STATE OF CALIFORNIA DEPARTMENT OF TRANSPORTATION

TECHNICAL REPORT DOCUMENTATION PAGE

TR0003 (REV. 10/98)

1. REPORT NUMBER	2. GOVERNMENT ASSOCIATION NUMBER	3. RECIPIENT'S CATALOG NUMBER
CA12-1777		
4. TITLE AND SUBTITLE AHMCT Intelligent Roadway Information System (IRIS) Technical Support and Testing		5. REPORT DATE December 31, 2011 6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) Michael Darter, Travis Swanston, Kin Yen, Bahram Ravani & Ty Lasky		8. PERFORMING ORGANIZATION REPORT NO. UCD-ARR-11-12-31-01
9. PERFORMING ORGANIZATION NAME AND ADDRESS AHMCT Research Center UCD Dept of Mechanical & Aerospace Engineering Davis, California 95616-5294		10. WORK UNIT NUMBER 11. CONTRACT OR GRANT NUMBER 65A0275, Task ID 1777
12. SPONSORING AGENCY AND ADDRESS California Department of Transportation Division of Research and Innovation, MS-83 1227 O Street Sacramento CA 95814		13. TYPE OF REPORT AND PERIOD COVERED Final Report November 2009 – December 2011 14. SPONSORING AGENCY CODE
15. SUPPLEMENTAL NOTES		

16. ABSTRACT

This report documents the research project "AHMCT IRIS Technical Support and Testing," performed under contract 65A0275, Task ID 1777. It presents an overview of the Intelligent Roadway Information System (IRIS), and its design and function. IRIS has provided ATMS capabilities to Caltrans' rural districts. IRIS is currently deployed in District 10, and near deployment in Districts 1, 2, and 5.

17. KEY WORDS	18. DISTRIBUTION STATEMENT	
ATMS, Traffic operations, Open source	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161	
19. SECURITY CLASSIFICATION (of this report)	20. NUMBER OF PAGES	21. PRICE
Unclassified	47	N/A

AHMCT Intelligent Roadway Information System (IRIS) Technical Support and Testing

Final Report

Federal Report Number: CA12-1777

December 31, 2011

Prepared for:

California Department of Transportation

Division of Research and Innovation

Advanced Highway Maintenance and Construction Technology Research Center

Department of Mechanical and Aerospace Engineering
University of California at Davis

AHMCT Intelligent Roadway Information System (IRIS) Technical Support and Testing

Michael Darter, Travis Swanston, Kin Yen, Bahram Ravani &
Ty A. Lasky: Principal Investigator

AHMCT Research Report: UCD-ARR-11-12-31-01
Final Report of Contract: IA65A0275, Task ID 1777

December 31, 2011

California Department of Transportation

Division of Research and Innovation

ABSTRACT

This report documents the research project "AHMCT IRIS Technical Support and Testing," performed under contract 65A0275, Task ID 1777. It presents an overview of the Intelligent Roadway Information System (IRIS), and its design and function. IRIS has provided ATMS capabilities to Caltrans' rural districts. IRIS is currently deployed in District 10, and near deployment in Districts 1, 2, and 5.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DISCLAIMER/DISCLOSURE

This document is disseminated in the interest of information exchange. 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 or the FEDERAL HIGHWAY ADMINISTRATION or the UNIVERSITY OF CALIFORNIA. This report does not constitute a standard, specification, or regulation. This report does not constitute an endorsement by the Department of any product described herein. The United States Government does not endorse products or manufacturers. Trade and manufacturers' names appear in this report only because they are considered essential to the object of the document. For individuals with sensory disabilities, this document is available in Braille, large print, audiocassette, or compact disk. To obtain a copy of this document in one of these alternate formats, please contact: the Division of Research and Innovation, MS-83, California Department of Transportation, P.O. Box 942873, Sacramento, CA 94273-0001.

The research reported herein was performed as part of the Advanced Highway Maintenance and Construction Technology (AHMCT) Research Center, within the Department of Mechanical and Aerospace Engineering at the University of California, Davis and the Division of Research and Innovation at the California Department of Transportation. It is evolutionary and voluntary. It is a cooperative venture of local, state and federal governments and universities.

LIST OF ACRONYMS AND ABBREVIATIONS

Acronym	Definition
AHMCT	Advanced Highway Maintenance and Construction Technology
API	Application Programming Interface
ATMS	Advanced Transportation Management System
AWS	Automated Warning System
Caltrans	California State Department of Transportation
CAWS	Caltrans Automated Warning System
CCB	Change Control Board
CCTV	Closed-Circuit TV
CMS	Changeable Message Sign
COCOMO	COnstructive COst MOdel
COTS	Commercial-Off-the-Shelf
DMS	Dynamic Message Sign
DRI	Division of Research and Innovation
FSR	Feasibility Study Report
GB	Gigabyte
GPL	General Public License
HAR	Highway Advisory Radio
IRIS	Intelligent Roadway Information System
LDAP	Lightweight Directory Access Protocol
MB	Megabyte
Mn/DOT	Minnesota Department of Transportation
MVDS	Microwave Vehicle Detection Station
NDA	Non-Disclosure Agreement
OS	Operating System
OSM	OpenStreetMap
PeMS	Performance Measurement System
RAM	Random Access Memory
RTMS	Remote Traffic Microwave Sensor
RWIS	Road Weather Information System
TMC	Transportation Management Center
TMS	Traffic Monitoring Station
UCD	University of California-Davis
URMS	Universal Ramp Metering System
VDS	Vehicle Detection Station
XML	eXtensible Markup Language

ACKNOWLEDGMENTS

The authors gratefully acknowledge the Division of Research and Innovation (DRI) of Caltrans which has supported this work through the AHMCT Research Center at the University of California-Davis, under contract 65A0275 Task ID 1777, and thank Roya Hassas and Fred Yazdan in particular. The authors also thank Stan Slavin for his commitment and strong guidance. The authors thank Doug Lau of Mn/DOT for his leading work on IRIS and his support for collaborative development. The authors thank Burwell Briggs for his work on the FSR. In addition, the authors thank Jagtar Dhaliwal, Alan Benson, Chad Riding, Jane Berner, and Mike Jenkinson. From District 1, the authors thank Joe Dower. From District 2, the authors thank Ian Turnbull and Clint Burkenpas. From District 5, the authors thank Sherwyn Gilliland and Steven Gee. From District 10, the authors thank Mohammad Battah, Veronica Cipponeri, Toni Moon, John Castro and Wilmar Kuhl. Finally, the authors thank the members of the Change Control Board and the remaining dedicated managers and operators in Districts 1, 2, 5, and 10.

CHAPTER 1:
INTRODUCTION

Background

In Phase I research and development [1], AHMCT tailored and extended the open-source Intelligent Roadway Information System (IRIS) originally (and currently) developed by the Minnesota Department of Transportation (Mn/DOT). This research was successfully demonstrated and deployed in District 10, Stockton as a Proof of Concept project.

As part of the current Phase II effort, AHMCT next provided low-level deployment support (maintenance, software patches, and technical support) for IRIS in District 10. Additional districts (District 1-Eureka, District 2-Shasta, and District 5-San Luis Obispo) began testing a subset of IRIS functionality (Changeable Message Sign (CMS) control) to provide feedback for the Caltrans Feasibility Study Report (FSR) process. Full IRIS capability, including video, for the added districts requires device driver development, beyond the resources available before the Phase II research; in addition, until the FSR process was complete, these added districts remained in the testing mode for CMS.

AHMCT IRIS testing and technical support was provided under this Phase II effort. The Phase II effort also included providing technical information in support of Caltrans' preparation and execution of the IRIS FSR.

Urban Advanced Transportation Management System (ATMS) programs are typically not well-suited to rural districts. As such, rural districts often address their needs with a set of disparate solutions, with associated management, administration, and operating difficulties. There is a need for a unified ATMS that is specifically designed for rural districts.

Rural districts cannot easily justify the one-time and recurring costs of an urban ATMS. A low-cost alternative would provide significant operational capabilities to the districts, with substantial savings to the Department.

Research Approach

The research tasks were as follows:

- IRIS Technical Support for District 10 Testing
- IRIS Technical Support for District 1 Testing
- IRIS Technical Support for District 5 Testing
- Software engineering process for IRIS ticket requests from Districts 1, 5, 10, subject to Change Control Board (CCB) guidance
- Technical information input in support of Caltrans' preparation of the IRIS Feasibility Study Report (FSR)

The research tasks included support for District 2 testing, and Automated Warning System (AWS) development.

Overview of Research Results

The district "test drives" were successful, based on feedback from the CCB.

District 10 has converted over to IRIS, i.e. it is fully deployed in District 10.

The number of software applications and servers in District 10 has been reduced, with IRIS assuming more roles.

Of most significance for District 10, IRIS took over the Caltrans Automated Warning System (CAWS). IRIS also took over the role of District 10's legacy middleware system which was responsible for acquiring all of the vehicle detection and weather information from the field devices, as well as reporting the vehicle detection information to the Performance Measurement System (PeMS).

AHMCT provided support for the Feasibility Study and Report. The FSR was approved and fully executed in April 2011, authorizing IRIS deployment to Districts 1, 2, and 5.

At this stage, IRIS is a platform that:

- Provides ATMS capabilities for rural districts
- Is extensible, reliable, and scalable
- Reduced life cycle costs 72% vs. the existing ATMS
- Shows the importance of an open development process

CHAPTER 2:
IRIS BACKGROUND

An Advanced Transportation Management System (ATMS) is a software tool that provides Transportation Management Center (TMC) operators and Traffic Managers with a real-time view of highway conditions so that accurate and timely actions can be performed in response to adverse environments or traffic incidents (Figure 2.1). It also provides operators with direct access and control to multiple types of roadway devices rather than having to operate multiple systems. An ATMS allows Caltrans to:

1. Effectively manage the freeways
2. Reduce traveler commuting times
3. Maximize roadway capacity
4. Provide a safer traveling medium for the general public.

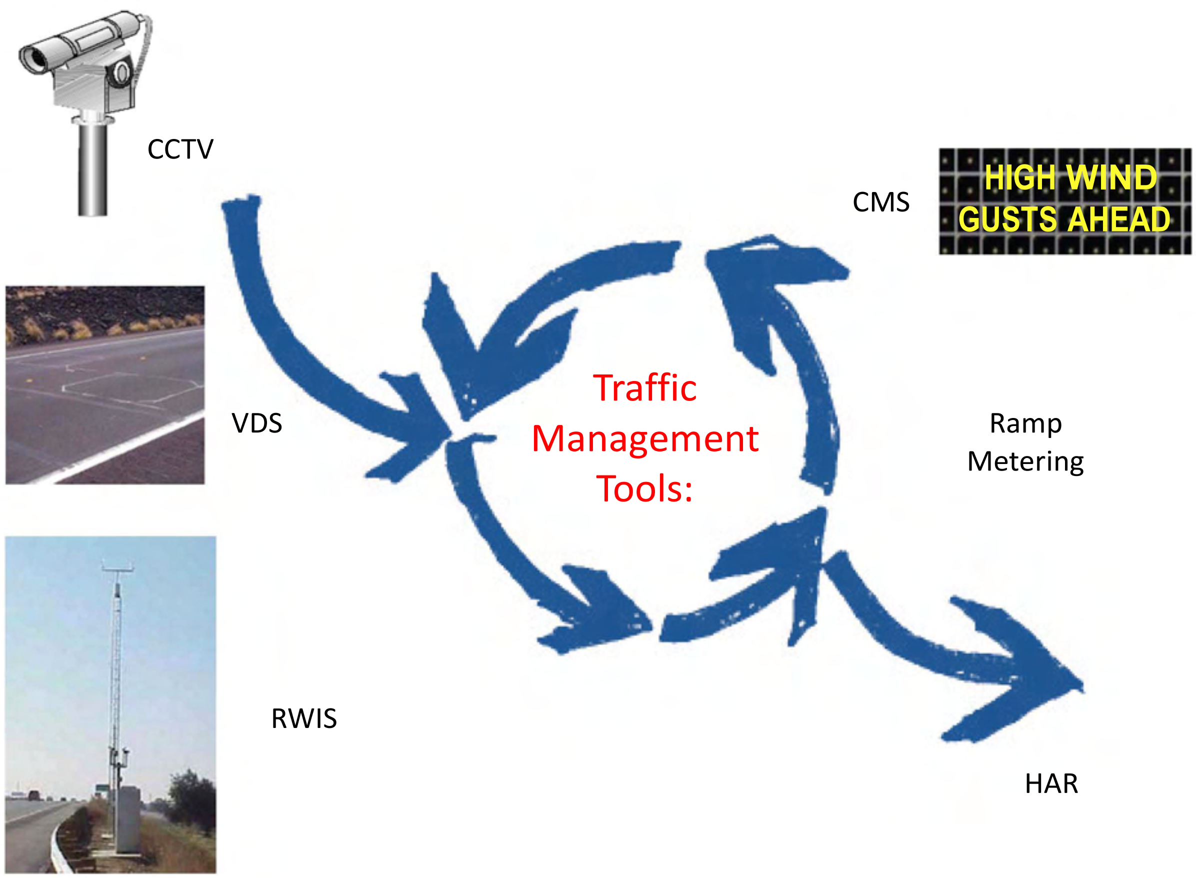

Figure 2.1: ATMS real-time functions

IRIS was developed by the Minnesota Department of Transportation (Mn/DOT), with development started in the early 1990s. It is used in Minneapolis/St. Paul, St. Cloud, and Rochester. In terms of infrastructure, Mn/DOT uses IRIS to manage approximately 135 DMS, 476 cameras, 5452 VDS, 433 ramp meters, 4 RWIS, 194 LCS, 1 Lane Marking (in-road lighting), and 2 static signs with wig-wag beacons. IRIS was 100% developed in-house, and represents a significant investment (COCOMO[1]: > $4 million)

MN/DOT released IRIS as open-source software under the General Public License (GPL) in May 2007. At least four agencies are using or evaluating IRIS. MN/DOT's motivations for open-sourcing included:

- Insuring affordable and manageable longevity of IRIS
- Fostering collaboration with other transportation agencies
- Getting source contributions back to IRIS
- Cultivating additional IRIS developers
- And lowering risks

4+ agencies using or evaluating

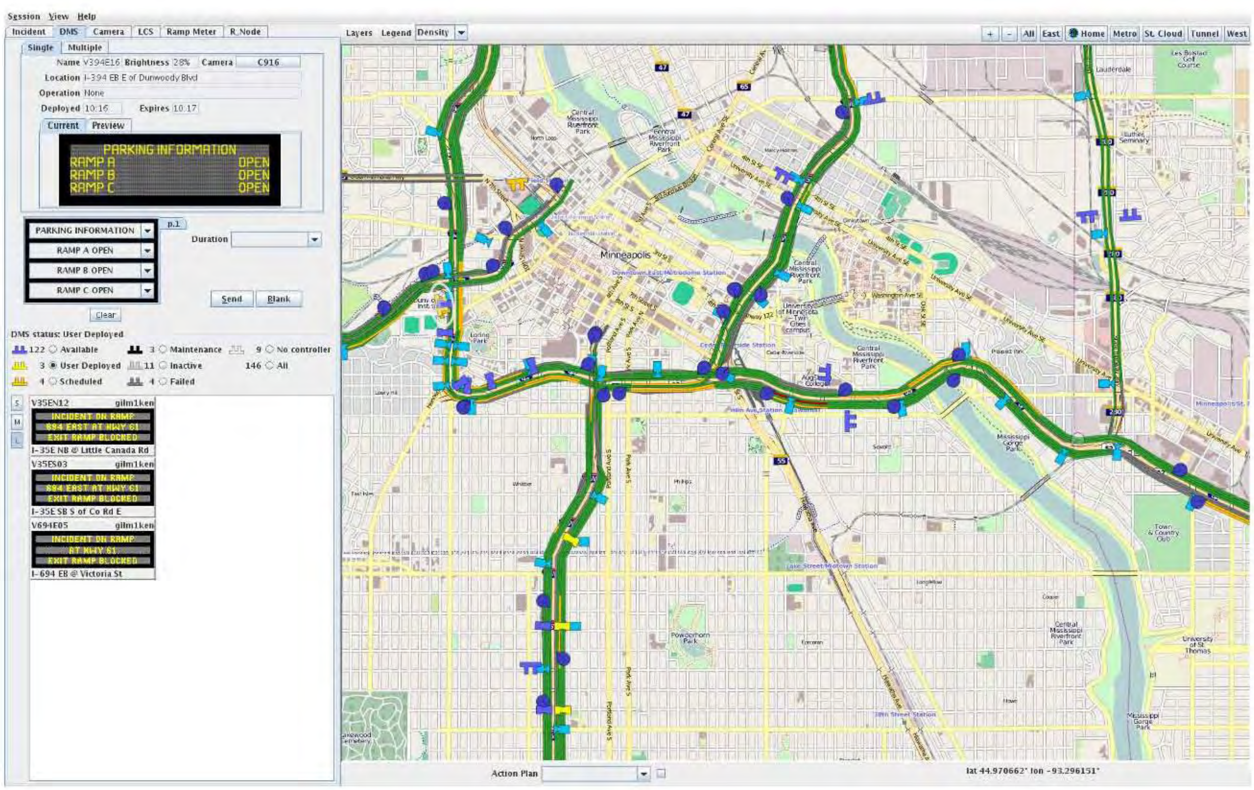

Figure 2.2: Minnesota IRIS DMS functionality

[1] COnstructive COst MOdel

IRIS has provided ATMS capabilities to Caltrans' rural districts. It is currently deployed in District 10, and near deployment in Districts 1, 2, and 5. A sample IRIS Dynamic Message Sign (DMS) control operating in Minnesota is shown in Figure 2.2.

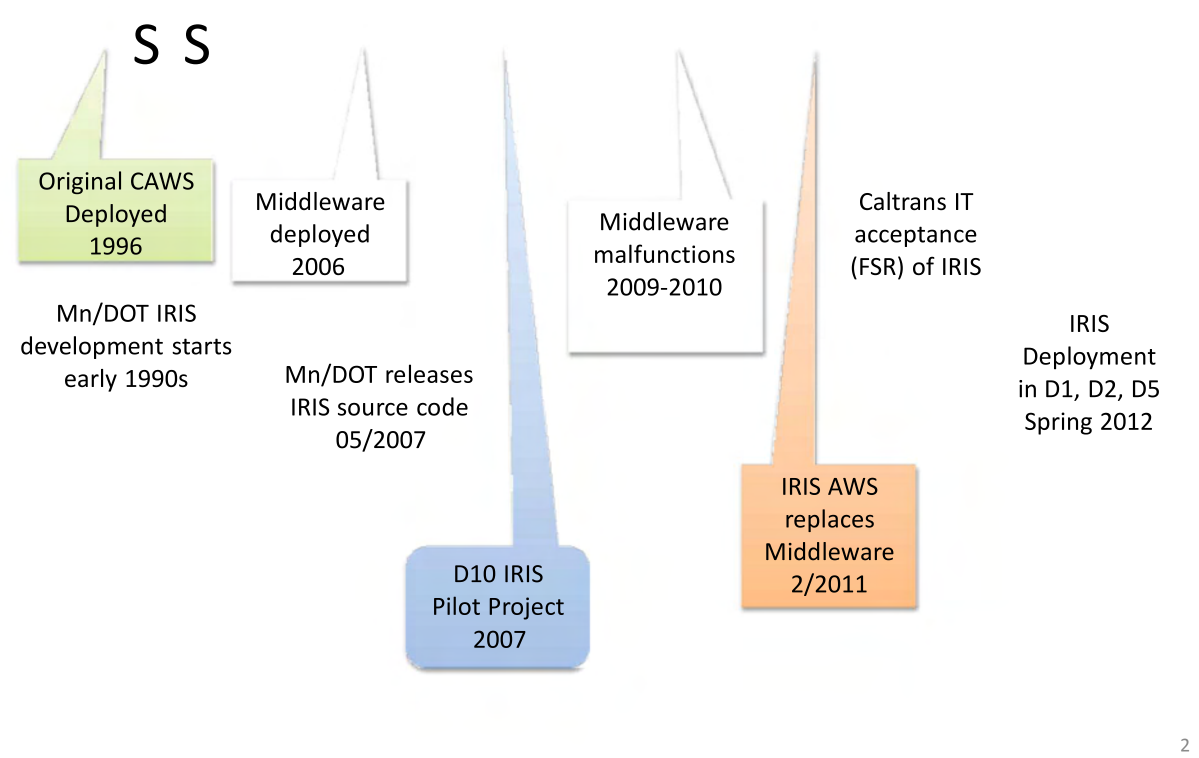

Figure 2.3: Caltrans District 10 + IRIS timeline

IRIS continues to evolve within Caltrans. An overview of the IRIS timeline is shown in Figure 2.3.

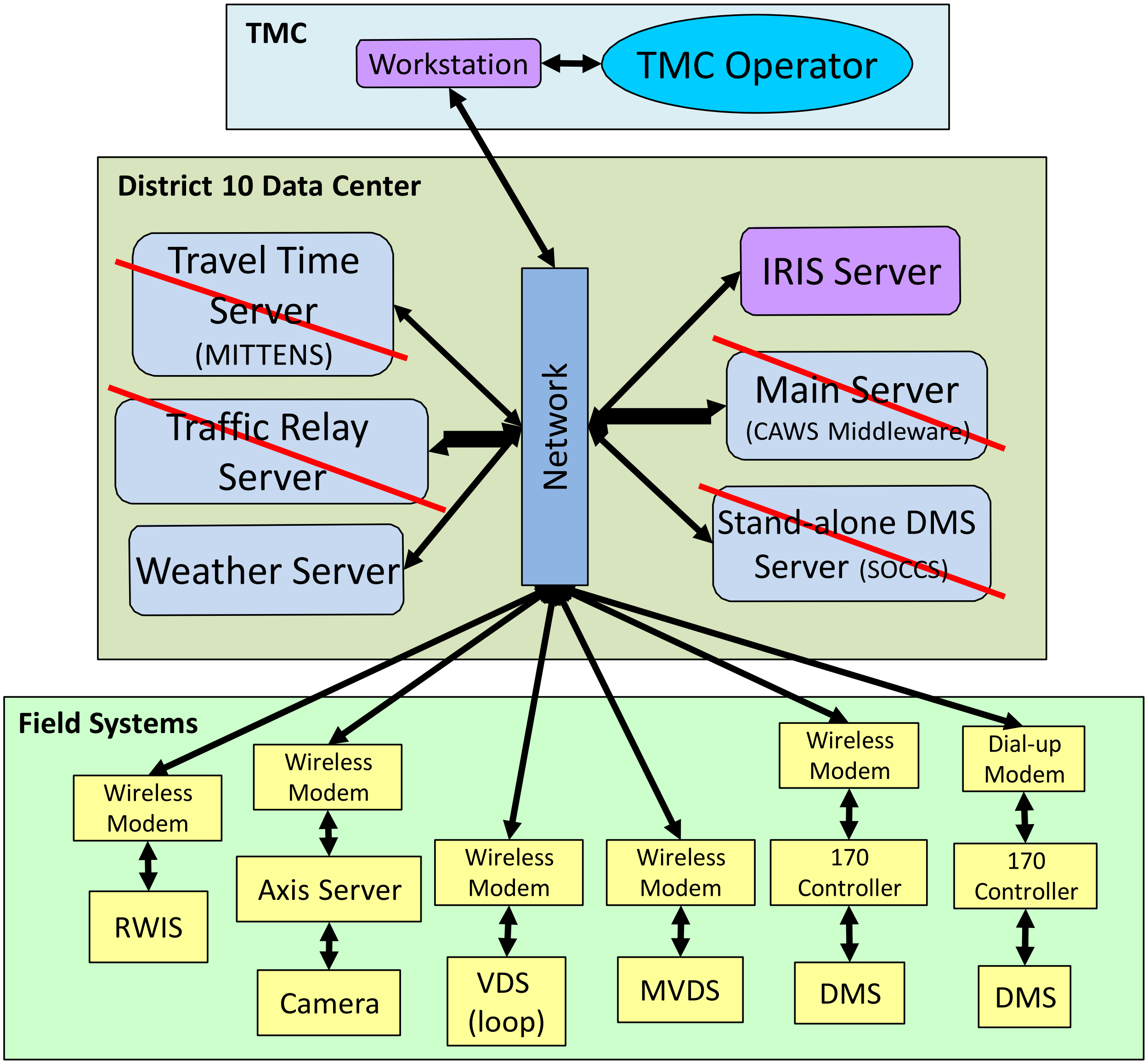

Figure 2.4: Caltrans District 10 architecture before and after IRIS

As an illustration of some of the benefits of IRIS, Figure 2.4 shows the Caltrans District 10 architecture before and after IRIS. Here the red hash marks illustrate that IRIS has eliminated the need for multiple separate systems including MITTENS Travel Time Server, Traffic Relay Server, CAWS Middleware, and Stand-alone DMS Server (SOCCS). Table 2.1 summarizes key changes.

Table 2.1: Benefits of IRIS: Before and after comparison

	Before	After
Mapping	None	✓
Number of server machines	4	1
Number of CMS controlled by Automated Warning System (AWS)	9	28 (unlimited*)
Types of VDS supported (MVDS, Loops, etc.)	1	5+ (unlimited*)
Types of RWIS supported (Manufacturers)	1	2+ (unlimited*)
Types of CMS controllers supported (Manufacturers)	1	4+ (unlimited*)
Mapped incidents	Via stand-alone app	Integrated
Standby backup system	None	✓
Source code availability	None or proprietary	✓
		* via device driver interface

Through this join effort, AHMCT has enabled Caltrans to add the following features that are now available to partnering agencies i.e. Mn/DoT:

- Device drivers
  - VDS: MVDS (EIS RTMS), URMS 2070, Wizard
  - CHP Incident
  - RWIS (SSI)
  - PeMS
  - Integration with external AWS (no longer used by District 10)
- Automated Warning System (AWS)
- Testing
  - Automated unit test cases
  - CMS simulation
- Generalization of IRIS, e.g. system attributes

- CMS message library
- Google Earth output
- RWIS map integration

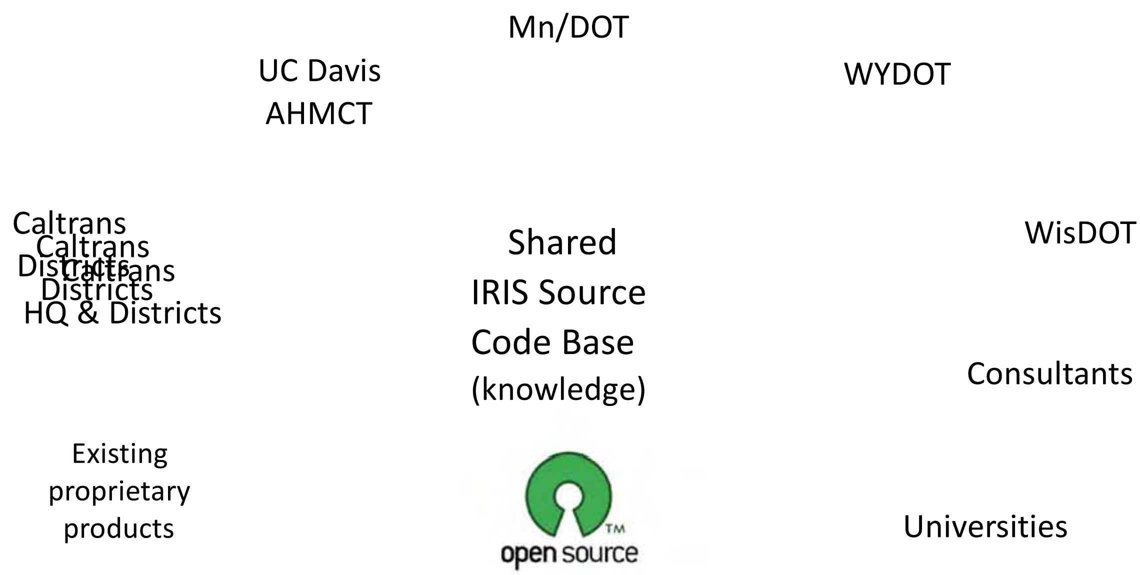

Figure 2.5: Multi-agency collaboration

Although less tangible, an important outcome of this research is the shared collaborative development model between Caltrans, Mn/DOT, AHMCT, and others (Figure 2.5). Recently, the collaboration has included consultants at the system engineering level, and is soon anticipated to include them at the design, coding, and test level.

CHAPTER 3: IRIS DESIGN AND FUNCTION

IRIS Environment

The following are important aspects of the IRIS runtime and development environments:

- Client-server architecture
- All dependent software packages are open-source
  - Free, no purchase requisition forms
  - No non-disclosure agreement (NDA) required
- IRIS is written in the Java programming language
  - ~150K lines of code
  - IRIS's design is heavily object-oriented, with ~1000 classes. The code is relatively complex, yet well-organized.
  - The IRIS build environment is Linux-based, and utilizes Apache Ant, the RPM Package Manager, and the Mercurial distributed version control system.
- Server
  - On the server side, our IRIS implementation consists of several independent services, including the main IRIS server, the IRIS video server (a video stream handler for CCTV video), sensorserver (a communications module for Signview-based CMS), casper (a CMS simulator), and a handful of other services that help support various other IRIS functions, such as RWIS, CMS XML feeds, log management, backup-server synchronization, etc.
  - Server-side requirements include OpenJDK, the Apache HTTD web server, Apache Tomcat, and the PostgreSQL database (a.k.a. "Postgres"), all of which are free, open-source technologies.
  - IRIS uses LDAP (Lightweight Directory Access Protocol) to authenticate users. An external LDAP server can be used, or a dedicated LDAP server can be run on the IRIS server.
- Client
  - The IRIS client software is launched using JWS (Java Web Start) technology, so nothing needs to be installed on the client machine in order to run IRIS (other than a Java runtime environment, which most machines already have installed).

  - Being a Java application, the IRIS client is cross-platform, and can theoretically run on any Java-compatible system. Current testing has proven its compatibility on various Windows and Linux platforms.

Challenges

In developing IRIS code, there were distinct challenge areas and successes. Both are noted here. Many of the challenges remain.

- At this phase of the project, the IRIS product was fairly established in District 10. District staff had acquired extensive experience with system. The system was fairly stable. There remained some defects that had been identified earlier, however, since they could not be reproduced, these defects simply remained on the "radar". Having achieved a stable, well-established system in District 10, the Department was able to officially approve a project to deploy this system to other districts. The focus was therefore shifted to deploying this system elsewhere in Caltrans, namely to Districts 1, 2, and 5. With a full-fledged project now underway starting in April 2011, some weaknesses were revealed in the ability to continue developing and deploying this system for other locations. Unlike the early days of IRIS research and development, at this phase of the project, only one primary developer was assigned to this project. This had a detrimental effect on the project when that resource was no longer available to the project. The project was able to redirect a new development resource. Unfortunately, that resource was not available full time to the project. Furthermore, due to the lack of an adequate transition period with the previous developer, the project suffered several months of delay.

- Managing ticket priorities: defer or fix? There is often tension between maintenance and new development. New development is subject to a schedule with deadlines. However, maintenance and bug fixes are often urgent. With limited resources, this prioritization is difficult.

- Bottom-up versus top-down design: Some features were not anticipated in the initial IRIS design, so there is often a tension between forcing a new feature to fit into the existing framework and modifying the framework itself to better support the new feature.

- Some unexpected regressions were experienced after developing new features or fixes

  - The creation of automated test cases has helped to catch regressions

  - End-to-end automated test cases throughout IRIS are needed

- Lack of mid- and low-level technical documentation

  - Software is documented using JavaDocs within the "classes" and "packages" of the source code. This is helpful, but not sufficient.

  - No system architecture document

    - No data flow diagrams
    - No detailed software or interface design documents
    - No comprehensive installation, administration, or maintenance documentation

- Tradeoffs between customized, agency-specific development and generalized development (i.e., implementing features that other agencies can make use of), and the pressures and time constraints associated with making these tradeoffs
    - Writing agency-specific code is easier and significantly faster
    - Generalizing an existing feature (i.e., to make it something other agencies can make use of) can be complex and consume significantly more developer time
    - However, generalized code, if ultimately merged upstream, reduces the complexity of maintaining multiple parallel development branches and changesets, saving a large amount of developer time in the longer-term
    - Code merges are voluntary (by both AHMCT and Mn/DOT). However, maintaining unmerged changesets is extremely undesirable, difficult, and time-consuming, especially over the long-term, therefore the tendency should be to merge as much as possible, and only keep unmerged those changesets which absolutely cannot be merged. This requires more resources upfront, but tends to minimize resource requirements in the future.
- Handling and timing merges between agencies
    - Requires constant communication between agencies to identify common features versus agency-specific features and how best to incorporate them
        - Has resulted in a long back log of unmerged change sets between agencies
    - The serial, single-branch model used by AHMCT to manage its changesets has shortcomings and has made cherry-picking features for upstream merging difficult as features are often intermingled among multiple changesets. In the future, a parallel model based on feature branches will be used.
- Excellent generalized designs take additional effort
    - E.g. RWIS (Road Weather Information System) icons: metric, English, physical quantity classes
    - E.g. Incidents
    - Support for field element operation modes that are unused by D10 but might be used by another agency

- Communicating how the open-source process is different
    - Collaborative development model is important, but may be hard to convey
    - Cost savings are easy to explain

Successes

- IRIS has succeeded in providing proven-useful ATMS functionality
    - Being open-source and well designed, the ability to customize IRIS for an agency is endless
- IRIS assumed Automated Warning System (AWS) functionality (4 months start to end)
    - Developed VDS (Vehicle Detection Station) data acquisition (RTMS or Remote Traffic Microwave Sensor, loops)
    - Developed RWIS data acquisition + user interface
    - Forwards traffic to PeMS (Performance Measurement System)
    - Developed AWS module
    - Testing and verification
- Reliability and code quality of IRIS are good
    - The IRIS server is especially robust
- Low cost, less than one-fourth the cost of proprietary ATMS [1].
- Collaborative design – while there is always room for improvement, the development process between AHMCT and Mn/DOT has worked well
- Ability to customize and generalize IRIS as needed to fulfill district requirements and feature requests
- Managing code between multiple agencies
    - Has worked surprisingly well (with exceptions)
    - Changeset concept works well. Any needed code differences are well managed through the use of the distributed version control model.

IRIS Functional Areas

The IRIS functional areas are described next. Screen shots are provided to illustrate each area, followed by brief descriptive text.

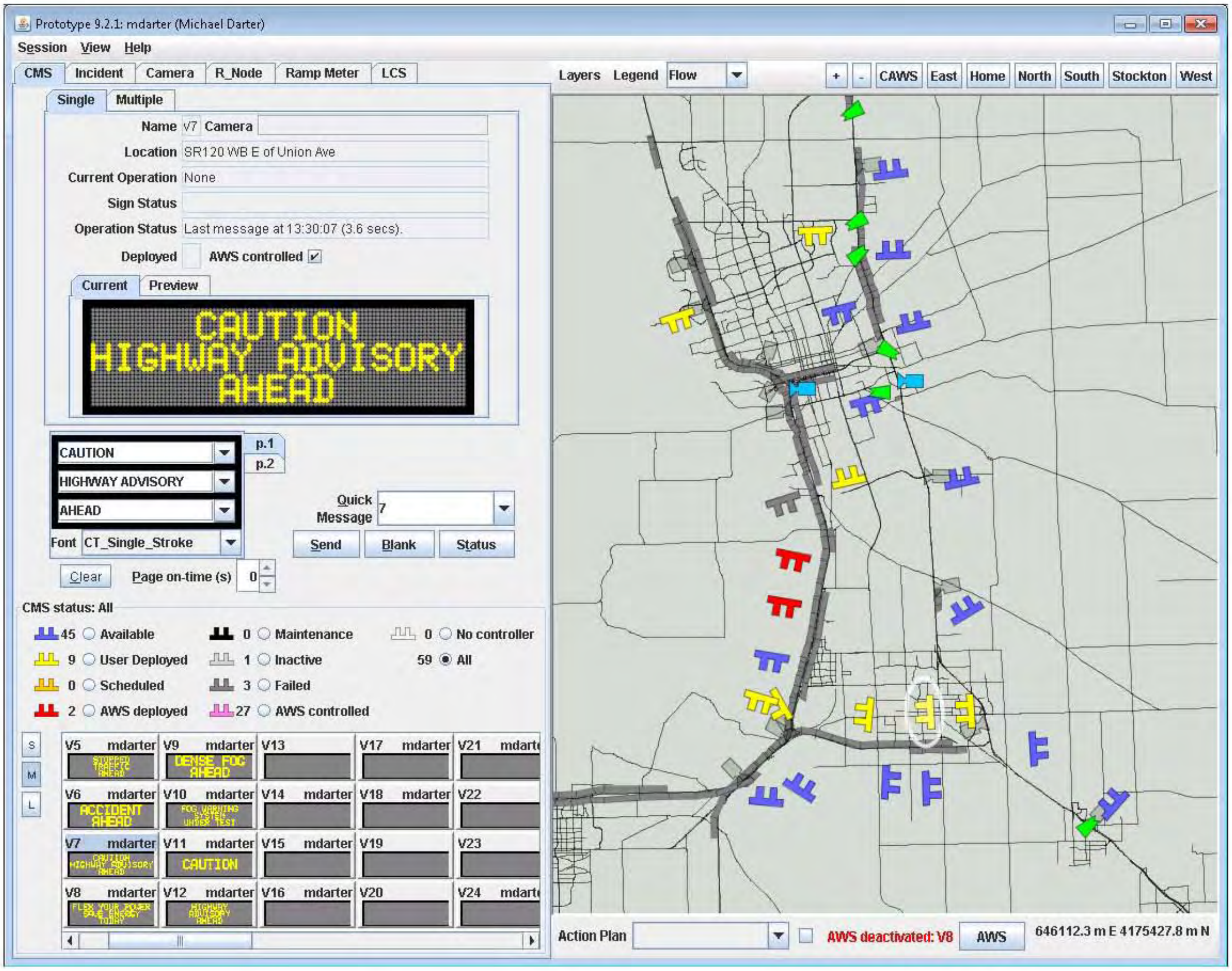

Figure 3.1: IRIS CMS control and monitoring

Figure 3.1 provides a view of the IRIS client, specifically the interface for CMS control and monitoring. Note that the map (right hand side) is going to be substantially upgraded in Release 9.3. The map shows sign locations and color-coded status, and the currently selected (circled) sign. The left side is the detailed CMS panel. At the bottom, the preview grid shows a small preview of all the signs. Above that is a radio button selection panel to filter signs by status (e.g. user deployed). The panel above that is for message composition. The top panel provides detailed status for the currently selected sign, as well as preview of the outgoing message.

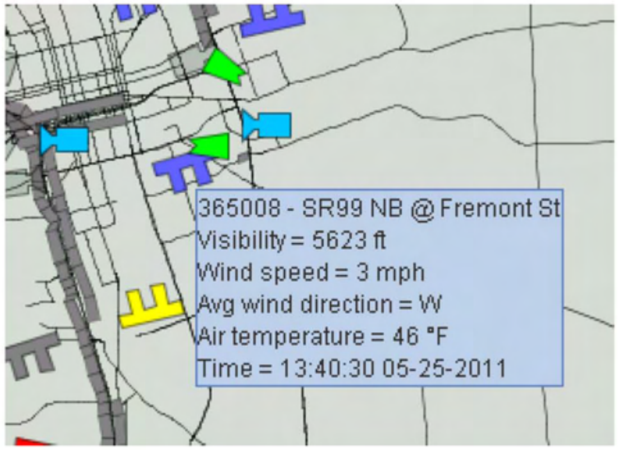

Figure 3.2: RWIS map icons

The Road Weather Information System (RWIS) is shown in Figure 3.2. A popup window appears on hovering the mouse over an RWIS icon, and provides the RWIS information. The information includes visibility, wind speed, average wind direction, and air temperature. RWIS information is an important component in the AWS decision process.

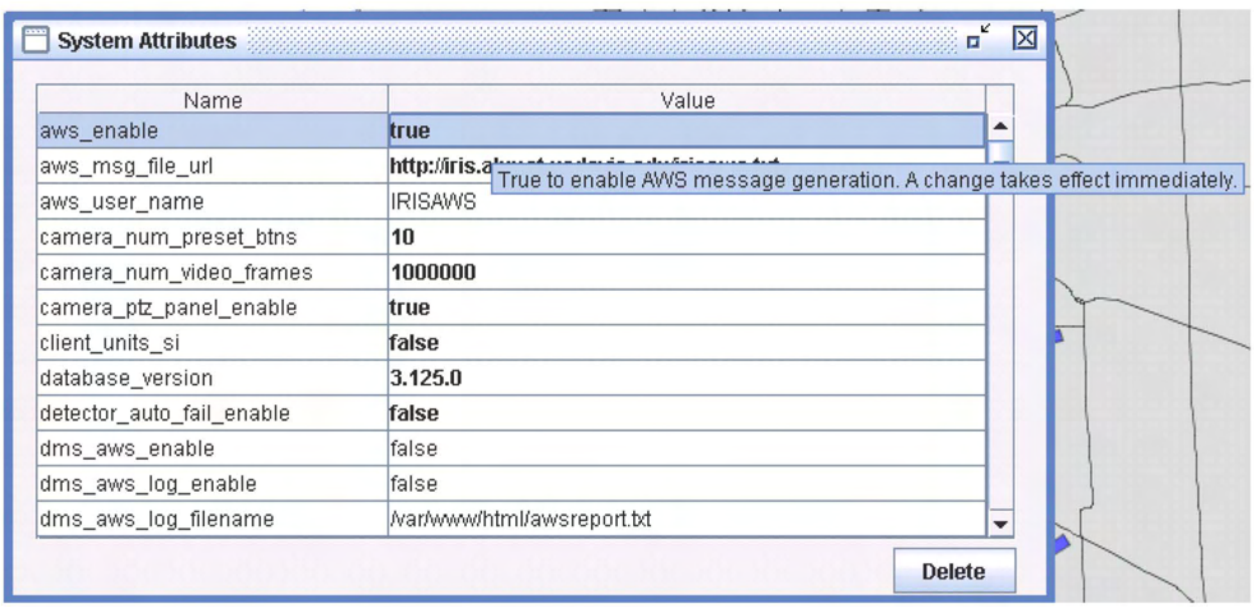

Figure 3.3: System attributes editor

Figure 3.3 shows the system attributes editor. IRIS has a large number of system attributes that modify its behavior without the need to recompile. This is particularly important for deploying in multiple districts, as well as multiple states. Thus, it is a significant factor in the ability for IRIS to be used in multiple states without forking the source code. A project fork happens when developers take a copy of source code from one software package and start independent development on it, creating a distinct piece of software.[2]

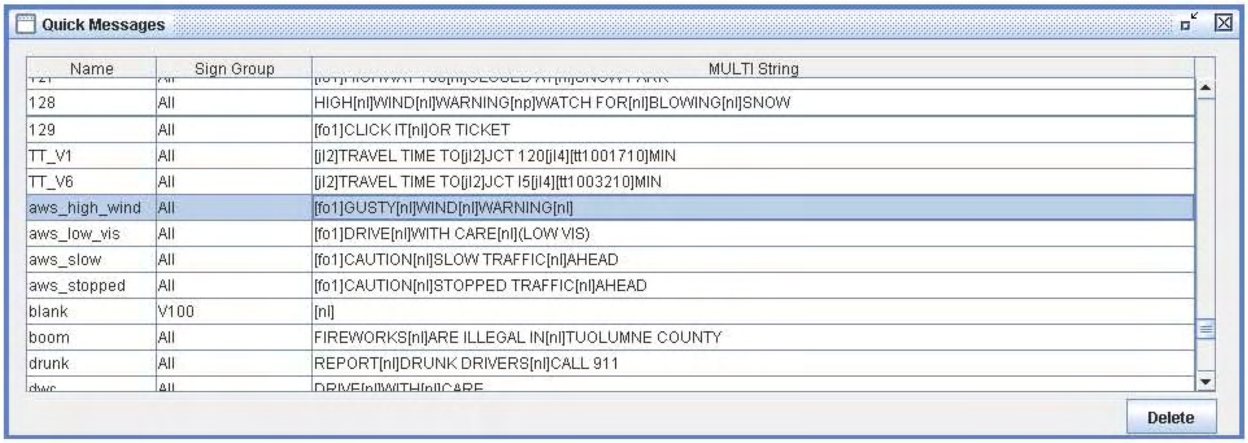

Figure 3.4: CMS message library

The CMS message library is shown in Figure 3.4. This library contains regularly used messages to allow easy CMS configuration. Operators can select messages from this library in order to quickly and accurately populate a given CMS to deploy a message.

2 See http://en.wikipedia.org/wiki/Fork_%28software_development%29

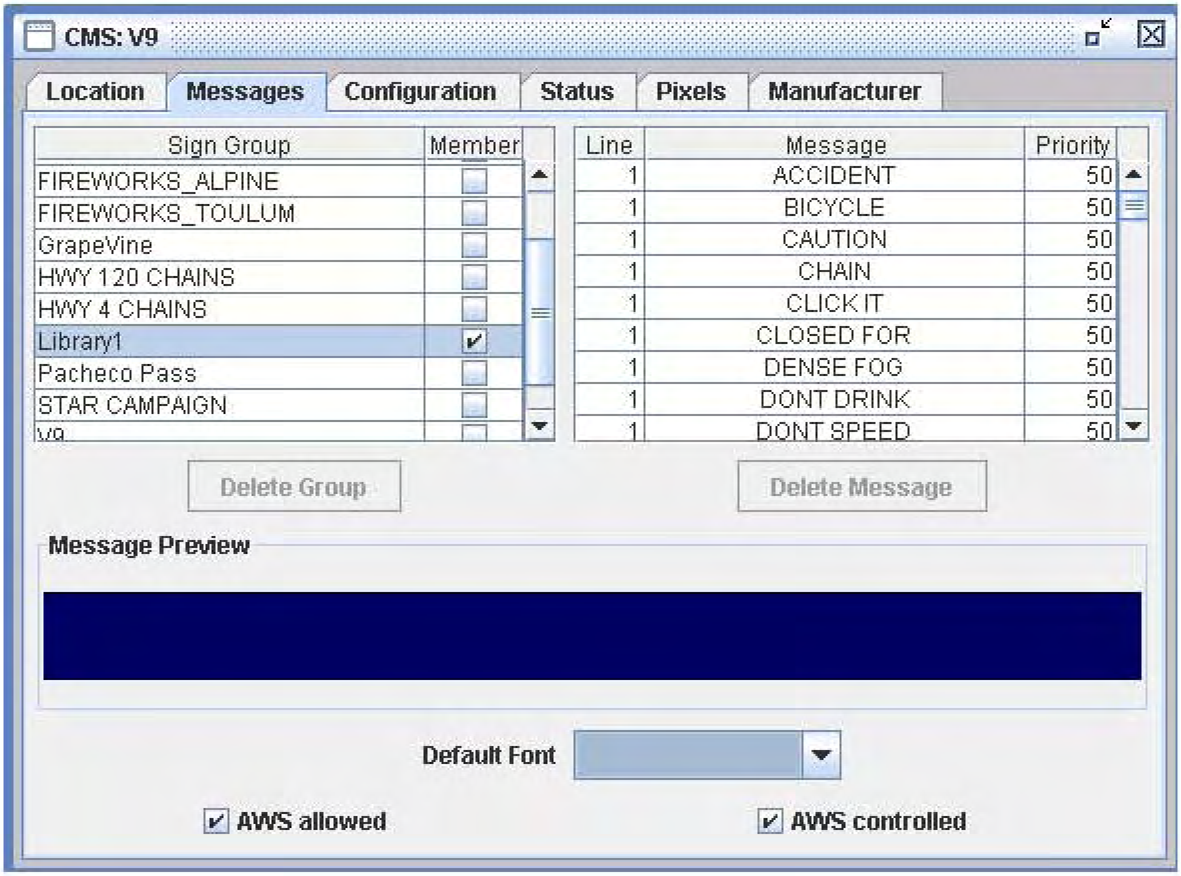

Figure 3.5: CMS definition

Figure 3.5 shows a part of the interface for configuring a CMS. IRIS has different access levels for a range of configuration tasks. Figure 3.5 shows sign number V9 being assigned to sign group Library1, which has messages shown in the right pane. This dialog also allows deleting messages from a selected library.

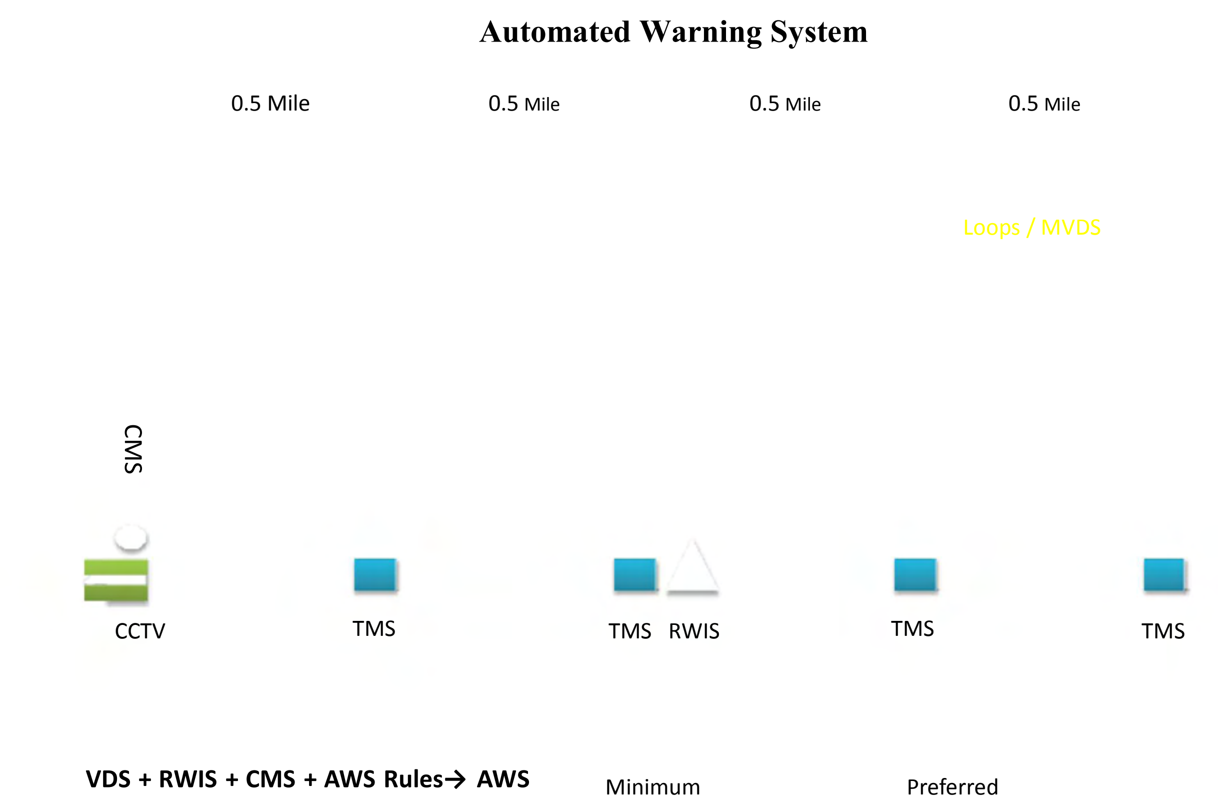

Figure 3.6: Typical Automated Warning System (AWS) configuration

Figure 3.6 provides a typical Automated Warning System (AWS) configuration. Here, CCTV is Closed-Circuit TV, TMS is a Traffic Monitoring Station, VDS is a Vehicle Detection Station, and MVDS is a Microwave Vehicle Detection Station. Under the current project, IRIS took on the full AWS for District 10. In addition, the AWS functionality was generalized so that other districts will be able to implement AWS when they deploy IRIS. The 4 basic rules of AWS are shown in Figure 3.7, with priority given by the item number, e.g. stopped traffic has the highest priority.

In March 2011 there were 3,913 CMS messages deployed in District 10. 73% (2,848) of those CMS messages were automatically deployed by AWS (73%), while 27% (1065) were “Custom” messages deployed by the TMC operators.

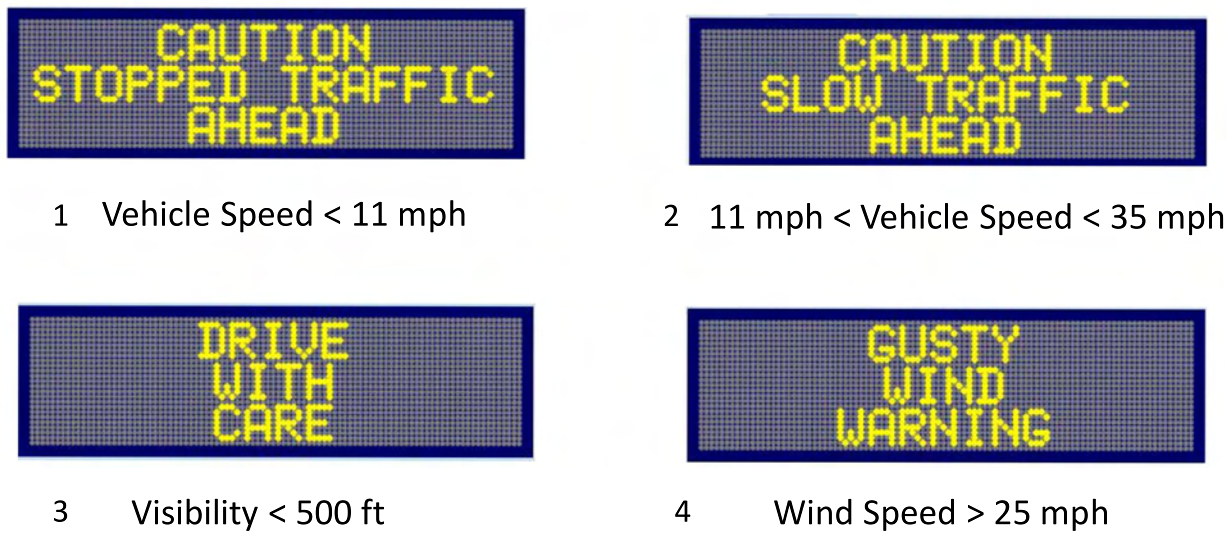

Figure 3.7: AWS message triggers

Figure 3.7 shows the AWS message triggers. IRIS looks for one of these 4 conditions to be present for 90 seconds at an AWS location, and then activates the CMS with the corresponding message.

Vehicle Detection Systems

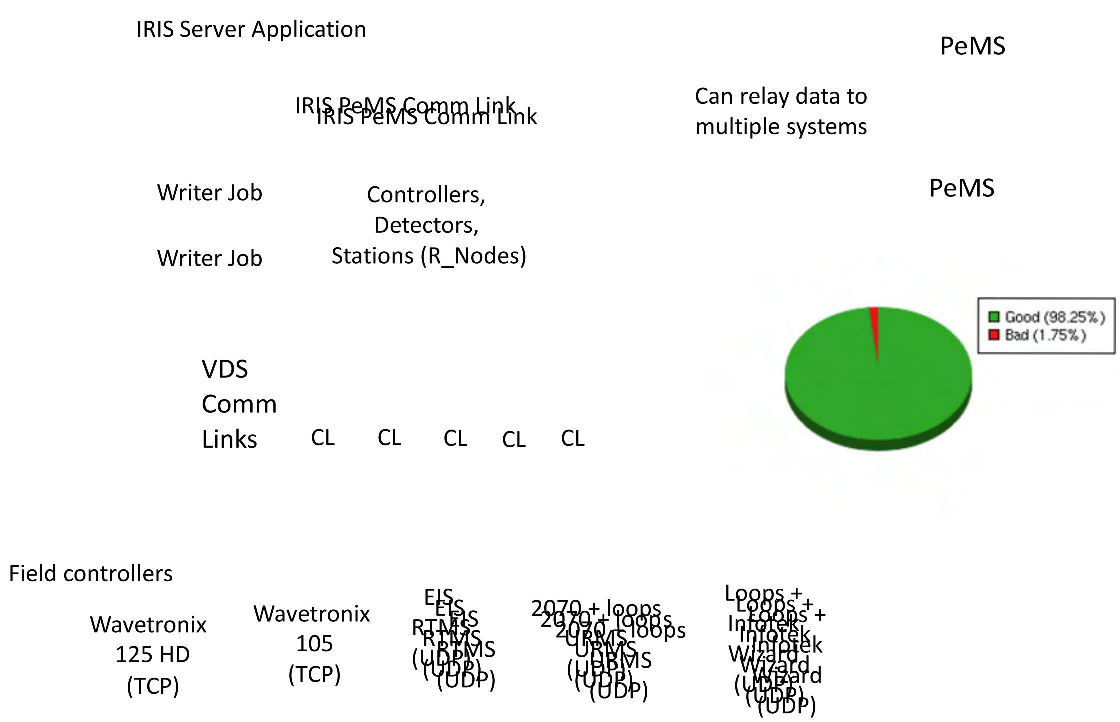

Figure 3.8: IRIS traffic data collection open-source data acquisition: Field→IRIS→PeMS

Figure 3.8 illustrates IRIS' capabilities with respect to a wide range of Vehicle Detection Systems (VDS), and its link with external systems, in particular PeMS.

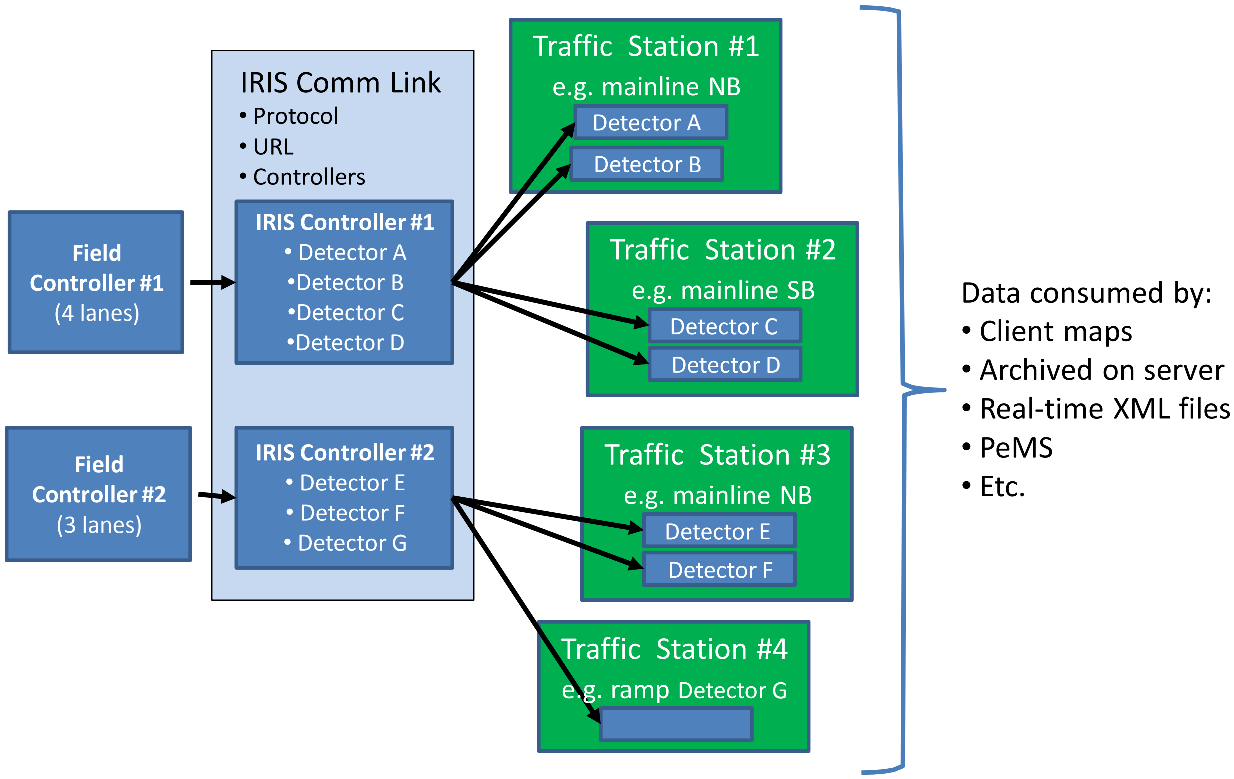

Figure 3.9: VDS configuration

Figure 3.9 provides a detailed view of VDS configuration. Field controller lanes are logically decoupled from station lanes. Lane order can be adjusted via specified station detector order.

Roadway Configuration

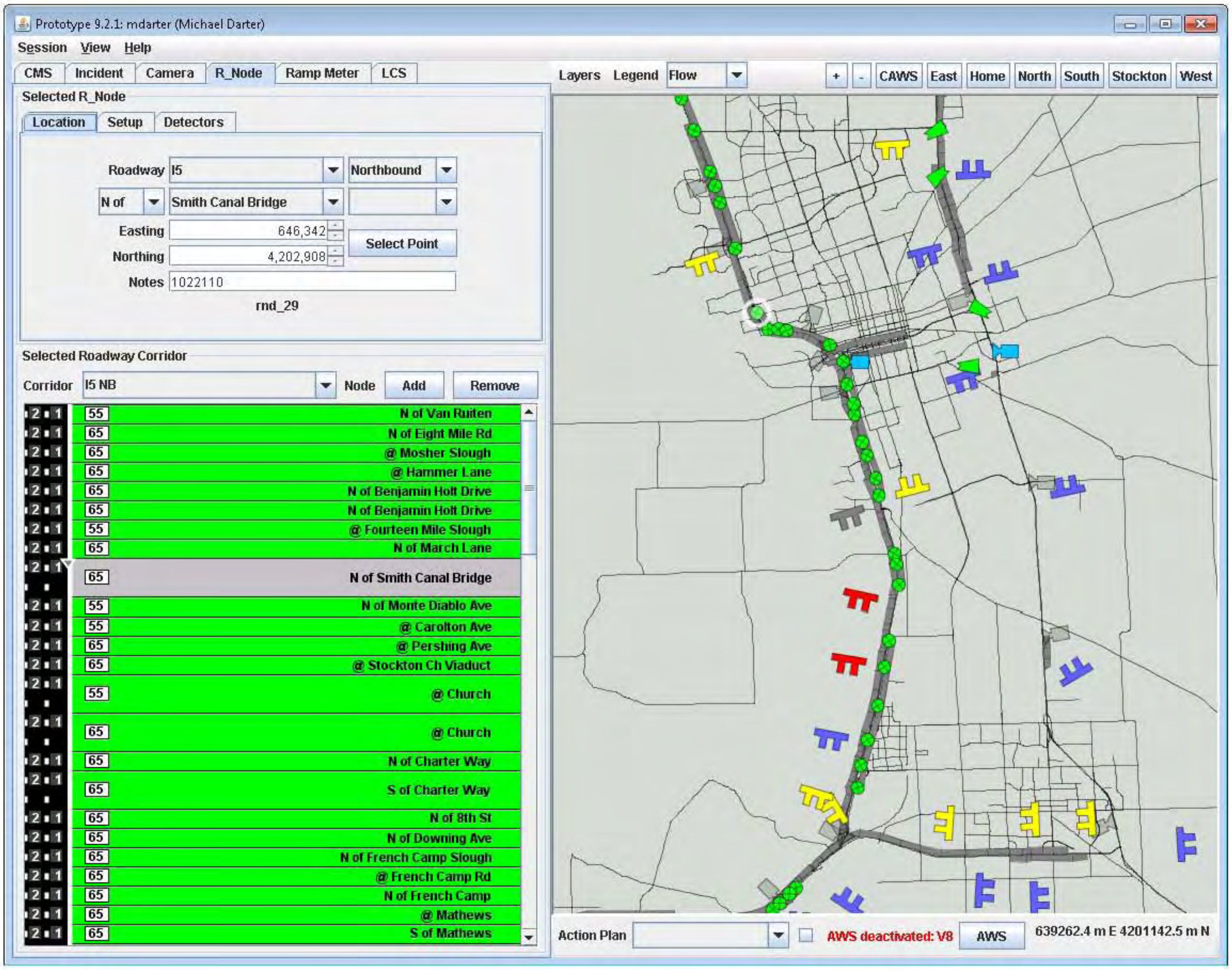

Figure 3.10: R_Node definition and editor

Figure 3.10 shows the R_node editor. This is where roadway segments are defined, including name and speed limit.

Communication Diagnostics

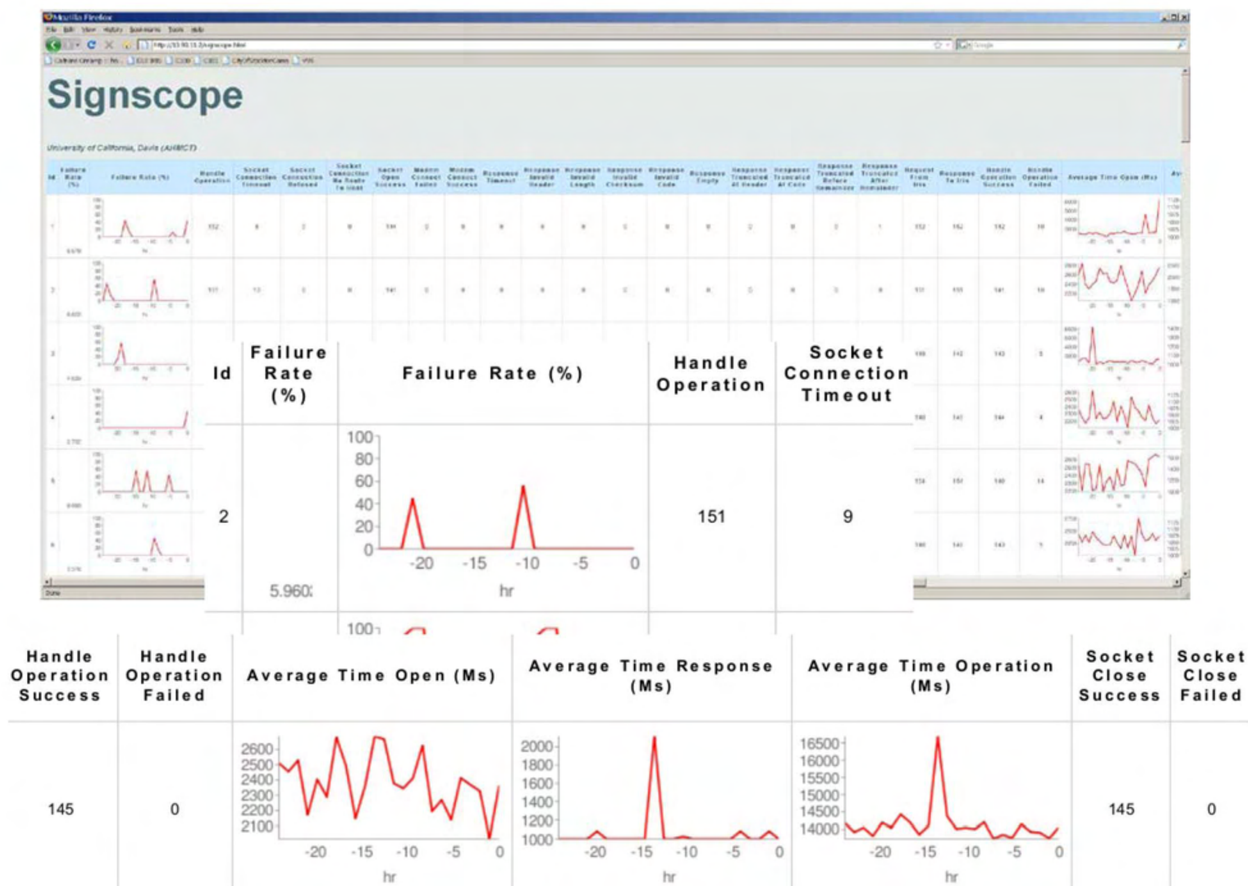

Figure 3.11: SignScope for diagnosing comm problems

Figure 3.11 presents SignScope, a diagnostic tool developed by AHMCT. This tool was essential in diagnosing communication problems, particularly during SensorServer development. It provides graphical illustration of communication response times and failure rates, making it easier to pinpoint trouble areas.

Data Flow and Architecture

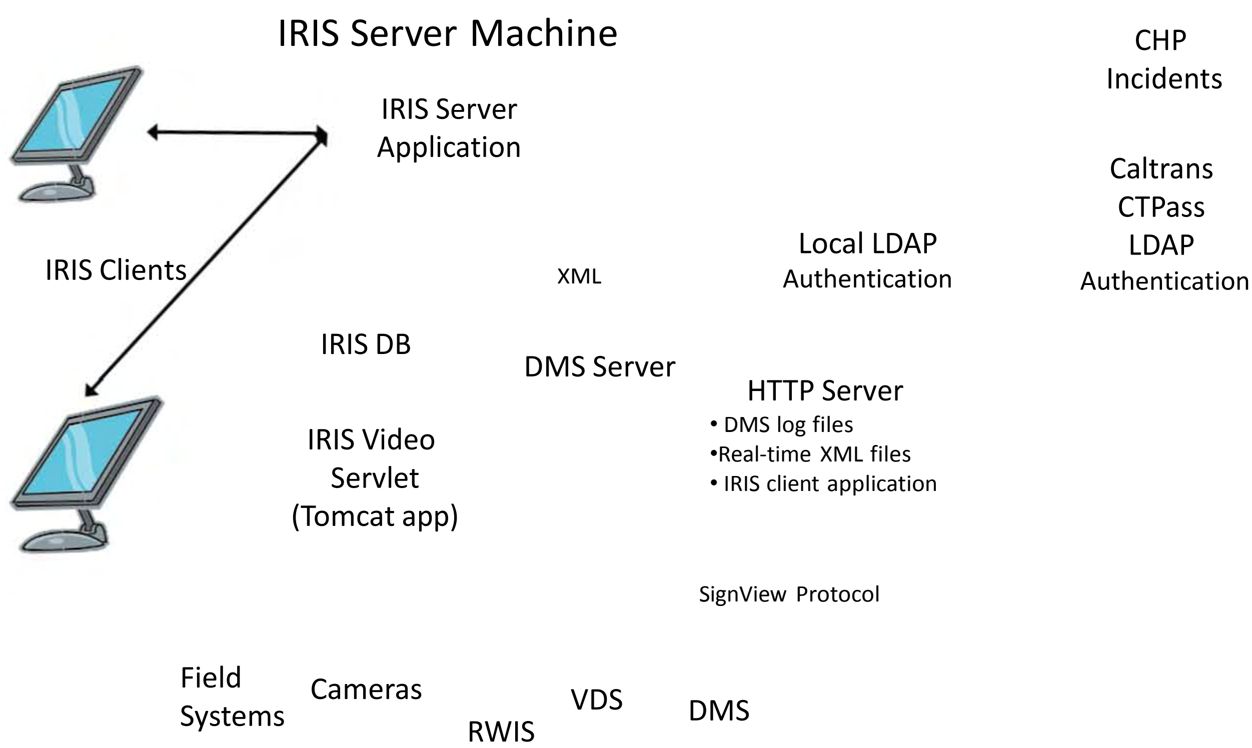

Figure 3.12: Data flow among IRIS applications

Figure 3.12 shows the data low among IRIS applications, including field systems and external systems, e.g. CHP incidents.

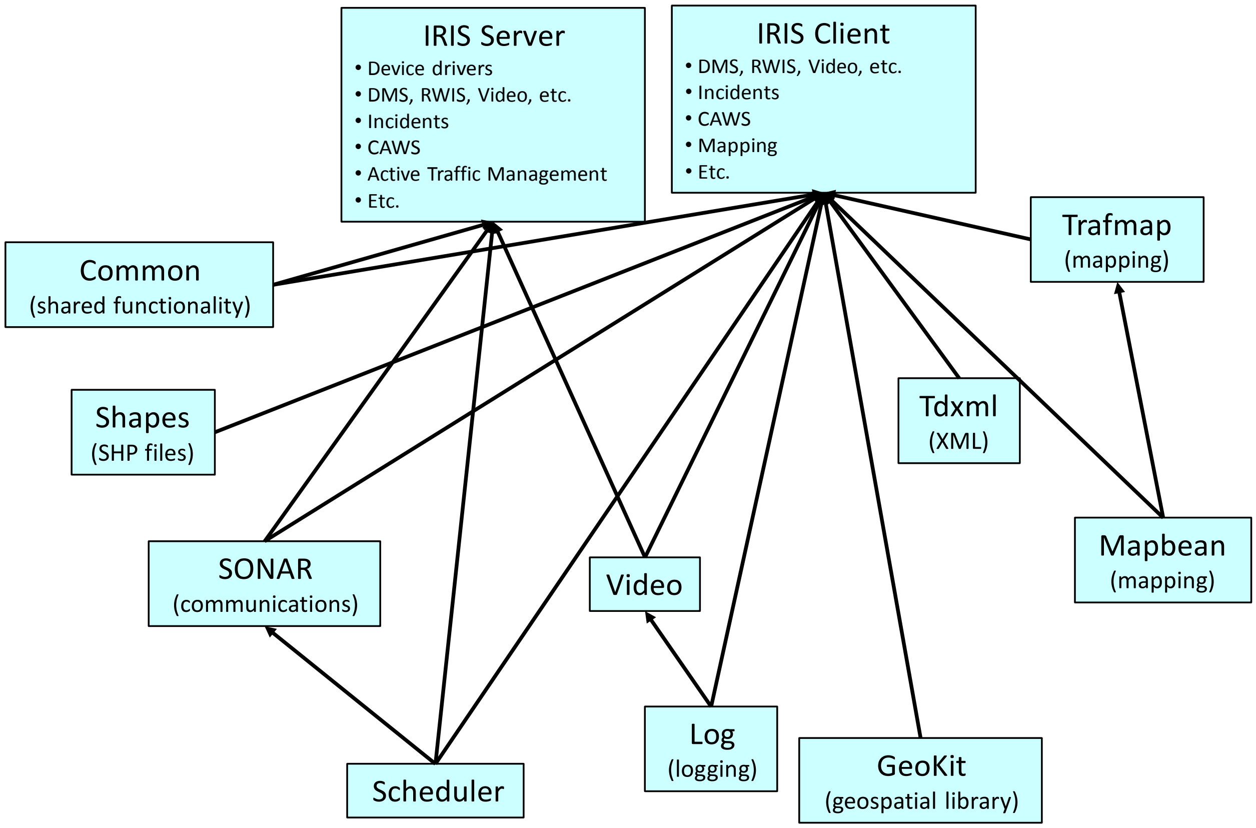

Figure 3.13: IRIS binary modules

Figure 3.13 presents the binary modules for IRIS, and their interconnections.

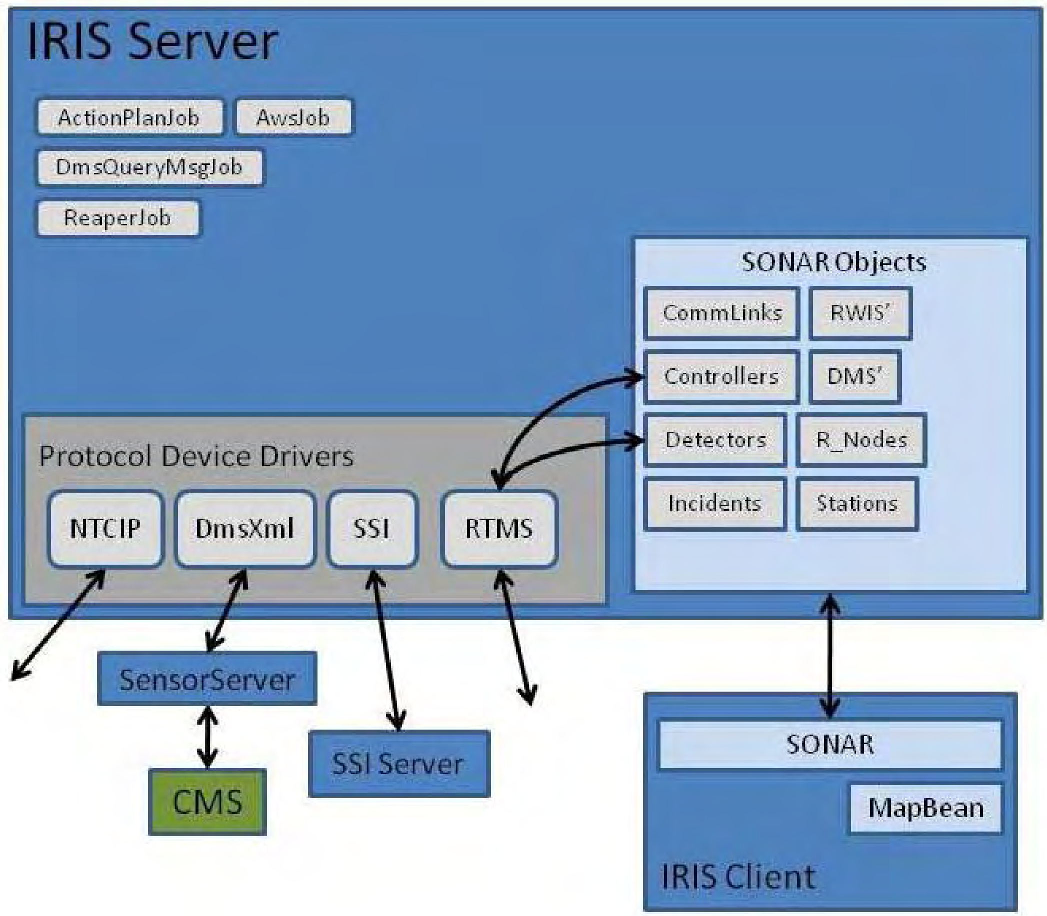

Figure 3.14: IRIS server architecture

Figure 3.14 illustrates the IRIS server architecture, and its connections with the IRIS client, SensorServer via DmsXml for CMS, and SSI server for weather.

Figure 3.15: IRIS architecture + design: Protocol device drivers

Figure 3.15 shows how IRIS communicates with external systems (hardware and software) using device drivers. A key ATMS function is to interface with external systems. IRIS has a powerful device driver interface for reading and/or writing to external systems. The drivers are open-source. A long-term IRIS goal is to develop and support as many device drivers as possible. Existing drivers include:

- VDS: Wavetronix 105, 125, EIS RTMS, URMS, Wizard, Canoga, others pending

- DMS: NTCIP A, B, C, SignView
- Video: Pelco D PTZ, Pelco switcher, Vicon PTZ, Vicon switcher, Manchester PTZ
- RWIS: Optical Scientific ORG-815, SSI
- CHP incident feed
- PeMS
- External AWS, e.g. CAWS

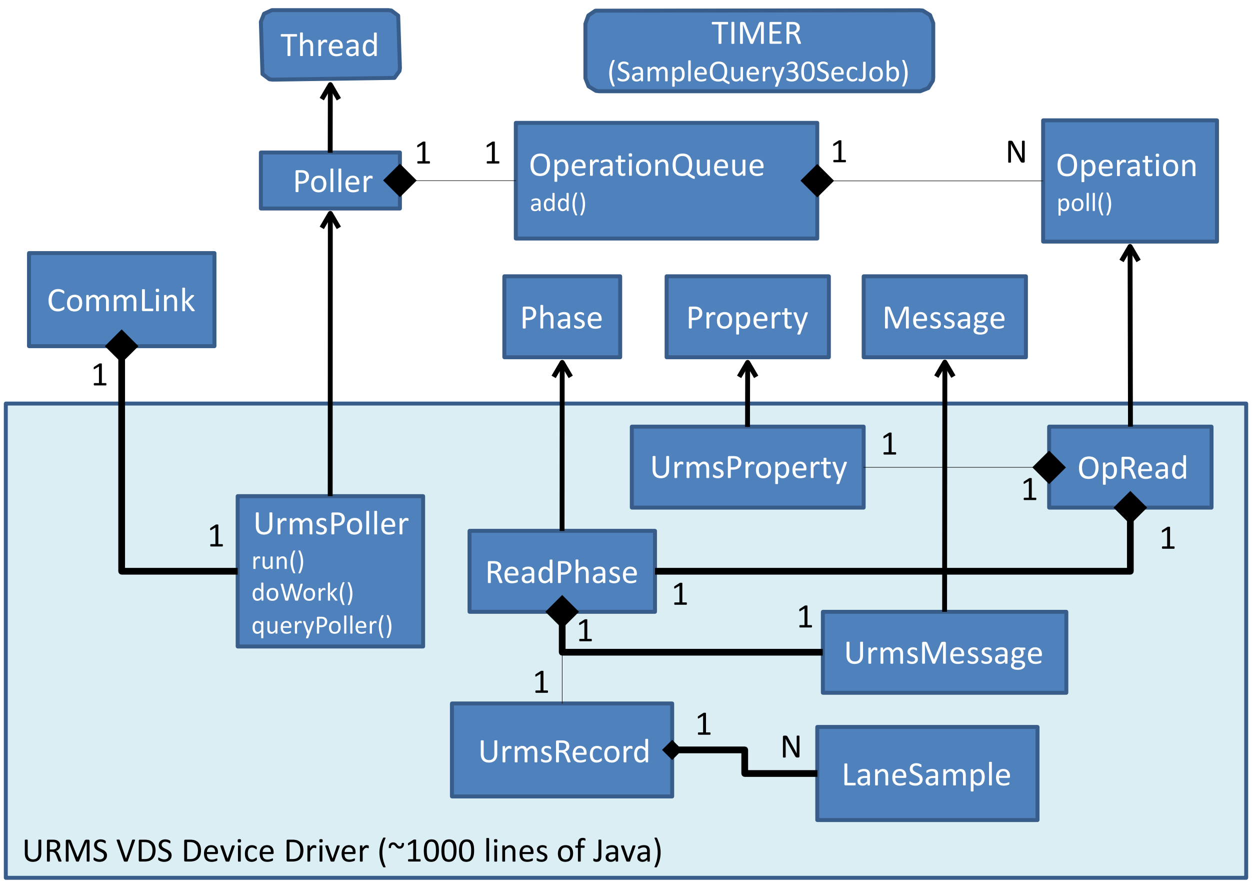

Figure 3.16: IRIS class structure for URMS device driver

Figure 3.16 gives the class structure for the Universal Ramp Metering System (URMS).

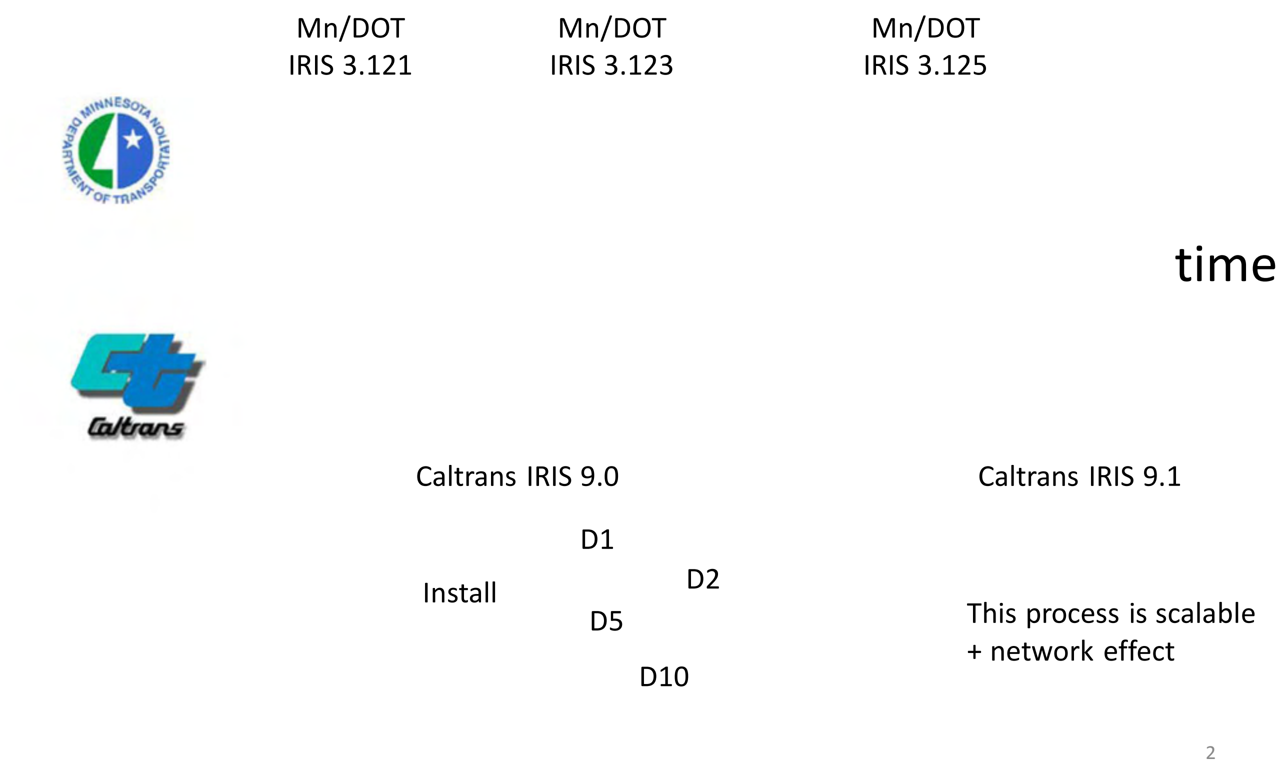

Figure 3.17: IRIS release and collaboration process

Figure 3.17 illustrates the collaboration process between Mn/DOT and Caltrans. It also shows recent major releases.

Video

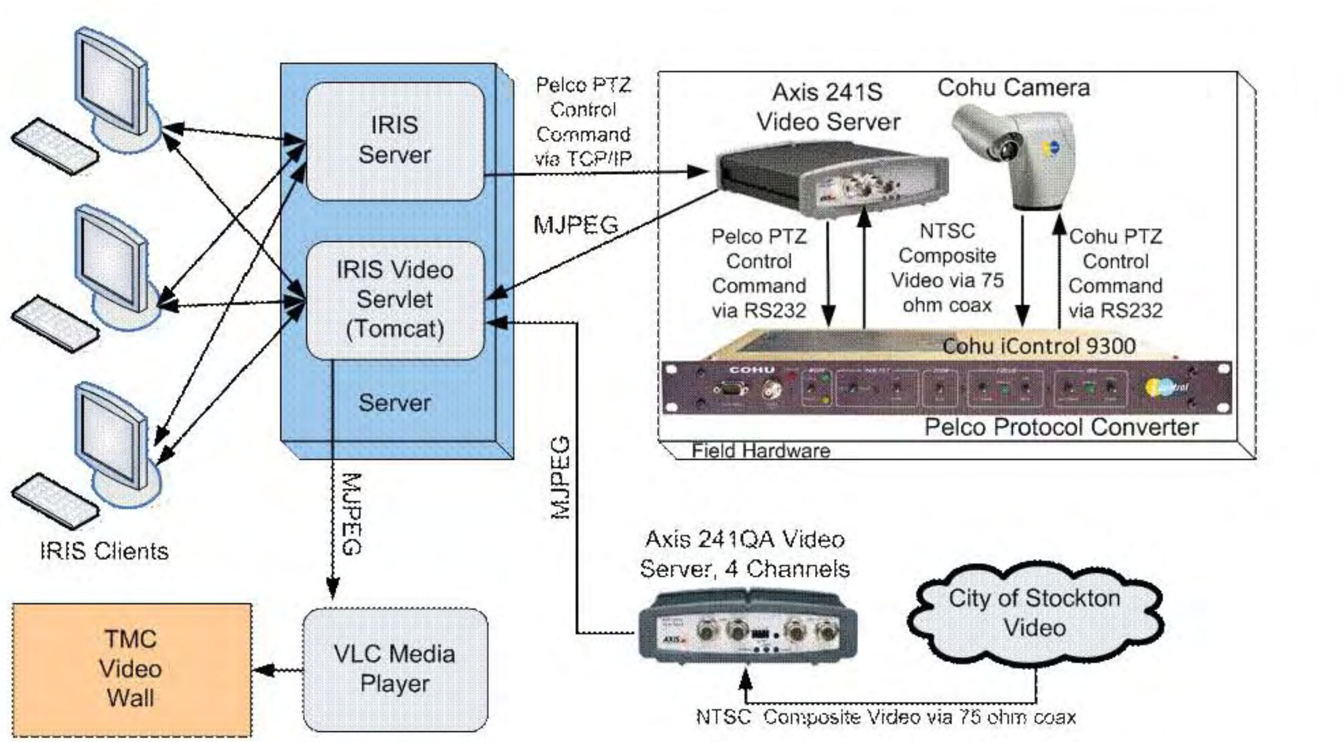

Figure 3.18: D10 video architecture

Figure 3.18 provides the District 10 video architecture.

Some challenges were encountered during the process of rolling out support for CCTV at District 10:

- Reliability problems were experienced with the Pelco PTZ driver
- User interface shortcomings were identified
    - Difficult to tell whether a particular camera is in maintenance mode, or should be working
    - No GUI feedback when connecting to cameras
- Support for configuring PTZ presets from within IRIS was lacking
- Developer resources were stretched, resulting in longer-than-desired turnaround times for fixes

Ultimately the priority of some of these issues was increased and our ability to perform CCTV-related testing in-house was enhanced, resulting in a better outcome.

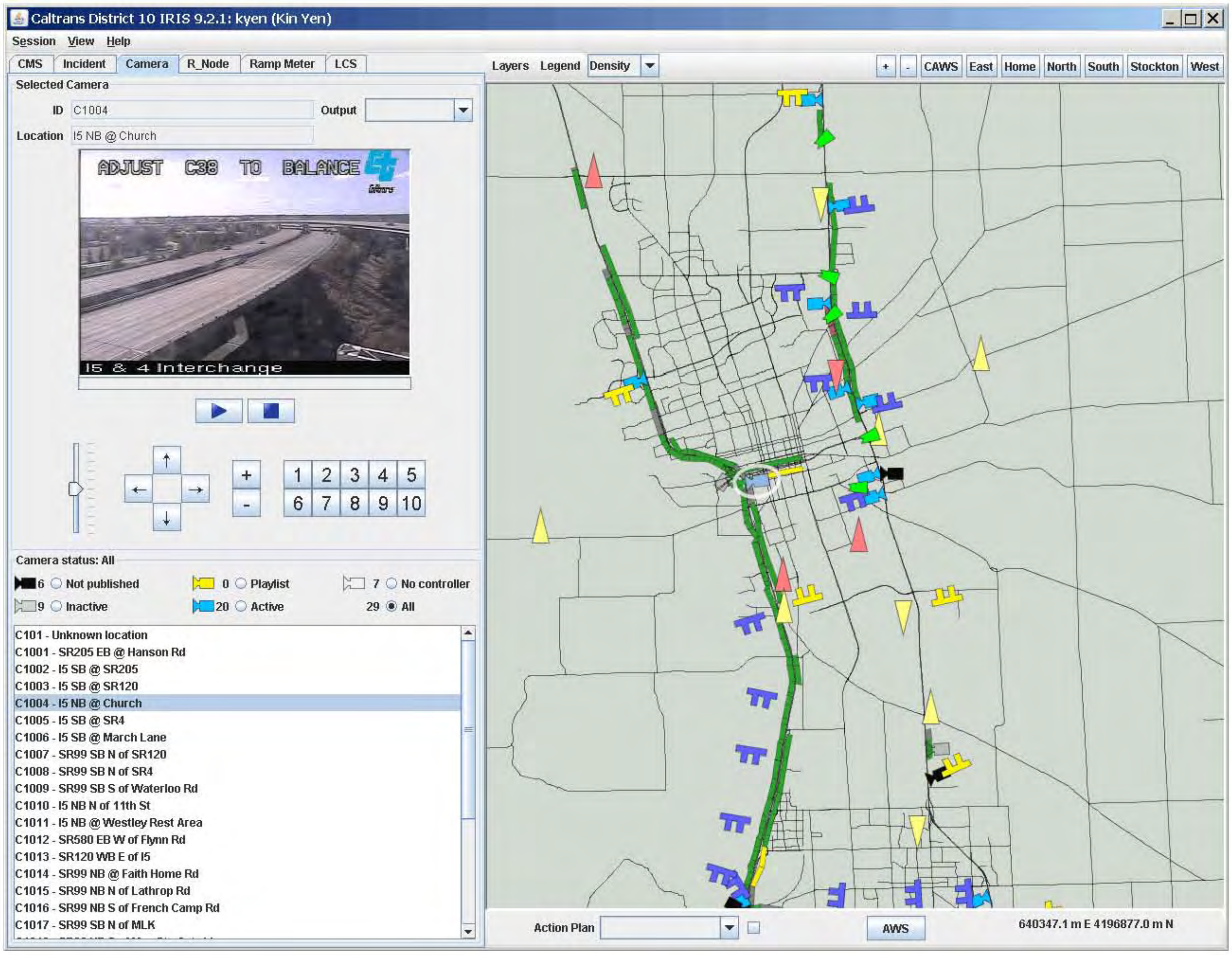

Figure 3.19: Video interface

Figure 3.19 shows the user interface for selecting, viewing, and controlling video cameras. Cameras can be selected on the map, or via the list in the bottom-left panel. Above that is a radio button selection for filtering the available cameras by status. Above that is the camera view and controls for the selected camera. Controls include pan, tilt, and zoom (PTZ) as well as up to 10 preset camera positions.

IRIS System Maintenance

Aside from ongoing software enhancement activities to meet user needs, required routine maintenance to keep the system running is minimal:

- Update IRIS configuration if field elements change or are added
    - E.g. VDS, CMS, cameras
- Delete contents of some log database tables
    - Comm_events, detector_events

Normal server maintenance includes:

- Check disk space (clear log files if necessary to free up space)
- Update operating system packages
- Backup:
    - IRIS PostgreSQL database
    - LDAP database
    - IRIS configuration files

Mapping

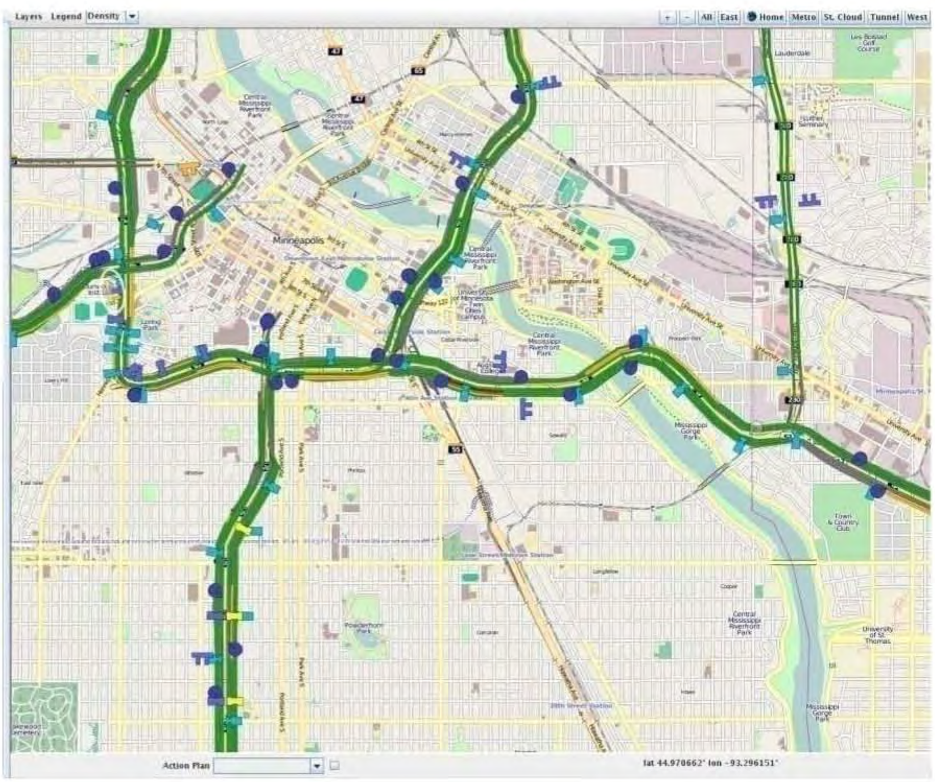

Figure 3.20: IRIS OpenStreetMap (OSM) mapping

Figure 3.20 shows the new IRIS mapping in operation in Minnesota. It is based on OpenStreetMap (OSM). The Caltrans IRIS will include OSM mapping in Release 9.3. Existing IRIS mapping has been shown in many of the above screen shots.

Continuing Enhancements

Ongoing and future IRIS enhancements include:

- IRIS Release 9.3 (in progress)
  - Support for D1/D2/D5 CCTV infrastructure
  - Support for PTZ presets
  - Enhanced VDS support (including RTMS G4 and Sensys devices)
  - OpenStreetMap support
  - CMS status XML feeds to provide data to external applications
  - Support for SUSE Enterprise Linux
- IRIS Release 9.4 and future
  - Ramp metering
  - Integration with TMCAL
  - URMS ramp-metering support
  - HAR support
  - Lane-closure support
  - Inter-district travel time generation

CHAPTER 4: CONCLUSIONS

Key contributions of this research project included:

- Replacement of CAWS and middleware, with functions assumed and generalized by IRIS
- Delivery of test drive IRIS systems to Districts 1, 2, and 5
- Testing and maintenance of IRIS for four districts. One (District 10) is in production and thus in maintenance, while three (Districts 1, 2, and 5) are test drives.

Benefits from the research and the continuing availability of the tools and data include:

- Availability of IRIS as a traffic management system for rural TMCs
- A unified tool and interface for device control and monitoring
- A collaborative model for cooperative development among multiple DOTs, universities, and private companies.

Lessons Learned

Over the course of the project, the challenges encountered have given us insight to better manage IRIS development and support into the future:

- Better developer transition planning is needed. At one point the project lost its lead developer and the transition to a new developer introduced undesirable delays. Having more than one developer with a comprehensive understanding of IRIS would help significantly. In addition, more complete internal documentation is needed to decrease the amount of time required for a new developer to get up-to-speed on the project.
- Maintaining unmerged code is not recommended. Rather than simply developing features that fulfill minimum requirements for a specific district, it is instead best to generalize these features as much as possible, so that they are more likely to be merged upstream into Mn/DOT's codebase. While this approach requires more time upfront, it will save time in the long run, since maintaining unmerged code outside of the main IRIS codebase is extremely time-consuming, especially during downstream merges. With unmerged changesets, each of these changesets must be refactored during every single downstream merge. In addition, writing more general code means that time will also be saved when attempting to support new Caltrans districts.
- The serial, single-branch model used by AHMCT to manage its changesets has shortcomings and adds to the difficulty of upstream merges to Mn/DOT. In the future, a parallel model based on feature branches will be used.

- A somewhat longer-than-desired backlog of patches accumulated toward the end of the project. An increase in communications and cooperation with Mn/DOT would help prevent this.

- Some defects were Windows-specific. Testing scope may need some rebalancing.

- Some defects were difficult to discover in testing due to differences between test and production environments. Better in-situ integration testing is needed.

Future Work

Based on the successful results of the current study, a variety of future work is recommended.

With the execution of the FSR, IRIS should be updated as needed, and fully deployed into Districts 1, 2, and 5. This effort is currently underway under a new research contract.

With IRIS deployed into Districts 1, 2, 5, and 10, on-going maintenance will be necessary. In addition, new features and enhancements will be requested, with development effort needed. This is planned under an upcoming maintenance contract.

REFERENCES

1. M.T. Darter, K.S. Yen, T.B. Swanston, B. Ravani, and T.A. Lasky, "Research & Development of Open-Source Advanced Transportation Management System Hardware and Software Components," Advanced Highway Maintenance and Construction Technology Research Center Rept. # UCD-ARR-09-08-31-01, 2009.

APPENDIX A:
RELEASE NOTES

(in report period)

- IRIS Release 9.0.6, installed 10/28/09, included the following tickets:
  - #299: Release 9.0.6 composite ticket
  - #316: defect repair: add CAWS report by time period.
  - #315: defect repair: Vertical CMS message centering is incorrect.
  - #314: defect repair: CMS 'get message' button behavior is incorrect.
  - #270: Sensor Server is returning the wrong error message when the phone line is busy.
  - #269: defect repair: duplicate logging issues.
  - #146: Complete configuration of permissions (originally a 9.0.0 requirement).
  - #295: defect repair: correct font combobox behavior.
  - #296: defect repair: correct quick message library combobox behavior.
  - #301: defect repair: CMS message lines being incorrectly trimmed.
  - #294: defect repair: null handling conformity.
  - #290: defect repair in CMS preview (not effected by Caltrans configuration).
  - #309: defect repaired: reinit message not clearing cached CMS message.

- IRIS Release 9.0.6a, installed 11/19/09, included the following tickets:
  - #325: defect repair: Sensor Server threading issue for modem CMS.

- IRIS Release 9.0.6b, developed, tested, installed 01/20/10, which includes the following:
  - #328: Incorrect blanking behavior for CMS already in error state.
- Additional support tasks and tickets that were completed:

- Task: LDAP server configuration issue due to prior IP address change.
- #327: CMS icons not showing up on map for view-only users.
- #117 and #126: investigation of missized message problem on V12 and V18.
- #308: multiple layout and mapping issues related to small screen sizes.
- #38: Fixed default selection problem with radio buttons in StyleSummary.
- Task: re-configuring V2, updating IMM steps.
- #323: Merge work, moving incident code from client to server.

- IRIS Release 9.0.7, developed, tested, installed 05/27/10:
- #334 Release 9.0.7 composite ticket.
- #323 Merge w/ Mn/DOT releases 3.97 to 3.114, free fixes and enhancements.
- #331 Client crashes sometimes during multi-selection.
- #308 The IRIS client needs to fit on a non-huge screen, map icon sizes.
- #311 Users should not be able to specify a page on-time less than the minimum allowed.
- #327 View-only users can't see CMS map icons.
- #332 Initial map screen is sometimes all pink.
- #320 Client freezes if a wrong user name / password entered.
- #298 Client randomly freezes and can't close.
- #321 Intermittent client login failure with EOF message (related to #298)
- #303 IRIS logout and log back in failure.
- #313 Map not visible when the user changes to Cameras tab

- IRIS Release 9.0.7a, developed, tested, installed 06/30/10, which includes the following:
- #350 Release 9.0.7a composite ticket.
- #355 merge, test, install.

- ◦ #172 Providing real-time CMS information to DRI state-wide system.
- ◦ #349 Change button text from "Get Msg" to "Status".
- ◦ #351 CMS message text not displayed correctly in Google Earth

---

Release 9.1 (D10: 11/08/10, D1: 11/18/10, D2: 11/24/10, D5: 11/29/10)

- Benefits:
- ◦ improved service by reading all microwave and loop VDS data and forwarding to PeMS
- ◦ improved reliability by eliminating multiple servers, databases,
- #340 Composite 9.1 ticket
- #339 Merge w/ Mn/DOT releases for 9.1 (IRIS 3.115-3.121 onward)
- #117 fix over-sized message warning problems for over-sized messages (width and height).
- #347 Update terminology (e.g. remove 'wizard', etc.).
- #359 Ability to read real-time traffic data via the EIS RTMS protocol driver.
- #361 Forward traffic data collected by IRIS to PeMS.
- #360 Ability to read real-time VDS loop data via a new protocol driver.
- #336 Support of multiple terminal server lines in sensorserver.

9.1.1 (D10: 11/18/10)

- #382 Enhanced logging

9.1.2 (D10: 12/02/10)

- #370 Slow login while real-time traffic collection is activated

---

9.1.3 (D10: 12/13/10)

- #390 Correct handling of traffic anomalies in PeMS protocol driver
- #391 Wizard protocol driver reading single loop VDS protocol

---

9.1.4 (D10: 12/16/10)

- #396 Incorrect datagram and IP mapping in Wizard driver

- IRIS Release 9.2 (Installed 01/05/11)

Purpose: provide AWS capabilities

- ◦ Add ability to generate AWS messages
- ◦ Test and Verify AWS functionality
- ◦ Proper closing of Comm link scheduler
- ◦ Interface with SSI RWIS
- ◦ RTMS VDS driver handling of correct number of lanes

- IRIS Release 9.2.1 (Installed in D2, D5, D10 04/11/11)

Purpose: AWS maintenance and enhancements

- ◦ Merged Mn/DoT change sets 3.122-125
- ◦ Add RWIS icons and real-time data to map
- ◦ Add support for the URMS ramp meter/ VDS protocol
- ◦ Wizard VDS protocol driver handling crazy field data
- ◦ CMS appears blank in client but message sent (garbage collection)
- ◦ AWS state of deactivated undeploying CMS should be updated
- ◦ Scheduled messages should not activate sign every 30 seconds

---

- IRIS Release 9.2.2 (Installed 6/28/11)

Purpose: Fix Camera control problems

◦ CCTV - change from sending UDP to TCP packet for the camera control with existing Pelco-D protocol.

- ◦ Develop a Cohu protocol driver for IRIS via TCP to Axis 241S video server.
- ◦ Not merged w/ Mn/DoT