC: The arterial agent also receives this alert (screen not shown).

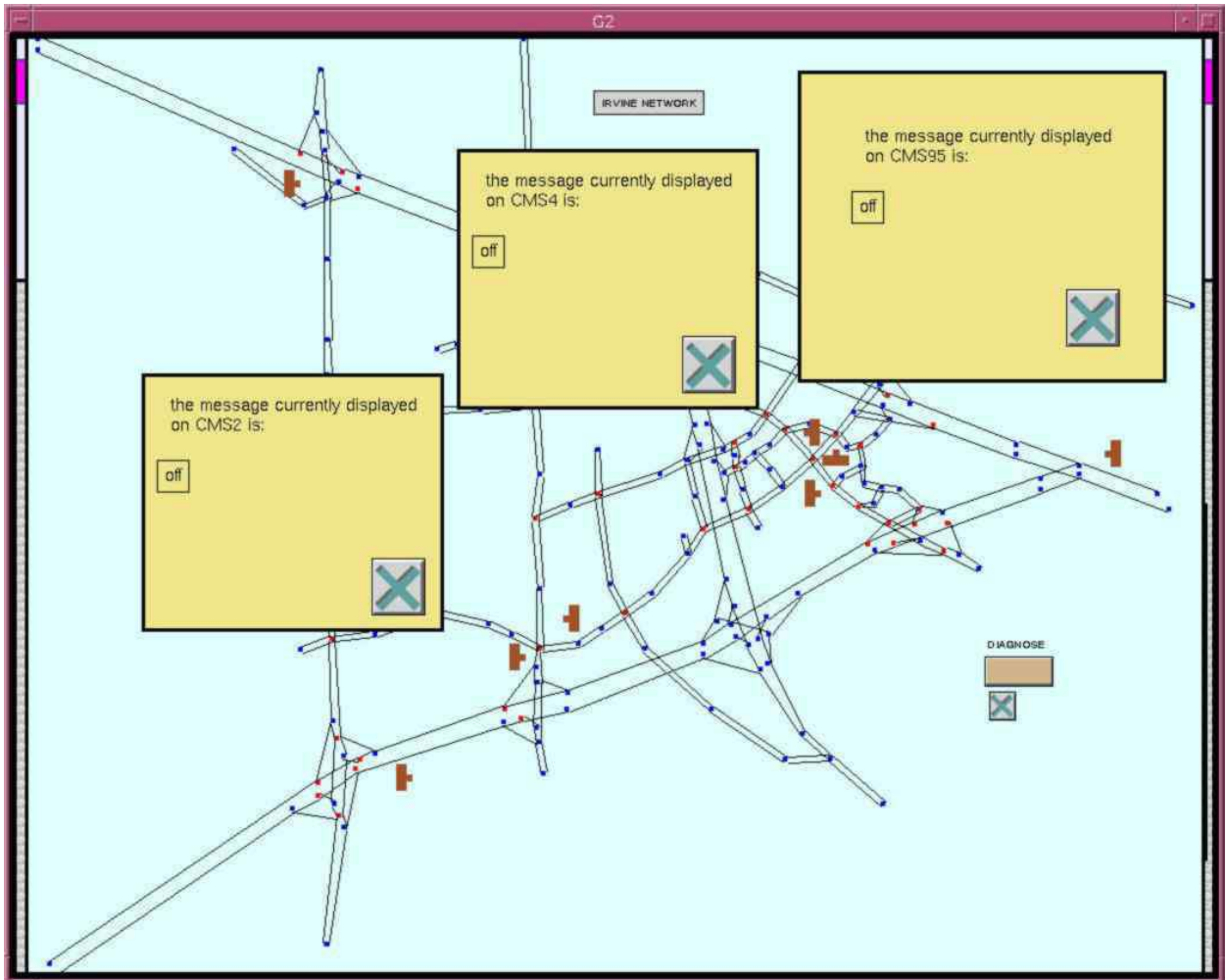


Screen 25

ITRAC: The **CARTESIUS** arterial agent after diagnosing that no problems remain on the network, thus indicating that GS2 was effective.

D12: The **CARTESIUS** freeway agent receives a similar system response.

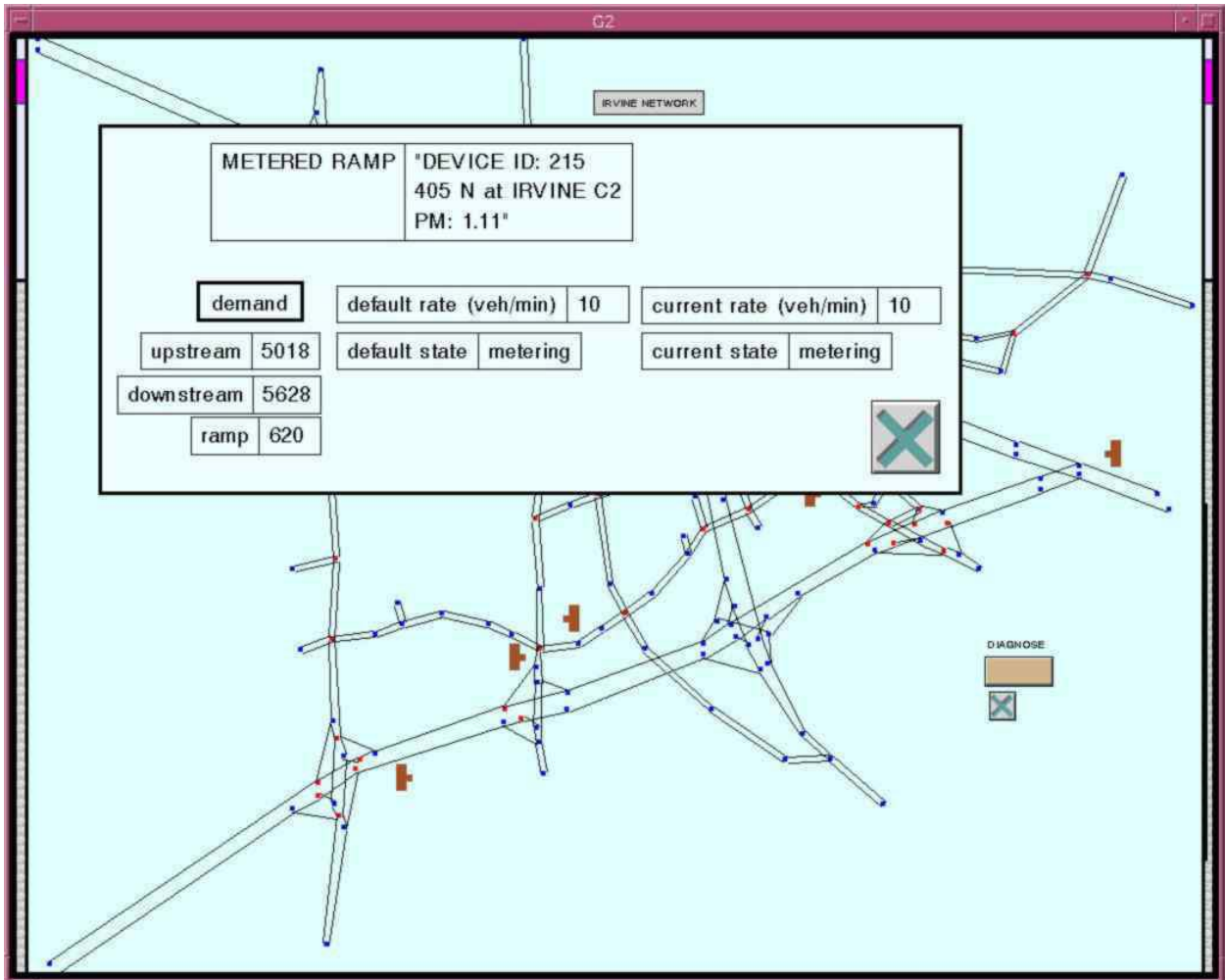
Each agent now has the option to implement the traffic management plans that were in force prior to the incident; the following screens depict some of these actions.



Screen 26

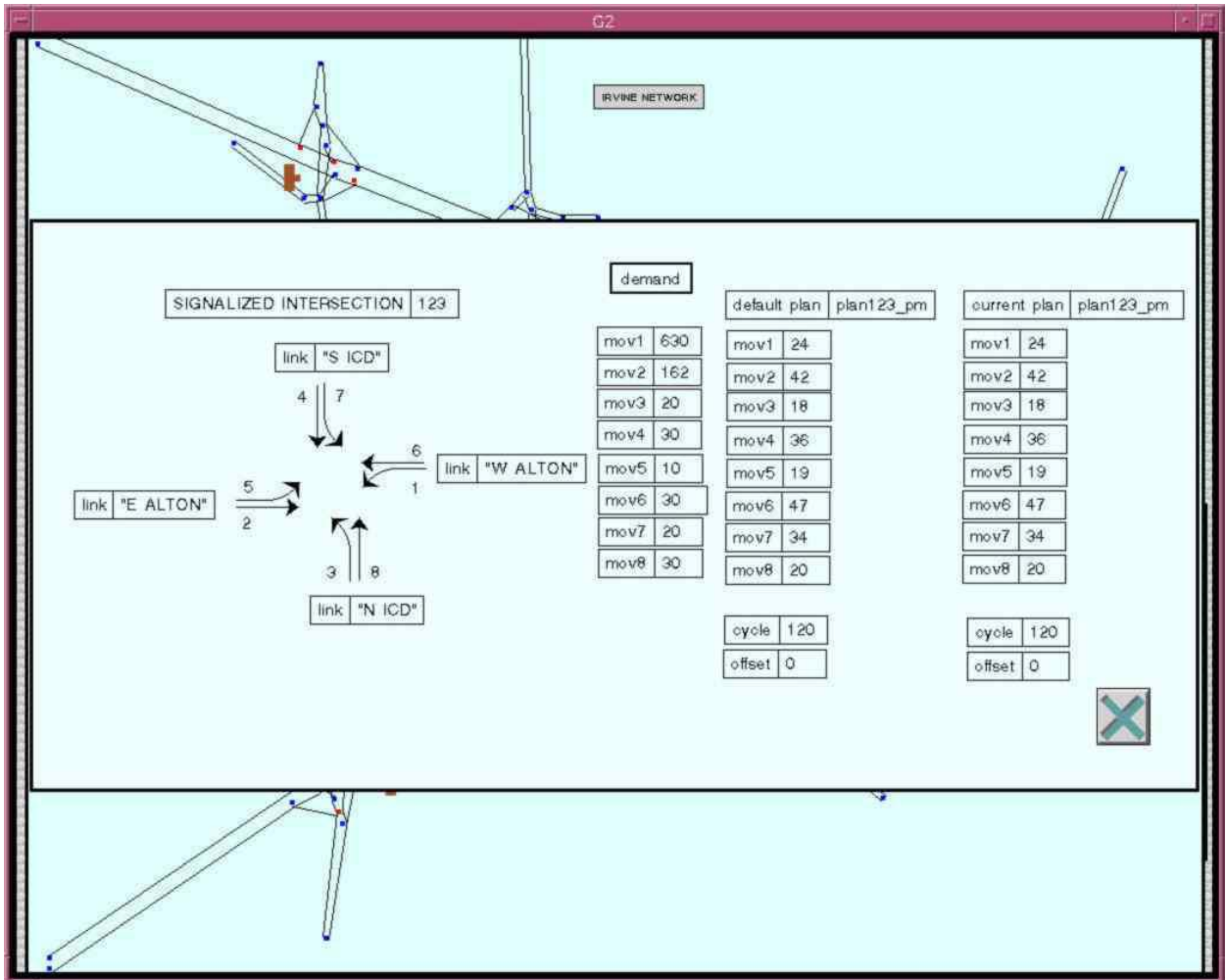
D12: The **CARTESIUS** freeway agent implements the default time-of-day plan in force prior to the incident. The CMS are reset to "off".

ITRAC: The **CARTESIUS** arterial operator takes similar action.



Screen 27

D12: The **CARTESIUS** freeway agent resets RMS plans (here, for I-405 at ICD, returning capacity to 10 veh/min).



Screen 28

ITRAC: The **CARTESIUS** arterial agent displays the prior signal plan being re-implemented at ICD and Alton (this was the default time-of-day plan).

At this point, the system has returned to the pre-incident management plan and each **CARTESIUS** agent continues to monitor the system.

Chapter 7

Summary and Future Research

7.1 Summary

This report has documented the initial testing and evaluation of the **TRICEPS** ATMS platform as part of the California ATMS Testbed project. The evaluation process involved the development of a **TRICEPS** ATMS application to assess the validity of a new distributed methodology for the provision of integrated ATMS strategies in response to congestion. Such methodology employs coordination mechanisms that support cooperation and conflict resolution between two distinct automatic problem-solving agents, within the distributed **CARTESIUS** system. The agents in the system have access to separate databases and data sources, and may use different control algorithms, thus reflecting the inherent administrative distribution of data and expertise among separate management agencies. The cornerstone of this cooperative approach is the assumption that effective *integrated* traffic control solutions can be obtained in *real-time* by relaxing the requirement that agents have shared access to all globally available information. The simulation-based validation of the system performance demonstrated the effectiveness of **CARTESIUS** in producing real-time, integrated control solutions that reduce the adverse impact of incidents on traffic circulation, network-wide.

7.2. Future Research

The most critical research and implementation need is to field test **TRICEPS** and **CARTESIUS** in the study area. This requires formal completion of the ITRAC data feed (currently underway as part of the Testbed project) and formalization of a series of field tests, first under simulated (i.e., staged) conditions (such as routine maintenance operations), then under real incident condition scenarios (with **TRICEPS** and **CARTESIUS** operating in partial integrated mode – analysis but no active field control, followed by extensive evaluation).

The decision modules utilized by **CARTESIUS** (e.g., incident detection, diversion strategies, optimal control), which currently are specific to the agencies and network associated with Caltrans District 12 and the City of Irvine research implementation, could be replaced by more general modules arising from the development of **TRICEPS** forming the foundation for a real-time, non-proprietary, adaptive ATMS. Area-specific knowledge would be input to **TRICEPS** for local applications. More critical would be the actual replacement of control algorithms in **CARTESIUS** with state-of-the-art algorithms residing in the **TRICEPS** workbench separate from **CARTESIUS**. Finally, extensions to three or more agents would illustrate the general applicability of this approach.

References

Akcelik,R (1981). *Traffic signals: capacity and timing analysis*. Research Report 123, Australian Road Research Board, Victoria, Australia.

Akcelik, R (1994). Estimation of green times and cycle times for vehicle-actuated signals, *Transportation Research Record* 1457, 63-72.

Carver,N, Cvetanovic,Z,. and Lesser,V (1991). Sophisticated cooperation in FA/C distributed problem-solving systems, *Proceedings of the National Conference on Artificial Intelligence*, 191-198.

Carver,N. and Lesser,V (1995). *A first step toward the formal analysis of solution quality in FA/C distributed interpretation systems*, Technical Report 94-37, Department of Computer Science, University of Massachusetts.

Gazis,DC (1974). *Traffic control - theory and application*. Wiley and Sons, New York.

Duncan,GI (1995) PARAMICS wide area microscopic simulation of ATT and traffic management, *Proceedings of the 28th International Symposium on Automotive Technology and Automation (ISATA)*, Stuttgart, Germany, 475-484.

Gensym, Inc. (1995). *G2: Reference Manual. Version 4.0*, Gensym Corporation, Cambridge, MA.

Jayakrishnan,R., Mahmassani,H and Hu,T-Y (1994). An evaluation tool for advanced traffic information and management systems in urban networks, *Transportation Research C*, 2(3), 129-147.

Jayakrishnan,R and Rindt, CR (1999). Distributed Computing and Simulation in a Traffic Research Testbed. *Computer-aided Civil and Infrastructure Engineering*, Vol. 14, No. 6, 1999, pp. 429-443.

JHK & Associates (1993). *Irvine Transportation Analysis Model*, City of Irvine, Irvine, CA.

Kulkarni,A and McNally,MG (1997). *Testbed O/D Demand Estimation*. Testbed Research Report. Institute of Transportation Studies, University of California, Irvine.

Leonard,DR, Gower,P, and Taylor,NB (1989). *CONTRAM, structure of the model*, TRRL Research Report, Transport and Road Research Laboratory, Crowthorne, England.

Lesser,VR (1991). *A retrospective view of FA/C distributed problem solving*. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 6, pp. 1347-1362.

Lesser,VR and Corkill,DD (1981). Functionally Accurate, Cooperative distributed systems, *IEEE Transactions on Systems, Man, and Cybernetics* SMC-11(1): 81--96.

Logi,F (1995). A software architecture for the integration of Advanced Transportation Management and Information Systems, Masters Thesis, Department of Civil and Environmental Engineering, University of California, Irvine.

Logi,F (1999). *CARTESIUS: a cooperative approach to real-time decision support for multi-jurisdictional traffic congestion management*. Ph.D. Dissertation, Department of Civil and Environmental Engineering., University of California.

Logi,F and Ritchie, SG (1997).. A Knowledge-Based System for Integrated Freeway and Arterial Incident Management and Control: Implementation and Validation Results. *Eighth IFAC Symposium on Transportation Systems*. Chania, Greece,.

Logi,F and Ritchie,SG (1998). A distributed approach to network-wide traffic control management. In Hendrickson, C. T. and S. G. Ritchie (eds.) *Proceedings of the 5th International Conference on Applications of Advanced Technologies in Transportation*. Newport Beach, CA,

McNally,MG, Rindt,C, and Logi,F (1999). TRICEPS: An ATIS Field Implementation for Control and Evaluation. California PATH Report 99-C2 for MOU 281.

Mahmassani, H., T-Y Hu, S. Peeta, and A. Ziliaskopoulos (1994) Development and Testing of Dynamic Traffic Assignment and Simulation Procedures for ATIS/ATMS Applications, Report DTFH61-90-R-00074-FG, U.S. DOT, Federal Highway Administration, McLean, Virginia.

Peeta S. and H. Mahmassani (1995) Multiple user classes real-time traffic assignment for online operations: rolling horizon solution framework, *Transportation research*. 3C, 83-98

Recker,WW (1997). *California Advanced Transportation Management Systems Testbed - Operational Phase*, Final Report prepared for the California Department of Transportation, Institute of Transportation Studies, University of California, Irvine, CA.

Rindt,CR, Jayakrishnan,R, and McNally,MG (1997). Implementation of a Real-time Integrated Control System in a Freeway/Arterial Corridor. *In* M. Papageorgiou & A. Pouliezios (eds.), *Transportation Systems 1997. (TS'97). Proceedings volume from the 8th IFAC Symposium*, Vol.3, Pergamon, Oxford, UK, pp.1097-1120.

UAG (1995). *Urban/Sys Tranplan User Manual*, Version 8.0, Urban Analysis Group, Danville, CA.

Quadstone Ltd. (1999) Paramics traffic simulation Modeller V3.0 Reference Manual.