STATE OF CALIFORNIA • DEPARTMENT OF TRANSPORTATION

TECHNICAL REPORT DOCUMENTATION PAGE

TR0003 (REV 10/98)

ADA Notice

For individuals with sensory disabilities, this document is available in alternate formats. For information call (916) 654-6410 or TDD (916) 654-3880 or write Records and Forms Management, 1120 N Street, MS-89, Sacramento, CA 95814.

1. REPORT NUMBER	2. GOVERNMENT ASSOCIATION NUMBER	3. RECIPIENT'S CATALOG NUMBER
CA15-2293		

4. TITLE AND SUBTITLE	5. REPORT DATE
Enhancement and Technical Support of Intelligent Roadway Information System (IRIS) in Caltrans Districts 1, 2, 5 and 10	December 31, 2014
	6. PERFORMING ORGANIZATION CODE AHMCT Research Center, UC Davis
7. AUTHOR	8. PERFORMING ORGANIZATION REPORT NO.
Travis Swanston, Kin Yen, Bahram Ravani & Ty Lasky	UCD-ARR-14-12-31-01
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 65A0397, Task ID 2293
	13. TYPE OF REPORT AND PERIOD COVERED Final Report June 2011 – December 2014
12. SPONSORING AGENCY AND ADDRESS California Department of Transportation P.O. Box 942873, MS #83 Sacramento, CA 94273-0001	14. SPONSORING AGENCY CODE Caltrans

15. SUPPLEMENTARY NOTES

16. ABSTRACT

This report documents the research project "Enhancement and Technical Support of Intelligent Roadway Information System (IRIS) in Caltrans Districts 1, 2, 5 and 10," performed under contract 65A0397, Task ID 2293. It presents a current overview of IRIS, and its design and function. The report also documents knowledge transfer for IRIS development from AHMCT researchers to two contractors. This knowledge transfer process included updates and improvements to the static mapping tool chain and the IRIS build process. IRIS provides Advanced Traffic Management System (ATMS) capabilities to Caltrans rural districts. IRIS is currently deployed in Caltrans Districts 1, 2, 5, and 10.

17. KEY WORDS	18. DISTRIBUTION STATEMENT	
Advanced Traffic Management System (ATMS), Traffic operations, Open source	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.	
19. SECURITY CLASSIFICATION *(of this report)*	20. NUMBER OF PAGES	21. COST OF REPORT CHARGED
Unclassified	51	

Reproduction of completed page authorized

DISCLAIMER/DISCLOSURE STATEMENT

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, Innovation and System Information at the California Department of Transportation. It is evolutionary and voluntary. It is a cooperative venture of local, State and Federal governments and universities.

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 accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California, the Federal Highway Administration, or the University of California. This publication does not constitute a standard, specification or regulation. This report does not constitute an endorsement of any product described herein.

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, Innovation and System Information, MS-83, California Department of Transportation, P.O. Box 942873, Sacramento, CA 94273-0001.

Advanced Highway Maintenance and Construction Technology Research Center

Department of Mechanical and Aerospace Engineering
University of California at Davis

Enhancement and Technical Support of Intelligent Roadway Information System (IRIS) in Caltrans Districts 1, 2, 5 and 10

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

Report Number: CA15-2293
AHMCT Research Report: UCD-ARR-14-12-31-01
Final Report of Contract: IA65A0397, Task ID 2293

December 31, 2014

California Department of Transportation

Division of Research, Innovation and System Information

ABSTRACT

This report documents the research project "Enhancement and Technical Support of Intelligent Roadway Information System (IRIS) in Caltrans Districts 1, 2, 5 and 10," performed under contract 65A0397, Task ID 2293. It presents a current overview of IRIS, and its design and function. The report also documents knowledge transfer for IRIS development from AHMCT researchers to two contractors. This knowledge transfer process included updates and improvements to the static mapping tool chain and the IRIS build process. IRIS provides Advanced Traffic Management System (ATMS) capabilities to Caltrans rural districts. IRIS is currently deployed in Caltrans Districts 1, 2, 5, and 10.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

Acronym	Definition
AHMCT	Advanced Highway Maintenance and Construction Technology
API	Application Programming Interface
ASE	Advanced Systems Engineering Consulting
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
CHP	California Highway Patrol
CMS	Changeable Message Sign
COCOMO	Constructive Cost Model
COTS	Commercial-Off-the-Shelf
CWWP	Commercial Wholesale Web Portal
DMS	Dynamic Message Sign
DRISI	Division of Research, Innovation and System Information
EIS	Electronic Integrated Systems
FSR	Feasibility Study Report
GB	Gigabyte
GNU	GNU's Not Unix
GPL	GNU General Public License
HAR	Highway Advisory Radio
HTTP	Hypertext Transport Protocol
IRIS	Intelligent Roadway Information System
JWS	Java Web Start
KML	Keyhole Markup Language
KSD	Knowledge Systems Design, Inc.
KSD	Knowledge Systems Design
LCS	Lane Control Signal
LDAP	Lightweight Directory Access Protocol
MB	Megabyte
MITTENS	Messaging Infrastructure for Travel Time Estimates to a Network of Signs
MnDOT	Minnesota Department of Transportation
MVDS	Microwave Vehicle Detection Station
NDA	Non-Disclosure Agreement
NDOR	Nebraska Department of Roads
OS	Operating System
OSM	OpenStreetMap
PeMS	Performance Measurement System
PTZ	Pan, Tilt, and Zoom
RAM	Random Access Memory
RTMS	Remote Traffic Microwave Sensor
RWIS	Road Weather Information System
SLES	SUSE Linux Enterprise Server
SOCCS	Satellite Operations Center Command System
SQL	Structured Query Language
SSI	Surface Systems, Inc.
SUSE	Software und System-Entwicklung (German for system and software development)
SwRI	Southwest Research Institute

Acronym	Definition
TMC	Transportation Management Center
TMCAL	Transportation Management Center Activity Logging
TMS	Traffic Monitoring Station
UCD	University of California-Davis
URMS	Universal Ramp Metering System
VDS	Vehicle Detection Station
VM	Virtual Machine
WYDOT	Wyoming Department of Transportation
XML	Extensible Markup Language

ACKNOWLEDGMENTS

The authors gratefully acknowledge the Division of Research, Innovation and System Information (DRISI) of Caltrans which has supported this work through the AHMCT Research Center at the University of California-Davis, under contract 65A0397 Task ID 2293, and thank Roya Hassas, Fred Yazdan, and Melissa Clark in particular. The authors also thank Stan Slavin for his continuing commitment and strong guidance to on-going IRIS development and use. The authors thank Doug Lau of MnDOT for his leading work on IRIS and his support for collaborative development. The authors also thank James Kranig of MnDOT for his leading management on IRIS, including its release as an open-source project. The authors are also grateful to Kai Leung and David Wells of Caltrans Headquarters Traffic Operations, and Chris Binger, David Busler, and Brian Sordi of Caltrans Headquarters Information Technology. From District 1, the authors thank Jim Sandford, John Carson, and Joe Dower. From District 2, the authors thank Ian Turnbull, Clint Burkenpas, Keith Koeppen, and Joe Baltazar. From District 5, the authors thank Sherwyn Gilliland and Steven Gee. From District 10, the authors thank Mohammad Battah, Veronica Cipponeri, Arlene Cordero, Toni Moon, John Castro, Wilmar Kuhl, and Joe Silvey. The authors also thank Michael Darter and Karl Petty of Berkeley Transportation Systems, and Sik Shum, Wee-Meng Tan, and John Keith of KSD. The authors are also grateful to Dan Rossiter and Tucker Brown of the Southwest Research Institute. The authors also thank Stephen Donecker and Kin Yen of the AHMCT Research Center for their assistance. The authors would like to send a special thanks to all of the dispatchers at the District 10 TMC, some of whom have been with the IRIS development since 2008, as they worked with IRIS and provided much of the info needed for feature requests and updates: Antoinette Moon, George Anzo, James Westphal, Tina Nunes, Lyn Serpa, Beth Robinson, Kit Sherlock, Rudy Anzo, Rachel George, Ingrid Donaville, John Ragusa, Laura Williams, Brenda Threadgill, Tami Harrison, Gracie Munoz, Stephany Kirkpatrick, Angela Duhart, Josh Neal, Kim Berry, and Pam Matedne. 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 [2], the Advanced Highway Maintenance and Construction Technology (AHMCT) Research Center tailored and extended the open-source Intelligent Roadway Information System (IRIS) originally (and currently) developed by the Minnesota Department of Transportation (MnDOT). This research was successfully demonstrated and deployed in District 10, Stockton as a Proof of Concept project.

In a Phase II effort, AHMCT next provided low-level deployment support (maintenance, software patches, and technical support) for IRIS in District 10 [1]. Additional districts (District 1-Eureka, District 2-Redding, 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. At this time, full IRIS capability, including video, for the added districts, required device driver development beyond the resources available for the Phase II research. In addition, until the FSR process was completed, these added districts were required to remain in the testing mode for CMS.

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

The current Phase III research followed the indications of the successful FSR produced during Phase II. This led to development of IRIS Release 9.3, and its full deployment in Districts 1, 2, 5, and 10. This was followed by a 9.4 maintenance release. Then, as a key part of the current effort, the Caltrans IRIS source code was merged with the MnDOT source tree in order to bring many needed new features and bug fixes into Caltrans IRIS. This new, merged tree formed the basis of IRIS Release 10.0.0. Now, an effort is underway to submit these changesets to MnDOT so that they can be included in the official MnDOT code. As of this report, the Caltrans IRIS source code has 66 outstanding changesets comprising approximately 17 functional enhancements waiting for acceptance by MnDOT. Fully integrating these changesets into the official MnDOT code will be accomplished by the Southwest Research Institute (SwRI) in a pending 10.0.1 release. This merge and acceptance process is a key component in each release, and is discussed in detail herein.

Most urban Advanced Transportation Management System (ATMS) programs are typically not well-suited to rural districts, in terms of both features and cost. 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. IRIS now meets that need for Caltrans.

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

Research Approach

The research tasks were as follows:

- IRIS technical support for Caltrans districts
- IRIS enhancements for Caltrans districts
- Software engineering process for IRIS ticket requests from Districts 1, 2, 5, and 10, subject to Change Control Board (CCB) guidance.

Overview of Research Results and Benefits

First and foremost, deployment of IRIS to Districts 1, 2, 5, and 10 was successful. Each of the districts has fully deployed IRIS in their districts. Secondly, the number of traffic management software applications and servers in the four districts has been reduced, with IRIS assuming their roles. Finally, through knowledge transfer from the research team, the IRIS support role is successfully transitioning to a third-party contractor, Southwest Research Institute (SwRI).

IRIS is a platform that:

- Provides cost-effective ATMS capabilities for rural districts
- Is extensible, reliable, and scalable, with support for a diverse set of field elements
- Reduces life cycle costs by approximately 72% versus the existing ATMS [2]
- Shows the importance and benefit 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.

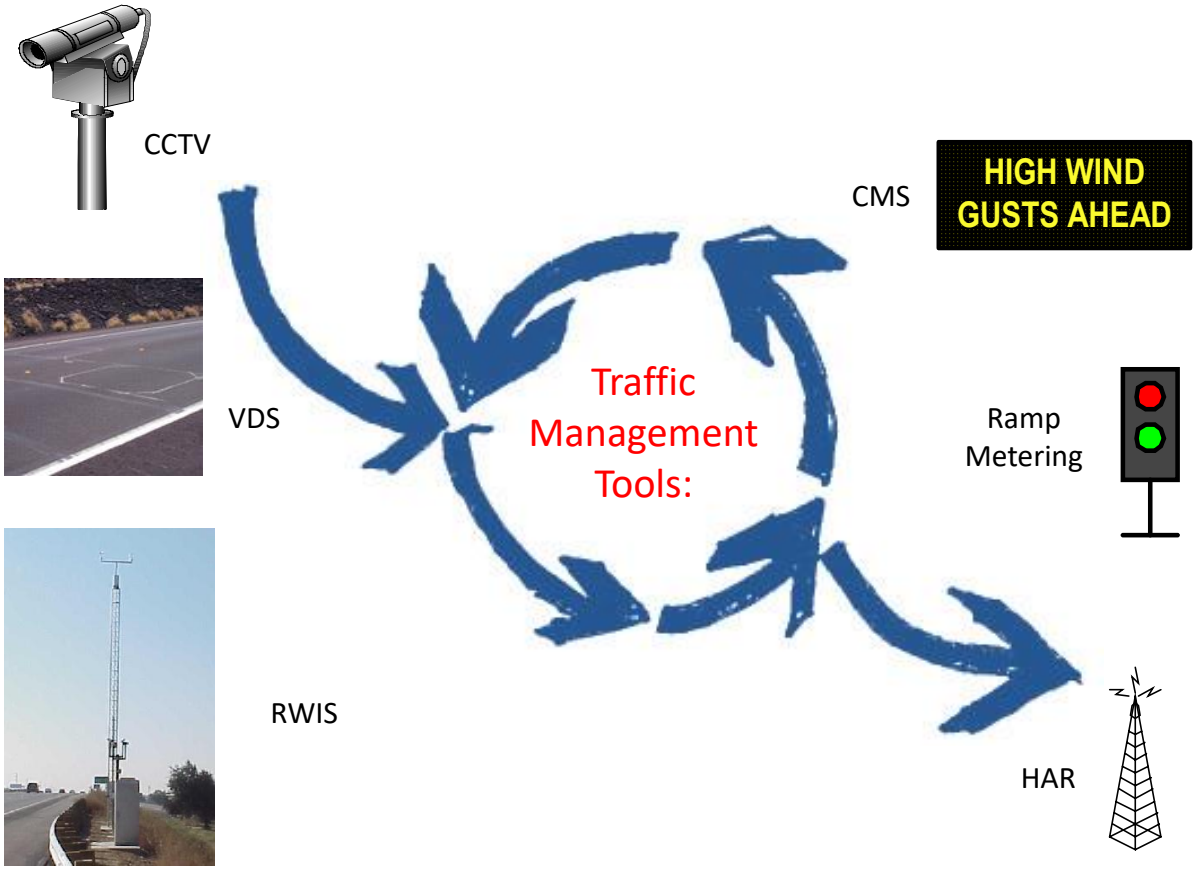

Figure 2.1: ATMS real-time functions

IRIS was developed by the Minnesota Department of Transportation (MnDOT), with development started in the early 1990s. It is used in Minneapolis/St. Paul, St. Cloud, and Rochester. In terms of infrastructure, MnDOT uses IRIS to manage approximately 135 DMS

(Dynamic Message Sign[1]), 476 cameras, 5452 VDS (Vehicle Detection Station), 433 ramp meters, 4 RWIS (Road Weather Information System), 194 LCS (Lane Controll Signal[2]), 1 Lane Marking (in-road lighting), and 2 static signs with wig-wag beacons. IRIS was 100% developed in-house, and represents a significant MnDOT investment (COCOMO[3]: > $4 million).

MnDOT released IRIS as open-source software under the GNU General Public License (GPL) in May 2007. At least four agencies are using or evaluating IRIS. Agencies using IRIS include MnDOT, Caltrans, and Wyoming DOT (WYDOT, for at least DMS and cameras). Agencies evaluating IRIS include Nebraska Department of Roads (NDOR) and the City of Bloomington, Minnesota. MnDOT's motivations for open-sourcing IRIS included:

- Ensuring affordable and manageable longevity of IRIS
- Fostering collaboration with other transportation agencies
- Getting source contributions back to IRIS
- Cultivating additional IRIS developers
- Lowering risks.

IRIS has provided ATMS capabilities to Caltrans rural districts. It is now deployed in Districts 1, 2, 5, and 10. A sample IRIS CMS[4] control operating in Stockton is shown in Figure 2.2.

[1] California uses the term Changeable Message Sign (CMS)
[2] In California, LCS = Lane Closure System
[3] Constructive Cost Model
[4] Minnesota and many other states refer to CMS as Dynamic Message Sign (DMS)

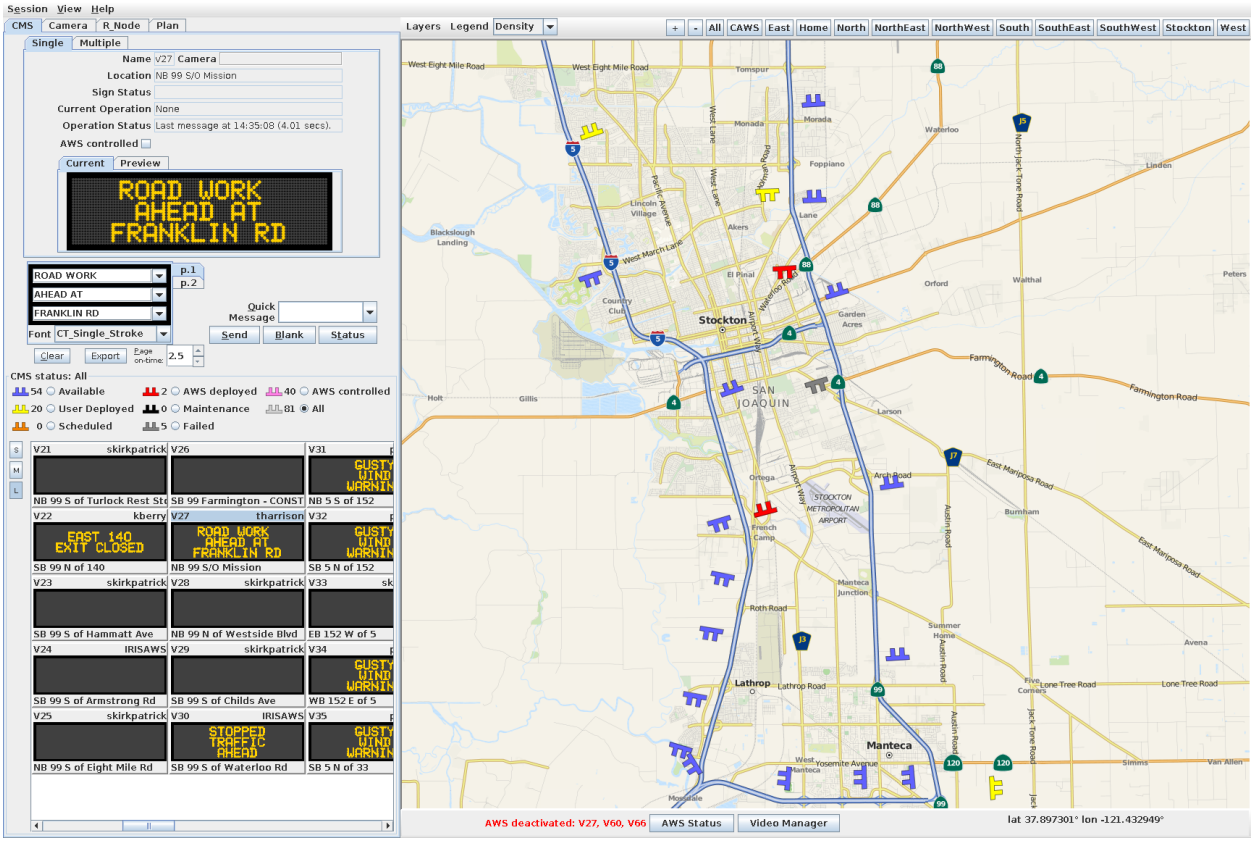

Figure 2.2: IRIS CMS functionality

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

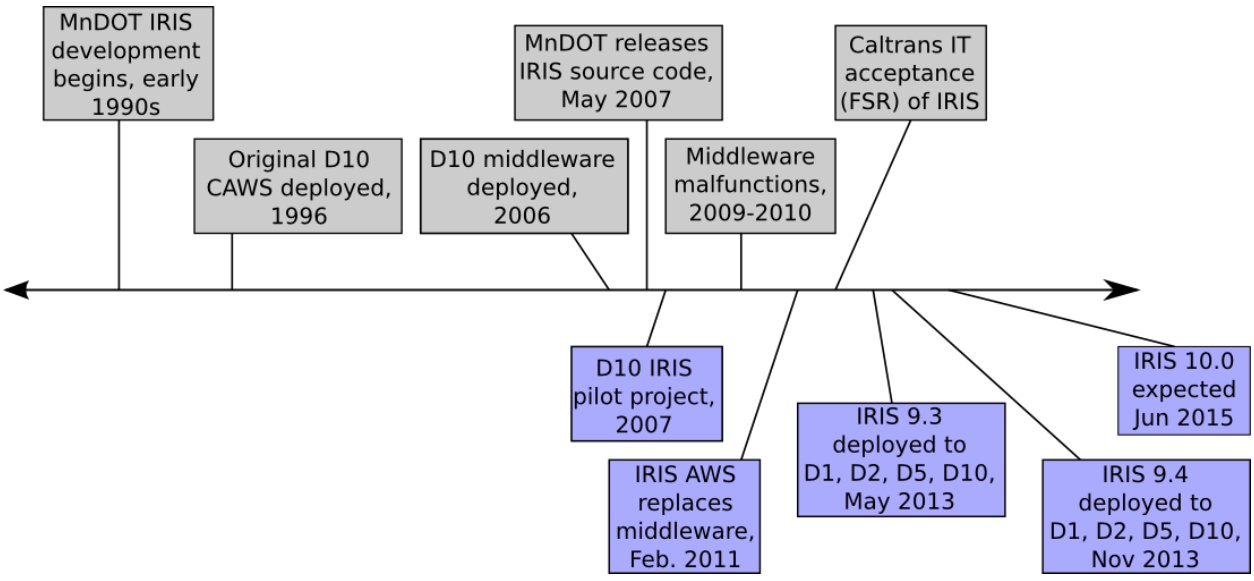

Figure 2.3: Caltrans IRIS timeline

As an illustration of part of the benefit of IRIS, Figure 2.4 shows the Caltrans District 10 architecture before and after IRIS deployment. In this image, the red hash marks illustrate the elements IRIS has replaced. There is no longer a need for multiple separate systems including the Traffic Relay Server, CAWS (Caltrans Automated Warning System) Middleware, and Stand-alone DMS (Dynamic Message Sign) Server SOCCS (Satellite Operations Center Command System). Note that middleware remains for RWIS. Table 2.1 summarizes key changes.

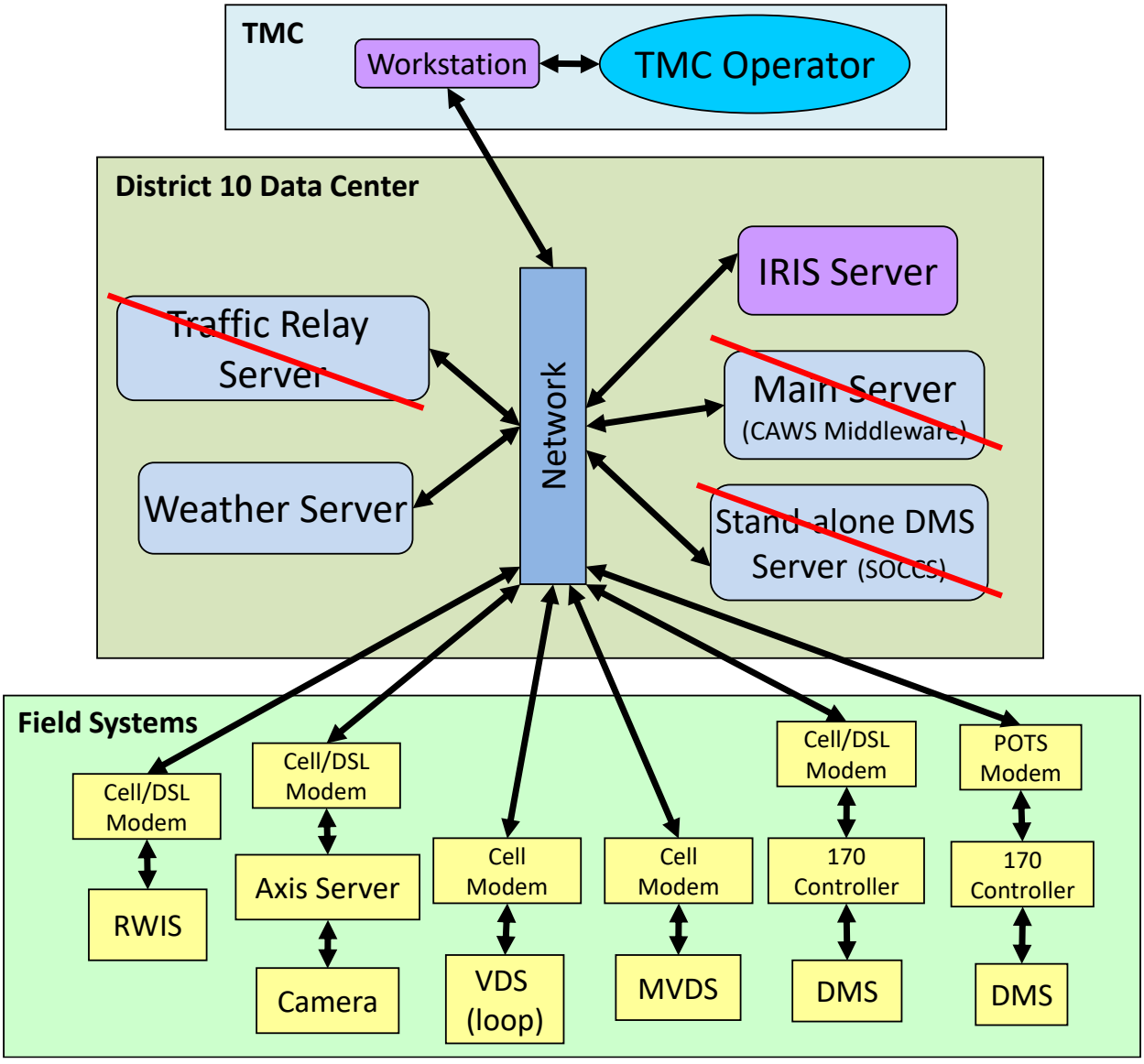

Figure 2.4: Caltrans District 10 architecture before and after IRIS

Table 2.1: Benefits of IRIS: Before and after comparison

	Before	After
Mapping	None	✓
Number of servers	4	1
Number of CMS controlled by Automated Warning System (AWS)	9	42 (unlimited*)
Types of VDS supported (Microwave Vehicle Detection Station (MVDS), loops, etc.)	1	7+ (unlimited*)
Types of RWIS supported (Manufacturers)	1	3+ (unlimited*)
Types of CMS controllers supported (Manufacturers)	1	4+ (unlimited*)
Source code availability	None or proprietary	✓
Standby/backup VM snapshots	None	✓
		* via device driver interface

Through this joint effort, AHMCT has enabled Caltrans to add the following features that are now available to partnering agencies, i.e. MnDOT:

- Device drivers
  - VDS: MVDS (EIS RTMS), URMS 2070, Wizard, Sensys AP240
  - RWIS (SSI, CWWP XML (Extensible Markup Language))
  - Performance Measurement System (PeMS)
- Automated Warning System (AWS)
- Testing
  - Automated unit test cases
  - CMS simulation
- Generalization of IRIS, e.g. system attributes
- CMS message library
- Google Earth output (KML)
- RWIS map integration.

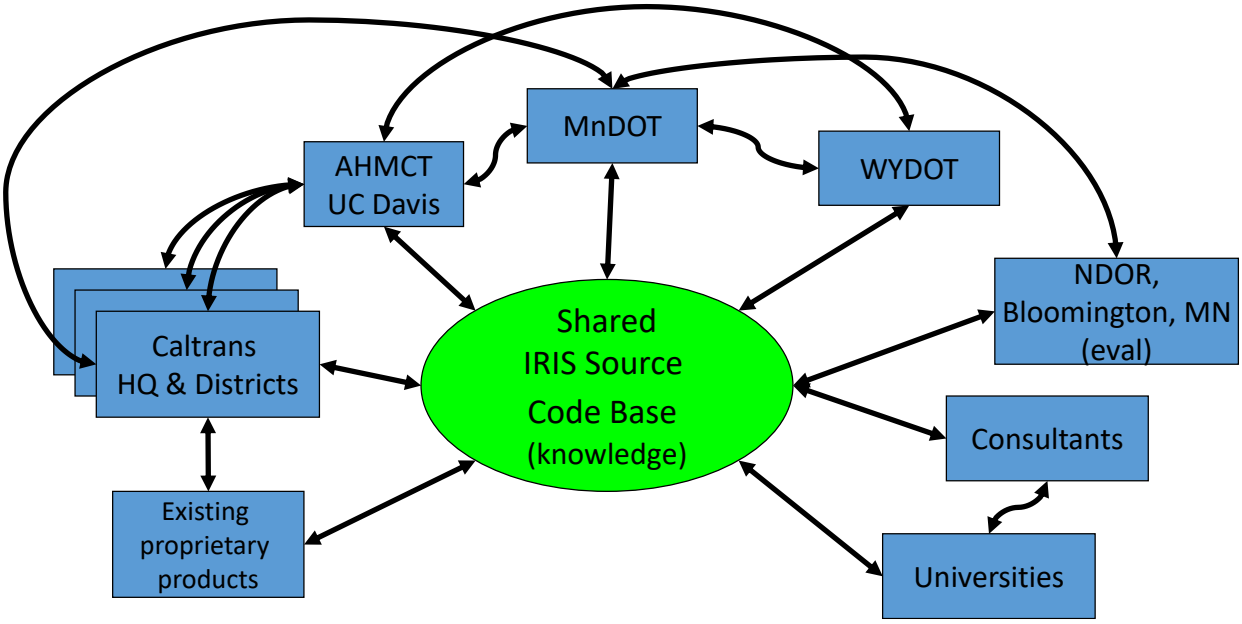

Figure 2.5: Multi-agency collaboration

Although less tangible, an important outcome of this research is the shared collaborative development model between Caltrans, MnDOT, AHMCT, Knowledge Systems Design, Inc. (KSD), Southwest Research Institute (SwRI), and others (Figure 2.5). Recently, the collaboration has included consultants at the system engineering level (Advanced Systems Engineering Consulting, ASE), and, under the current phase effort, at the design, coding, test, and support level (KSD and SwRI).

Figure 2.6 summarizes how the Caltrans version of IRIS is managed. While a revision control system (Mercurial) is used during the development of Caltrans IRIS enhancements, the final versions of these enhancements are stored and managed as patch files, rather than via a repository. This scheme has evolved as a result of our current working arrangement with MnDOT, who requested that we submit our changes to them as patch files rather than as repository pull requests. An alternative approach to storing the Caltrans changes as patch files might be to maintain a separate branch for each enhancement, but given the current number of extant Caltrans enhancements, this approach could prove cumbersome in practice.

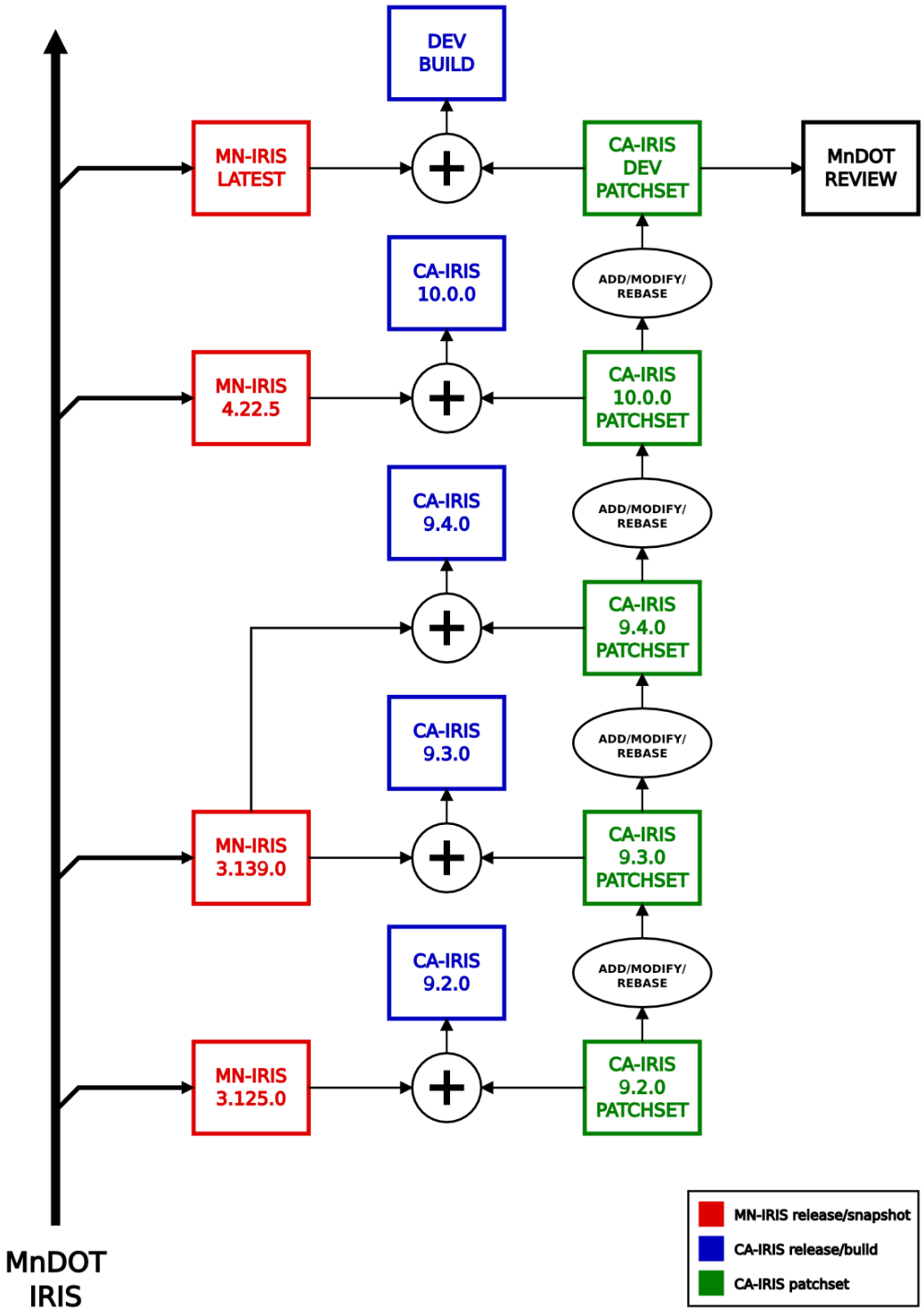

Figure 2.6: Caltrans IRIS source code management workflow

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 (e.g. Java, PostgreSQL, Apache HTTP (Hypertext Transport Protocol), Apache Tomcat, Apache Ant) are open-source
    - Free, no purchase requisition forms
    - No non-disclosure agreement (NDA) required
- IRIS is written in the Java programming language
    - ~350K lines of code
    - IRIS's design is heavily object-oriented, with ~3200 classes. The code is relatively complex, yet well-organized.
    - The IRIS build environment is Linux-based, and utilizes Java, Apache Ant, and the Mercurial distributed version control system. IRIS may be transitioning to Git. Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Git was initially designed and developed by Linus Torvalds for Linux kernel development in 2005, and has since become the most widely adopted version control system for software development[5].
- 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 closed-circuit TV (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 reports, CMS XML feeds, log management, etc.
    - Server-side requirements include Java, Apache HTTP (web server), Apache Tomcat (web server and servlet container), and the PostgreSQL relational database (a.k.a. "Postgres"), all of which are free, open-source technologies.

[5] https://en.wikipedia.org/wiki/Git_(software)

  - IRIS can optionally use LDAP (Lightweight Directory Access Protocol) to authenticate users based on their Caltrans credentials.

- 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 should 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 the start of this phase of the research, IRIS was well established in District 10, and 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 the current project to deploy this system to other districts. The focus was therefore shifted to deploying IRIS elsewhere in Caltrans to Districts 1, 2, and 5. With this full-fledged project now underway, having started in June 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 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. Due to the lack of an adequate transition period with the previous developer, the project suffered several months of delay.

- Managing ticket priorities: to defer or fix bugs? It can be difficult to properly allocate development resources 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. The current project was focused on maintenance instead of new features or major enhancements. Great care needs to be taken in estimating level of effort and managing customers' expectations for enhancements.

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

- Under previous phase efforts, 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.

- Scarcity of mid- and low-level technical documentation:
  - Software is documented using Javadoc comments within the "classes" and "packages" of the source code. This is very helpful to developers, but not sufficient.
  - There is no system architecture document
  - There are no data flow diagrams
  - There are no detailed software or interface design documents
  - Existing installation/administration/maintenance documentation is lacking in some areas.

- Differing approaches to documentation by participating agencies. Caltrans has been working towards upfront detailed design. MnDOT has taken a more dynamic but less detailed approach that does not match well with the current Caltrans approach. In an open-source effort such as IRIS, highly detailed, upfront documentation efforts can be rendered obsolete quite easily as one party modifies related code in a manner inconsistent with existing documentation. This is not intended as a criticism of Caltrans' or MnDOT's approach. Rather, it is a fundamental characteristic of open-source development. Developing a consistent and effective approach to documentation for an open-source project with multiple contributors is a challenging issue, and approaches that can be quite effective for monolithic efforts may not be successful for open-source development. There are effective models available in the open-source community.

- 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 an enormous amount of developer time in the longer-term
  - Code merges are voluntary (by both AHMCT and MnDOT). 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 of 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-centric branches is highly recommended.
- Excellent generalized designs take additional effort, as support must be maintained for operation modes that, though not used by Caltrans, may be used by another agency.
- Communicating how the open-source process is different:
    - The 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
- During this project phase, IRIS has been successfully put into production use in Caltrans Districts 1, 2, and 5, bringing the current number of districts using IRIS to four
- The application scope of IRIS has been successfully extended to fulfill new district requirements and feature requests
    - Developed new features (e.g., video wall control, CCTV connection management, external video viewer support, CCTV "return home" feature, new CMS feed, CMS composer enhancements, many new options to manage CCTV usage, customized site naming and location description formatting)
    - Developed new device drivers (e.g., Cohu PTZ, Axis PTZ, Sensys AP240, Axis decoder, RTMS G4 backport)
    - Enhanced reporting (e.g., status reports, monthly CMS activation reports)

  - Generated a custom California map to meet the specific requirements of the districts using IRIS.

- The complex build and deployment process for the Caltrans IRIS implementation has been greatly improved and automated, and a number of tools have been developed to manage it. This significantly eases the burden of training new developers.

- IRIS administration and development knowledge transfer to the incoming IRIS maintenance contractor (SwRI) has been good, and SwRI has already begun to perform IRIS builds in their development environment. SwRI began its maintenance work on IRIS in July 2014, and is assuming full responsibility for IRIS maintenance through June 2017 as of July 1, 2015.

- Under a previous project phase, 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 and user interface

  - Forwards traffic to PeMS (Performance Measurement System)

  - Developed AWS module

  - Testing and verification

- Reliability and code quality of IRIS are good. IRIS is quite robust

- Low cost, less than one-fourth the cost of proprietary ATMS [2].

IRIS Functional Areas

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

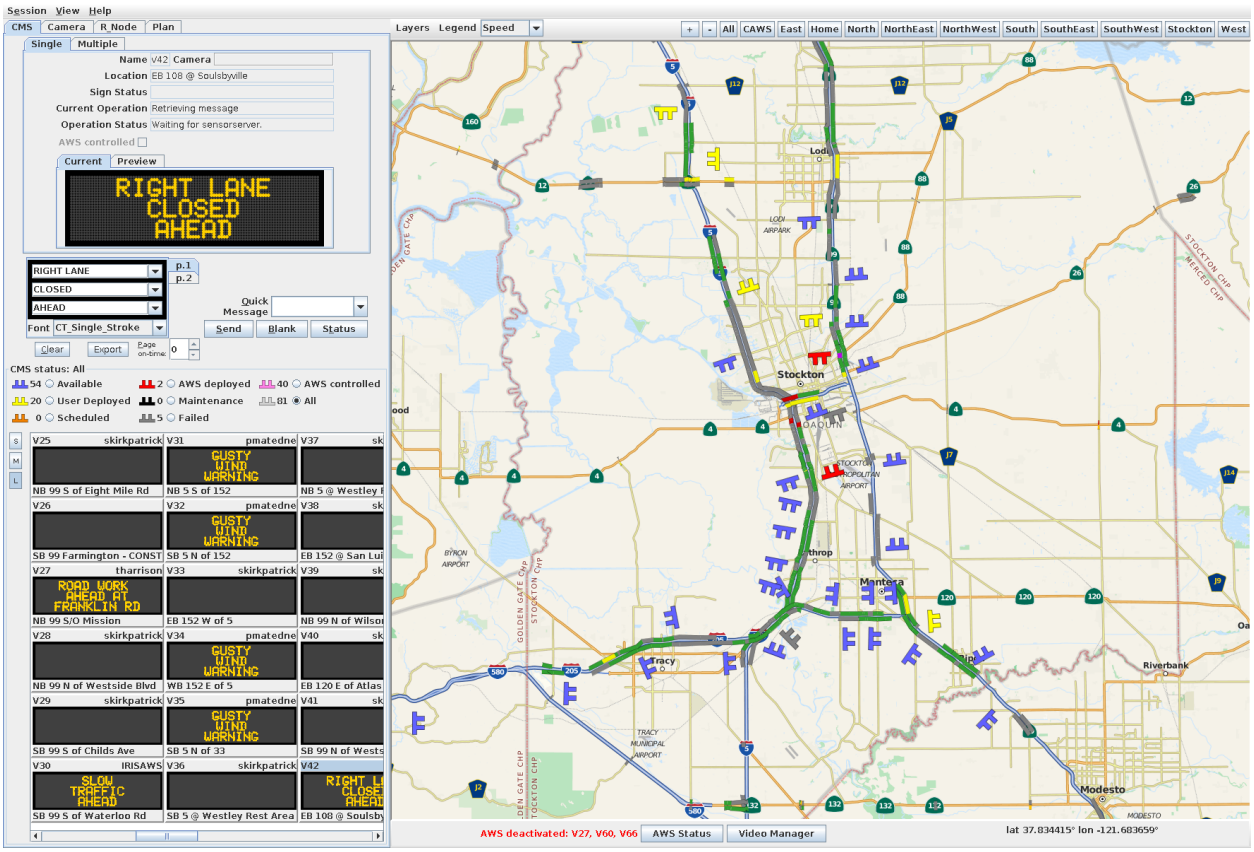

Figure 3.1: IRIS CMS control and monitoring

Figure 3.1 provides a view of the IRIS client, specifically the interface for CMS control and monitoring. The map (right hand side) was substantially upgraded in Release 9.3. The map shows sign locations and color-coded status, and the currently selected 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.

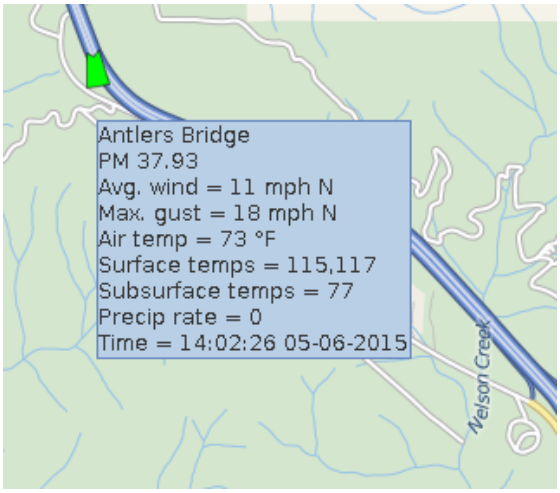

Figure 3.2: RWIS map icon and popup window

The Road Weather Information System (RWIS) is shown in Figure 3.2. When the user hovers the pointer over an RWIS icon, a popup window appears displaying 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.

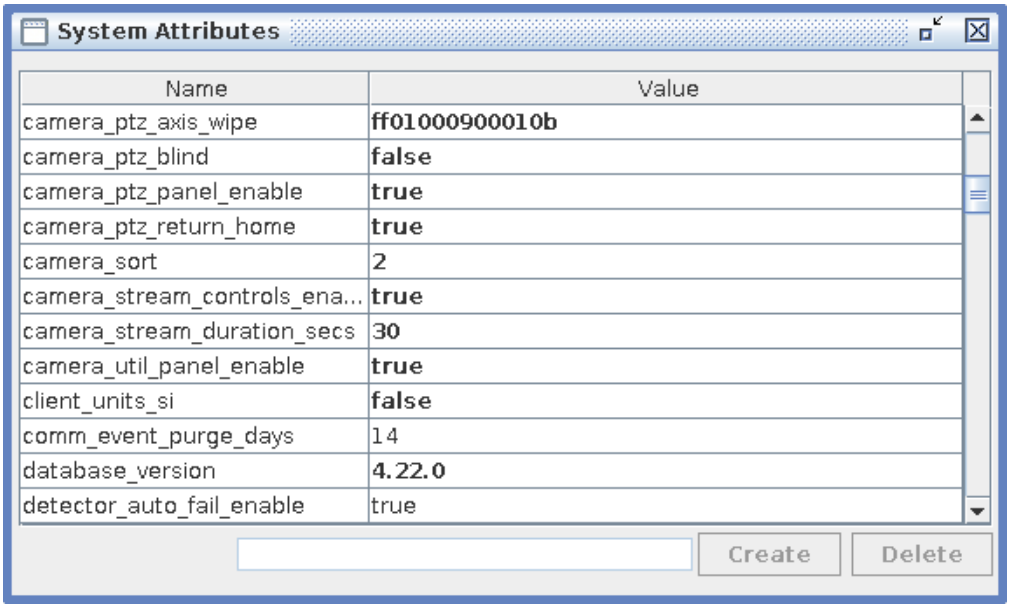

Figure 3.3: System attributes editor

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 starts

independent development on it, creating a distinct piece of software.[6] Forking is extremely undesirable.

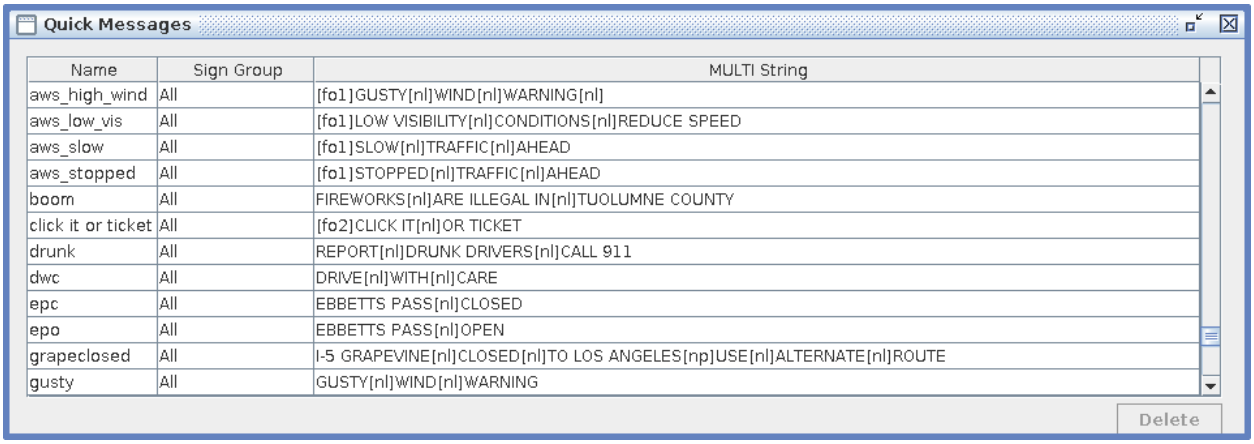

Figure 3.4: CMS message library

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.

[6] See http://en.wikipedia.org/wiki/Fork_%28software_development%29

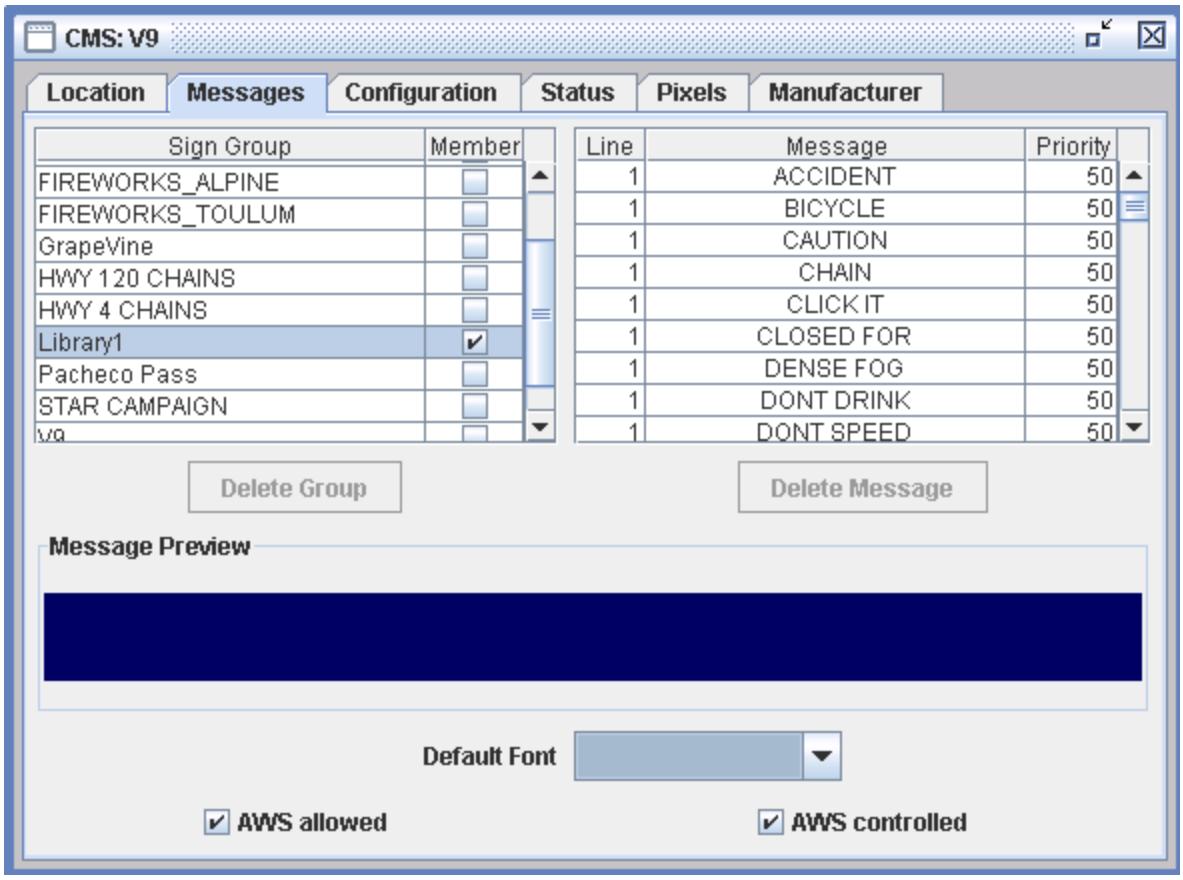

Figure 3.5: CMS definition

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.

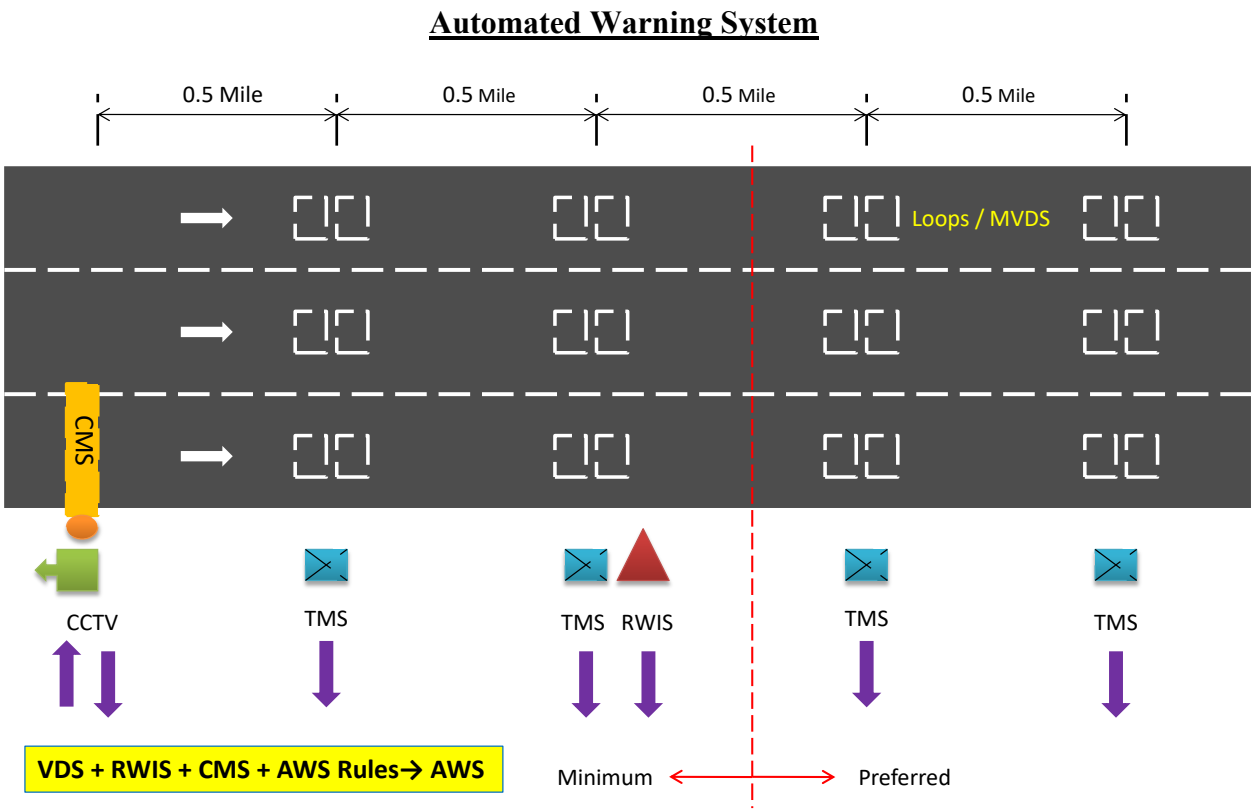

Figure 3.6: Typical Automated Warning System (AWS) configuration

Figure 3.6 provides a typical Automated Warning System (AWS) configuration. Here, TMS is a Traffic Monitoring Station. Under prior phase research, IRIS has assumed the full AWS role 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 four 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 December 2014, there were 7,344 CMS messages deployed in District 10. 81% (5,985) of these messages were automatically deployed by the IRIS AWS module, while 19% (1,359) of these messages were manually deployed by TMC operators.

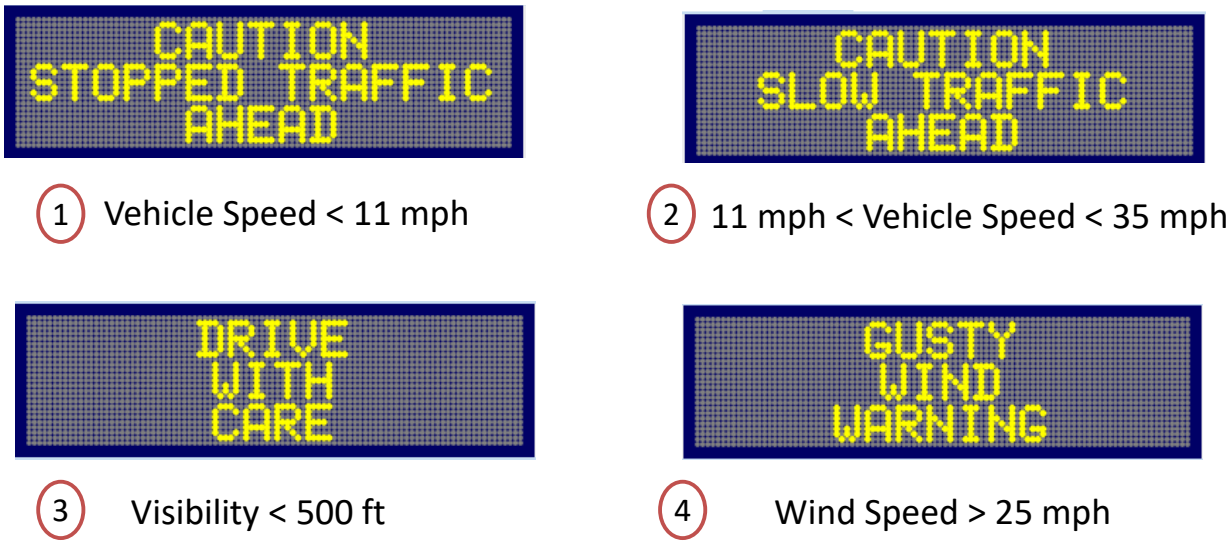

Figure 3.7: AWS message triggers

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

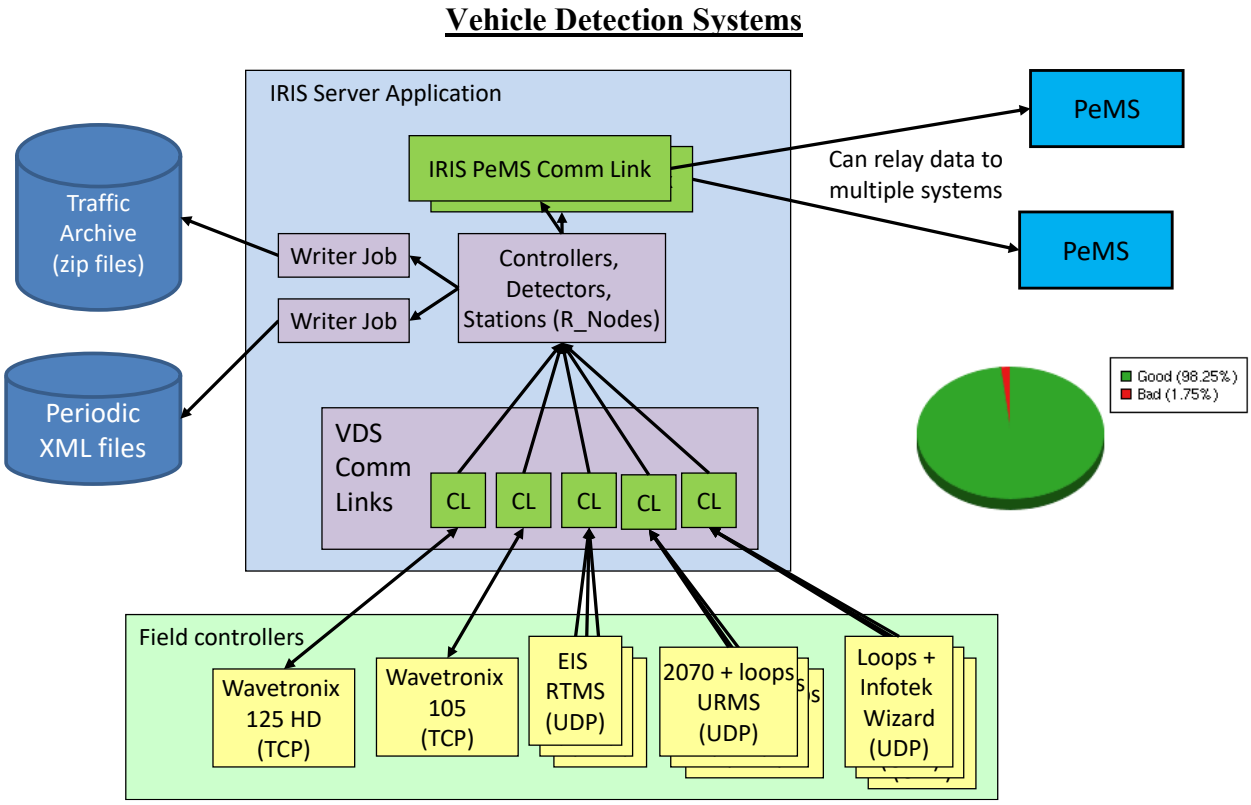

Figure 3.8: IRIS traffic data collection open-source data acquisition: Field→IRIS→PeMS

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

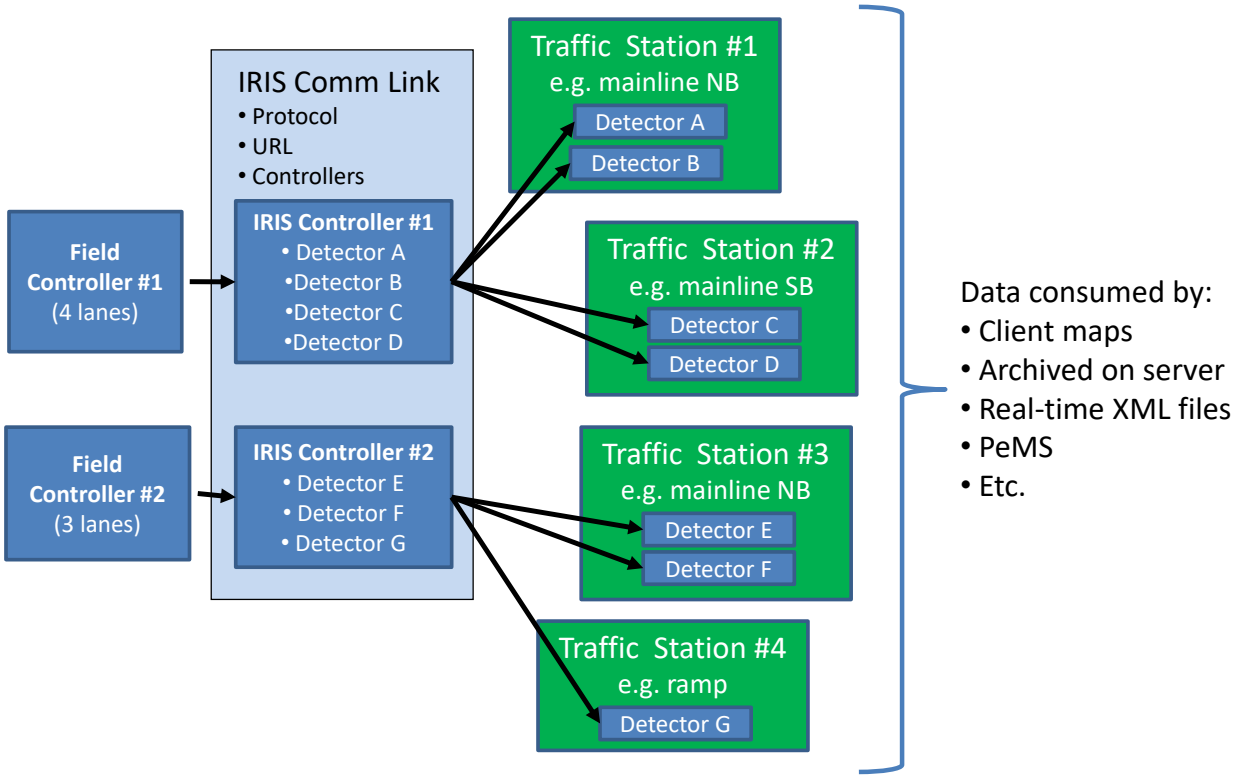

Figure 3.9: VDS configuration

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

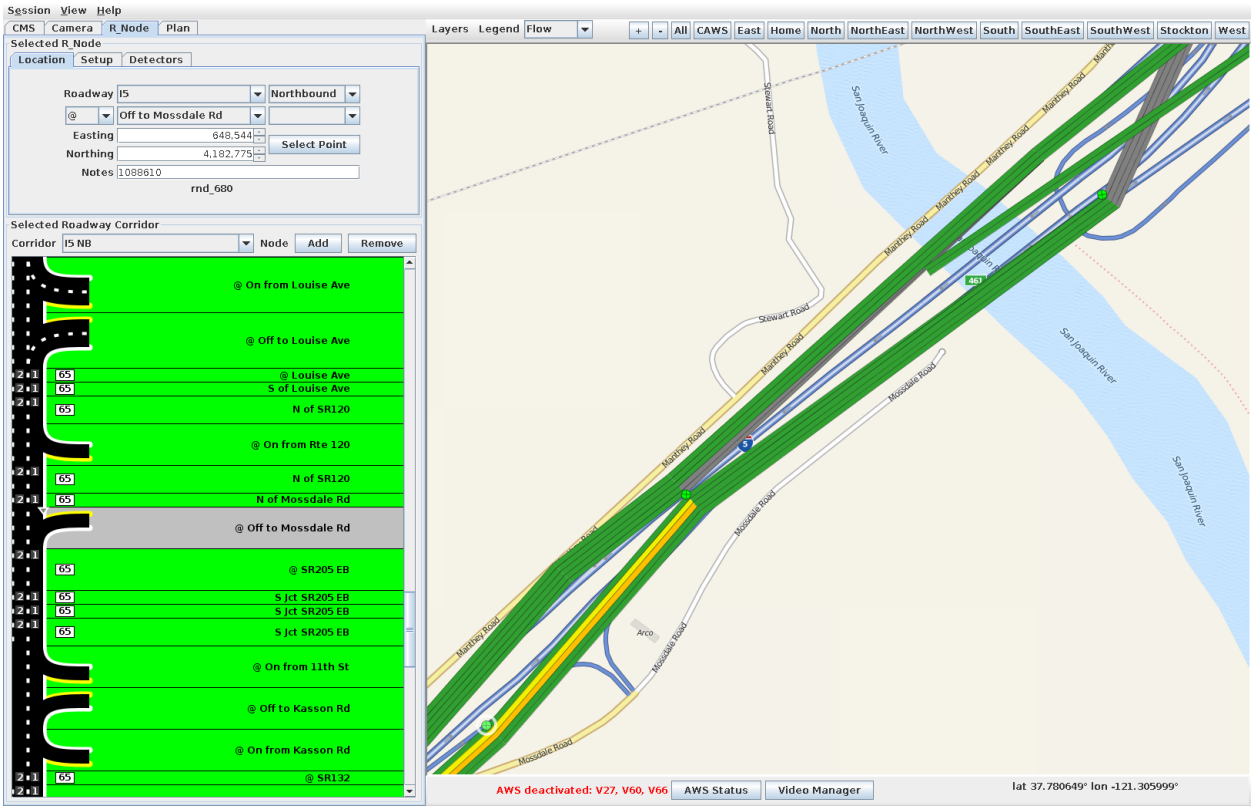

Figure 3.10: R_node definition and editor

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 communication problems

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

Data Flow and Architecture

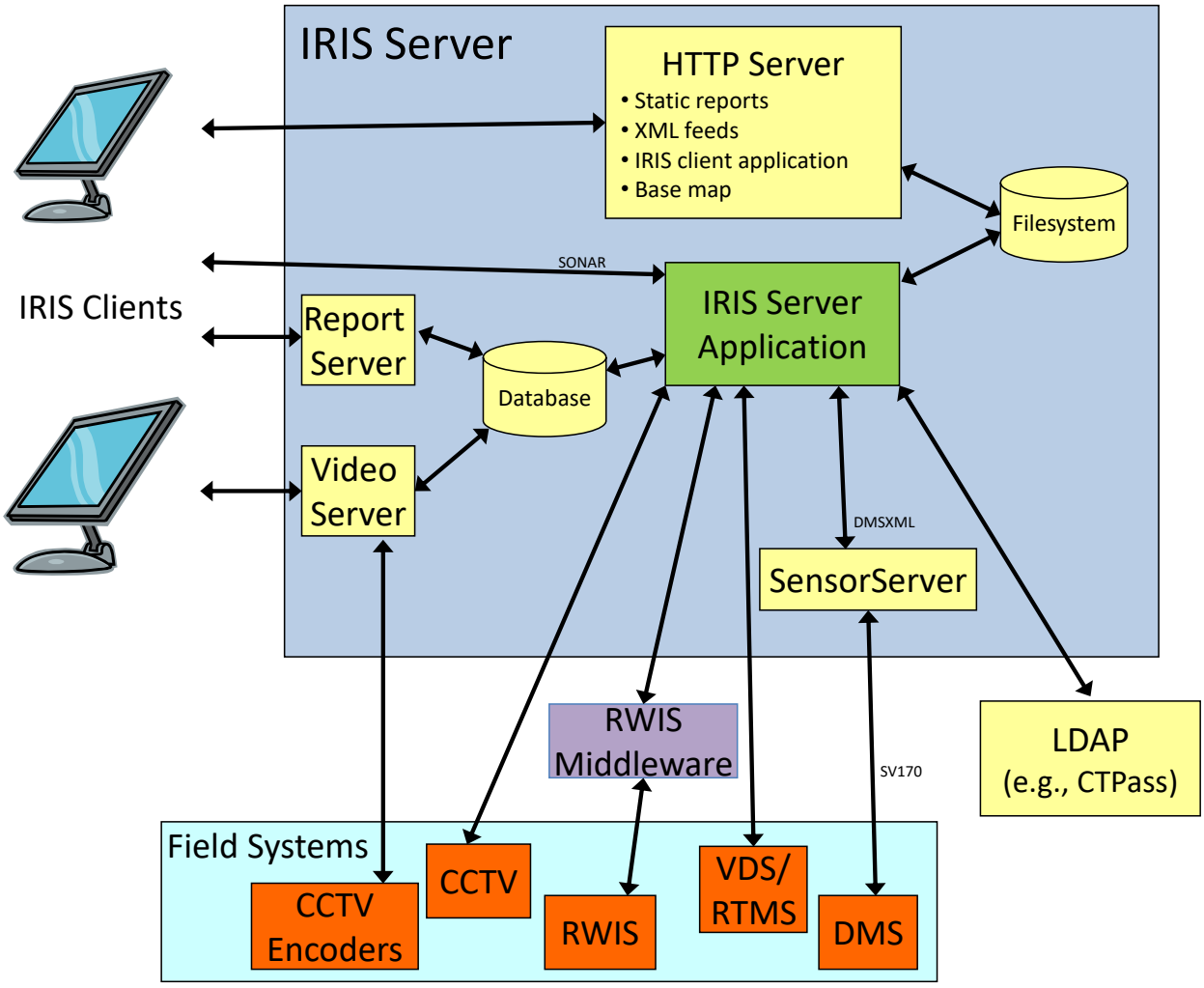

Figure 3.12: General data flow in an IRIS deployment

Figure 3.12 summarizes the general data flow in an IRIS ATMS deployment, including field systems and external systems (remaining RWIS middleware, LDAP).

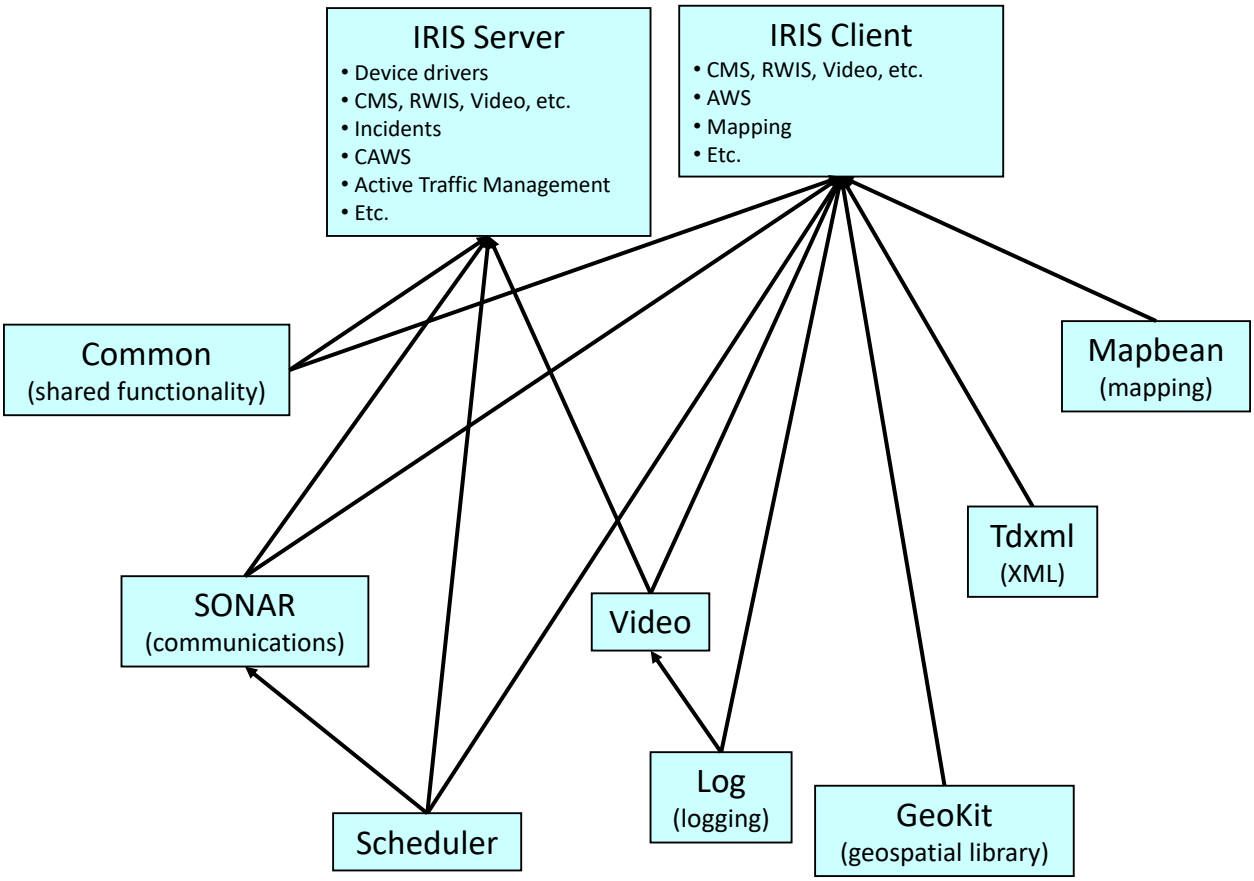

Figure 3.13: IRIS binary modules

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

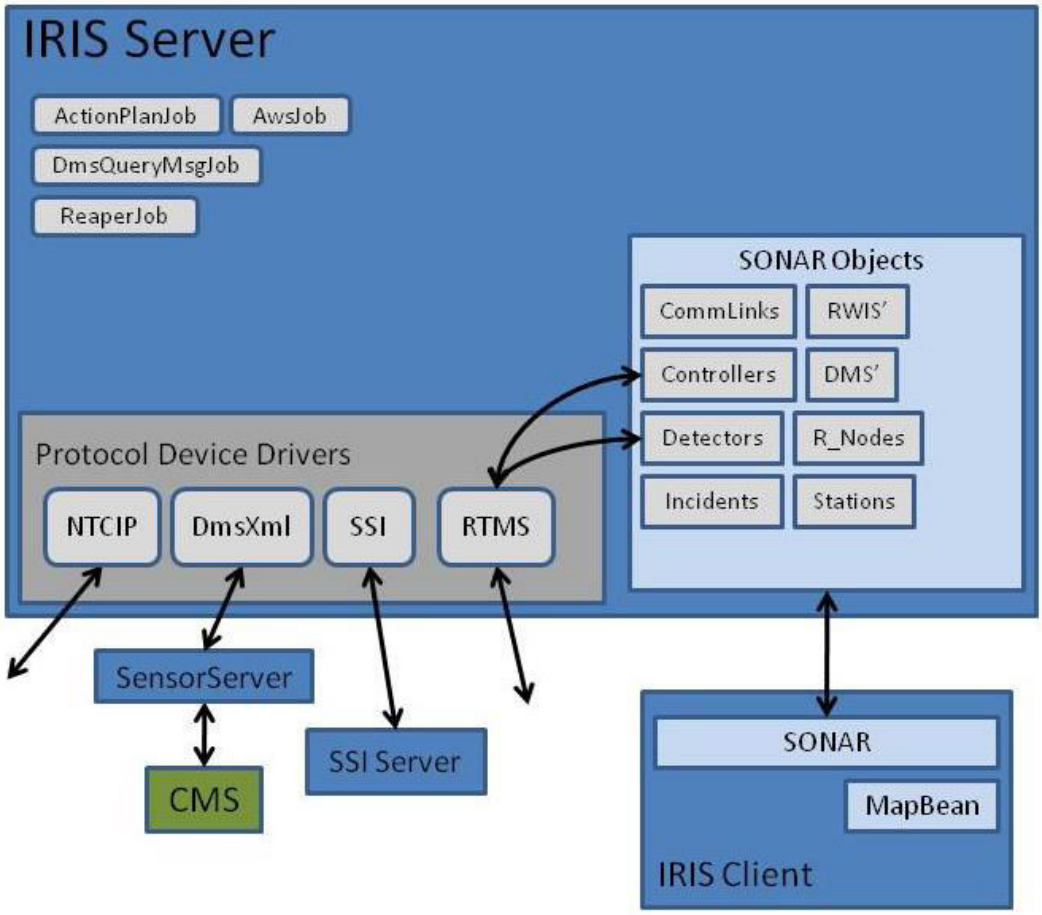

Figure 3.14: IRIS server architecture

Figure 3.14 illustrates the IRIS server architecture, and its connections with the IRIS client, SensorServer via DMSXML for CMS, and RWIS middleware (e.g., SSI server) for weather.

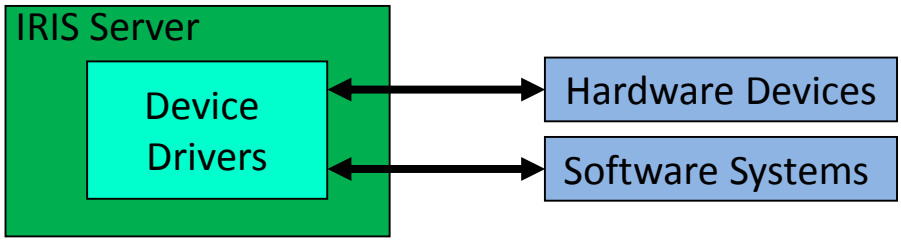

Figure 3.15: IRIS architecture + design: Protocol device drivers

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, EIS RTMS G4, URMS, Wizard, Canoga, Sensys AP240
- CMS / DMS: NTCIP A, B, C, SV170
- Video: Pelco D PTZ, Pelco switcher, Vicon PTZ, Vicon switcher, Manchester PTZ, Cohu PTZ, Axis PTZ, Axis Decoders,
- RWIS: Optical Scientific ORG-815, SSI, CWWP XML
- PeMS.

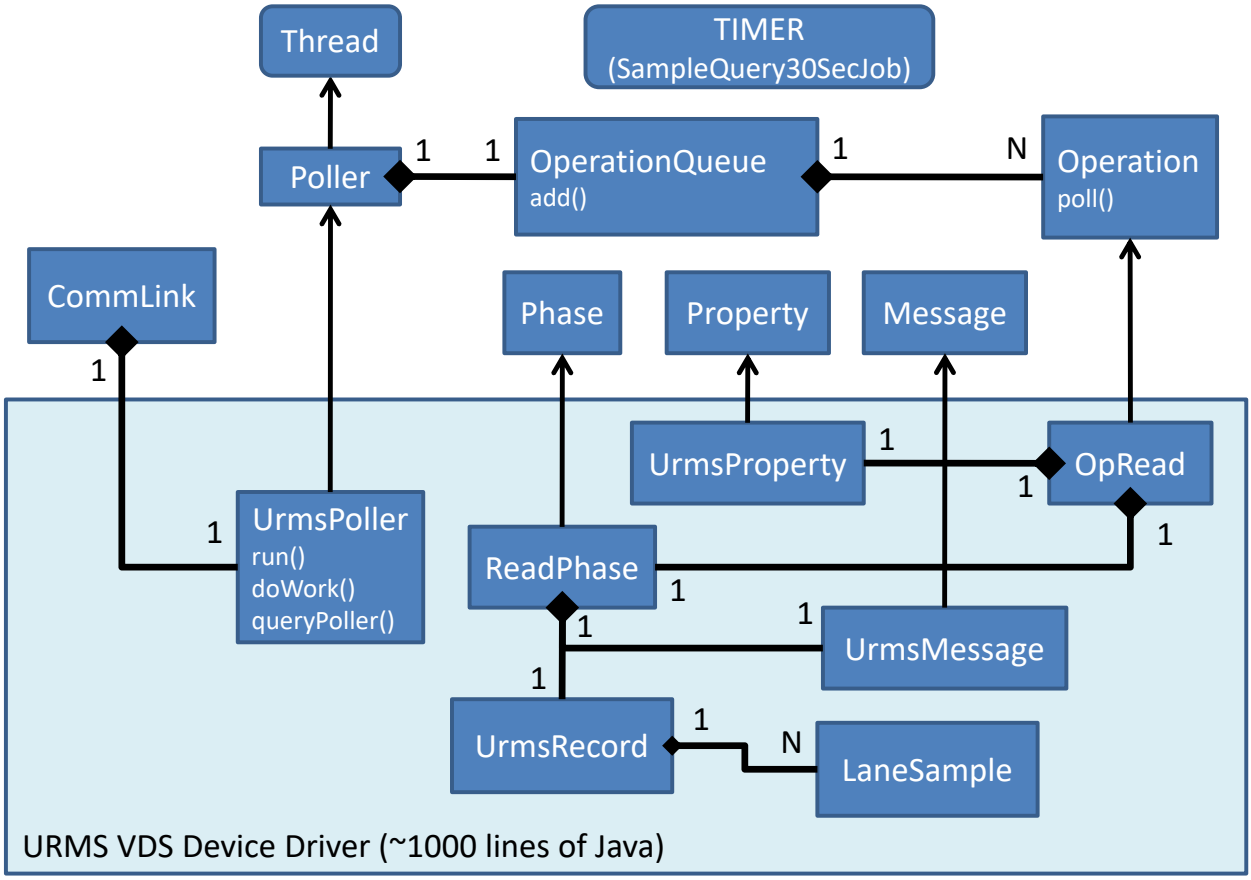

Figure 3.16: IRIS class structure for URMS device driver

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

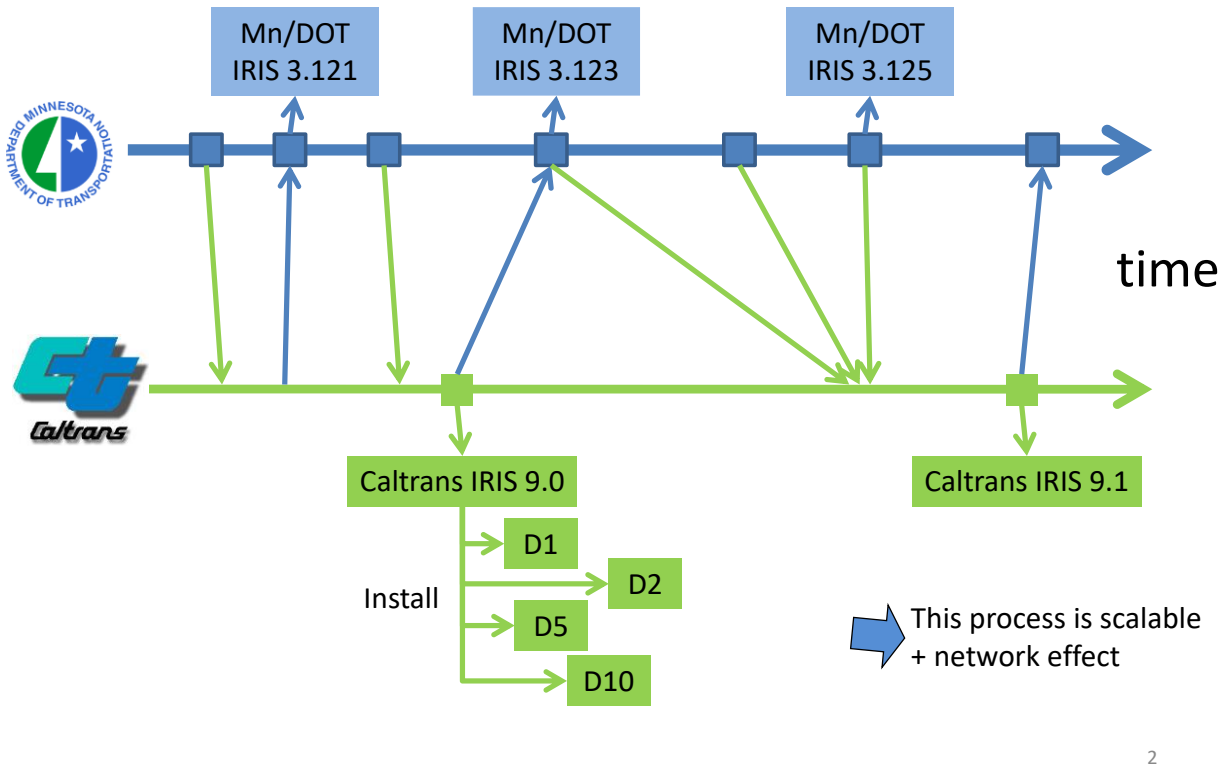

Figure 3.17: Ideal IRIS release and collaboration process

Figure 3.17 illustrates the ideal collaboration process between MnDOT and Caltrans. The figure details are for Caltrans IRIS 9.1, which was released in November, 2008. It does not represent the current version status in MnDOT or Caltrans; rather, it is meant to illustrate the general concept. The goal is to keep the California code fork as small as feasible at all times.

Video

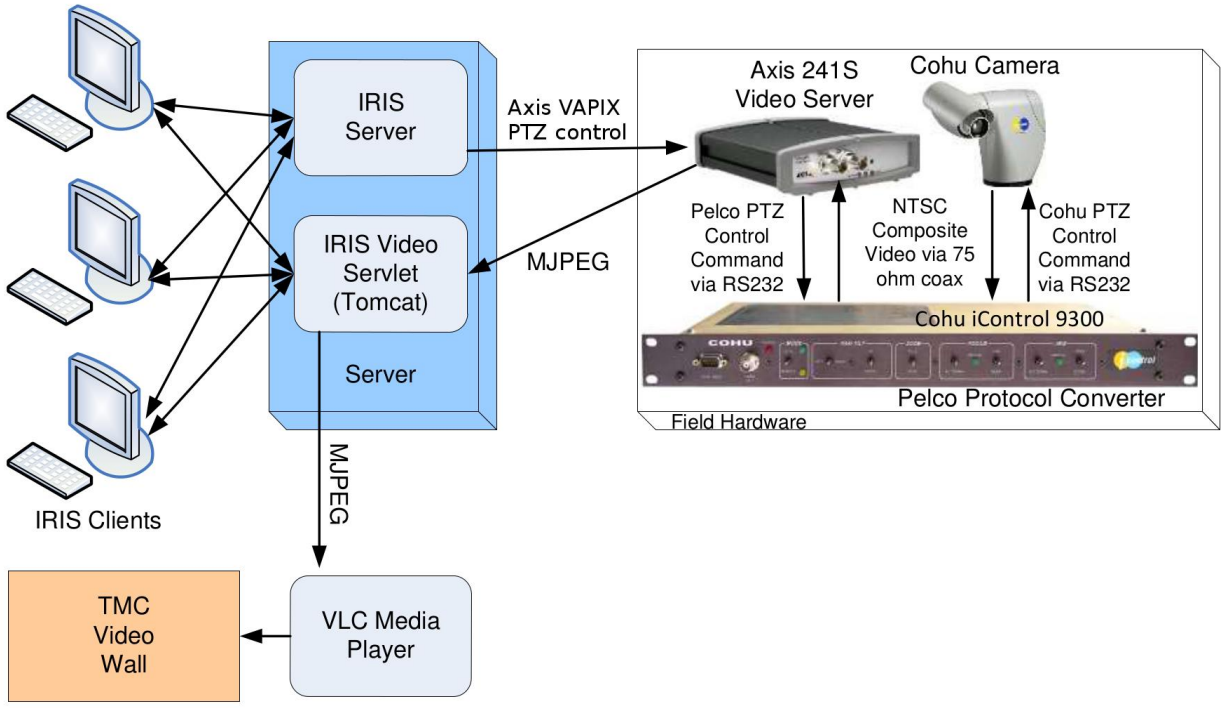

Figure 3.18: D10 video architecture

Figure 3.18 provides the District 10 video architecture.

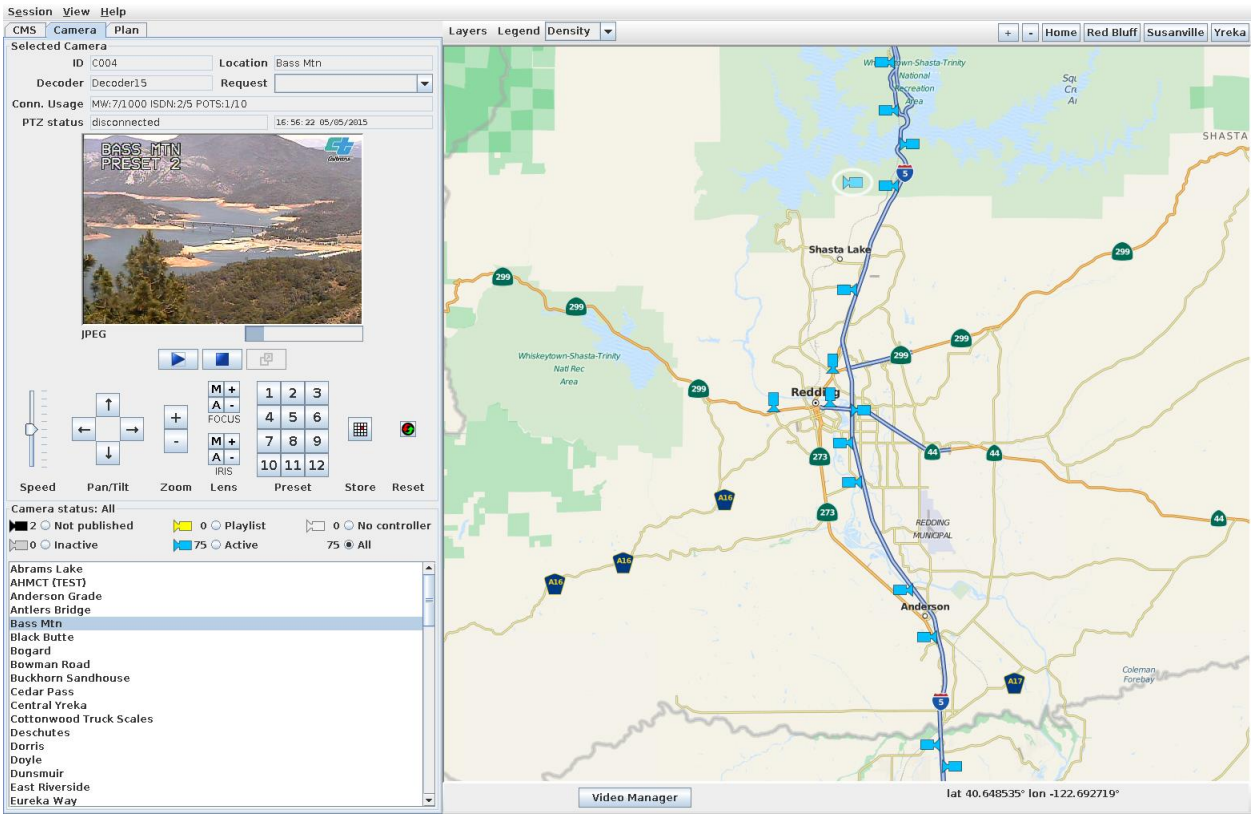

Figure 3.19: Video interface

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 ten preset camera positions.

IRIS System Maintenance

As part of the rollout of IRIS 9.3.0, we transitioned from deploying IRIS onto bare-metal Fedora systems to deploying IRIS onto virtual VMware machines running SUSE Linux Enterprise Server (SLES). This change required a number of modifications to the IRIS build and deployment process, but has ultimately allowed Caltrans to benefit from the intrinsic flexibility and ease of recovery that is inherent in the use of virtual machines.

District IRIS administrators have received Caltrans training on their VMware systems, and should be prepared to perform basic administrative tasks on the IRIS virtual machines. In addition to maintenance at the VMware level, some routine maintenance on the IRIS systems themselves is required in order to keep them running effectively and efficiently. Generally speaking, maintenance includes:

- Performing regular VM snapshots/backups

- Updating operating system packages
- Ensuring that sufficient free disk space exists
- Updating IRIS configuration if field elements change or are added (e.g., VDS, CMS, CCTV, RWIS sites, AWS configuration)

Mapping

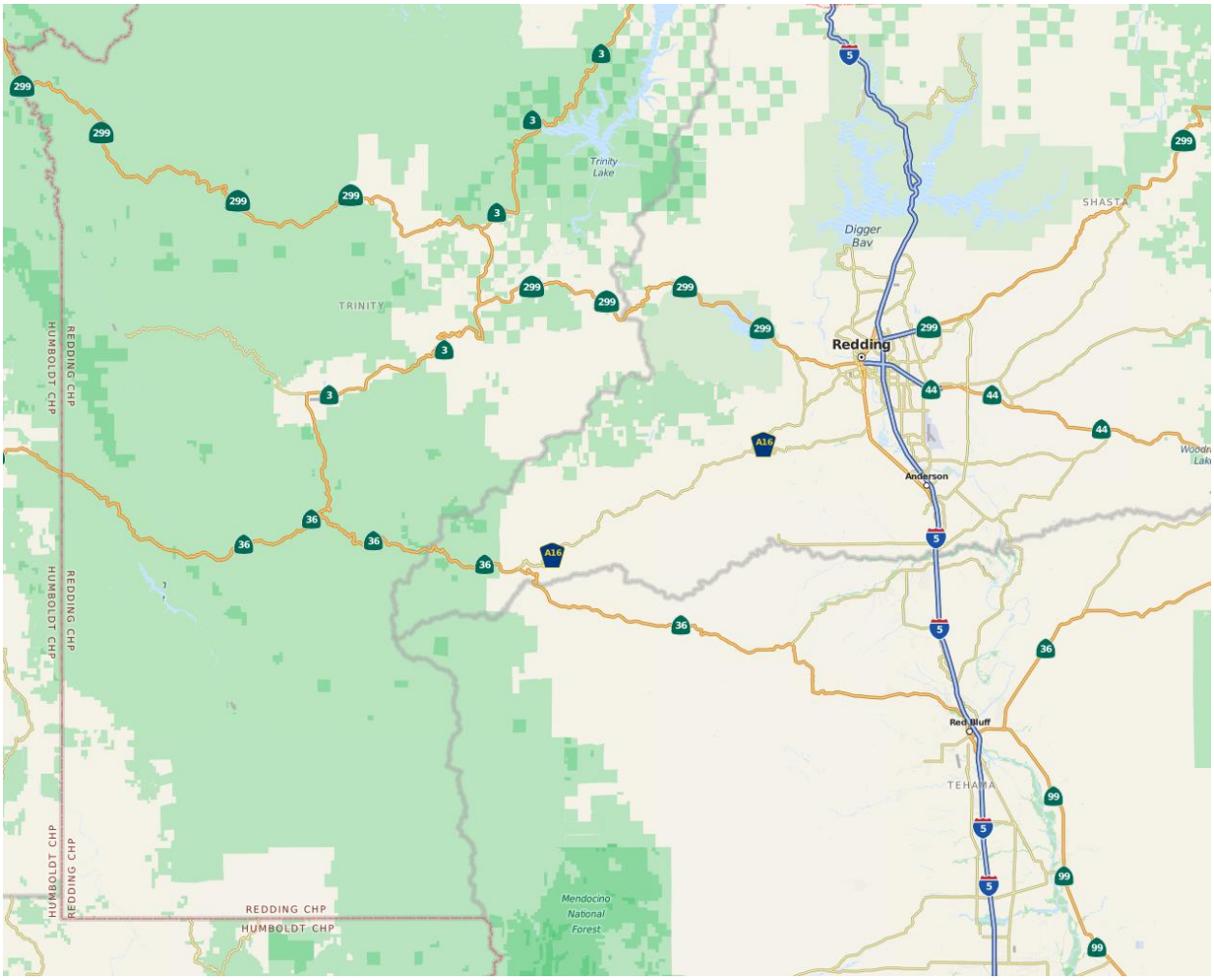

Figure 3.20: New Caltrans IRIS map

Figure 3.20 shows the new Caltrans IRIS. It is based on data from the OpenStreetMap (OSM) project, and was first introduced in release 9.3.0. The California map tiles were originally generated by AHMCT, and were significantly updated by KSD, as discussed in Chapter 4.

Completed Enhancements

Completed IRIS enhancements include:

- IRIS Release 9.3
    - Support for D1/D2/D5 CCTV infrastructure
    - Support for Axis video decoders
    - Support for Cohu and Axis PTZ protocols
    - 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
    - Various new reports, including monthly CMS activation report
    - Support for SUSE Enterprise Linux
- IRIS Release 10.0
    - Merge of California IRIS code with MnDOT version 4.22
    - Updated base map
    - Assorted bug fixes
    - Acceptance testing planned for May 2015
    - Release planned in June 2015

Future Enhancements

Future IRIS enhancements include:

- IRIS Release 10.1 and beyond
    - CCTV home presets for daylight and nighttime schedules
    - Bulk configuration for field elements
    - Ability to configure AWS with IRIS client
    - Integration with weigh-in-motion
    - Support for Bluetooth-based travel times
    - Spell check CMS messages, including abbreviations

- CMS XML feed
- PeMS enhancements
- Monitoring and control of portable CMS
- Add support for district chain control operations
- Ramp metering
- Integration with TMCAL (Transportation Management Center Activity Logging)
- URMS ramp-metering support
- HAR support
- Lane-closure support
- Inter-district travel time generation

IRIS Release 10.1 and future releases are anticipated biannually under maintenance contract with SwRI. AHMCT has participated on the requirements definition for Release 10.1. SwRI is now established as the Caltrans' IRIS maintenance contractor for the next two years. Competitively bid IRIS software maintenance contracts are expected to continue into the future. In addition to ongoing maintenance, it is anticipated that Caltrans will pursue more substantial enhancements to IRIS, including potential capabilities such as Highway Advisory Radio, ramp metering, and extensive graphical user interface features. These more extensive enhancements are beyond IRIS maintenance, and will require additional resources. As such, it is anticipated that Caltrans will go to bid for development of these enhancements as they are identified and prioritized by the Caltrans IRIS team, and as funding becomes available.

CHAPTER 4:
KNOWLEDGE TRANSFER FOR SYSTEM MAINTENANCE

With the successful deployment of IRIS to four rural districts in Caltrans, this project is the closing phase for active research on the IRIS platform. Caltrans has established a maintenance contract with SwRI, and AHMCT supported knowledge transfer to SwRI during the final months of this project. This process will continue beyond the end of the current project. AHMCT has also supported knowledge transfer to its subcontractor, KSD, during their work on multiple IRIS tickets. Those efforts will be detailed here.

KSD Enhancement of IRIS Static Map

The IRIS OSM-based static map was added in IRIS 9.3, using tiles generated by AHMCT. This was a substantial improvement over prior IRIS mapping, which was simply line images representing roadways. However, several potential improvements were identified by the operators and managers, including:

- Updating the map to reflect the current OSM database
- Enhancing the visibility of county boundaries
- Reducing the density of highway signs
- Adding highway exit numbers
- And adding California Highway Patrol (CHP) dispatch boundaries and labels.

KSD performed this work from October through December 2014. An essential component of this effort involved updating the mapping tool chain, which was last used in 2012. Many of the tools used in the mapping process changed significantly in this period. Executables had to be updated, configuration files had to be revised, and mapping scripts required modification. AHMCT updated the mapping tool chain, and provided a fast mapping server for KSD's use. The mapping tool chain is documented in the California IRIS developer documentation for future reference. As mapping updates are generally a significant effort, mapping updates are relatively infrequent, and it is highly likely that the tool chain will need significant updating each time the map is updated.

SwRI Test Case Development

SwRI began its role as the IRIS maintenance contractor in July 2014. Their staff attended regular and as-needed IRIS meetings throughout the remainder of the project, and beyond. To facilitate the knowledge transfer process, Caltrans organized an in-person meeting, held at the AHMCT Research Center, on July 8 - 9, 2014.

In order to rapidly get SwRI up to speed, SwRI was tasked with acceptance test case development for IRIS 10.0. SwRI began working on test cases in August 2014, and finished the

final test cases at the end of this research project. Final IRIS 10.0 acceptance testing and release will follow the conclusion of this project.

Improvement and Simplification of the IRIS Build Process

The IRIS build process had been developed by and for AHMCT researchers. It was appropriate for such use. However, when the time came for transfer to another development team, it was clear that the process could be improved, simplified, and streamlined. AHMCT has made these process improvements in order to facilitate transitioning of development and support to SwRI. The current IRIS build process is detailed in the California IRIS developer documentation for future reference.

CHAPTER 5:
CONCLUSIONS

Key contributions of this research project included:

- Deployment of IRIS to Districts 1, 2, 5, and 10
- Enhancements and maintenance of IRIS for four districts
- A substantially simplified, improved, and streamlined IRIS build process
- Significantly improved IRIS mapping.

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 which may lead to better manage IRIS development and support into the future:

- More detailed communications with the Caltrans districts is important, especially during the design phase of new enhancements. Meeting the requirements alone does not guarantee acceptance by the district.
- 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 (forking) is not recommended. In order to prevent this, it is essential that feedback from MnDOT is solicited as early as possible in the requirements or design phase, since designing enhancements to be as general as possible increases the likelihood that the changes will be accepted upstream. While this approach requires more time upfront, it will save time in the long run, since maintaining a code fork is extremely time-consuming. With unmerged changesets, each of these changesets must be refactored during every 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 MnDOT. In the future, a parallel model based on feature branches is suggested.

Future Work

With IRIS deployed into Districts 1, 2, 5, and 10, on-going software maintenance will continue through third-party software integrators using the State's competitive bidding process. It is important to plan for continued maintenance, but also to have a mechanism to gear up for major enhancements and overhauls, i.e. enhancements that are above and beyond the capacity of planned maintenance team efforts.

Under previous phase efforts, some unexpected regressions were experienced after developing new features or fixes. The creation of automated test cases has helped to catch regressions in advance of acceptance testing and release. End-to-end automated test cases throughout IRIS are needed. Significant work has been spent in developing such tests. However, further work is needed, and this should be an ongoing area of effort and attention.

As noted in Chapter 3, developing a consistent and effective approach to documentation for an open-source project with multiple contributors is challenging. Approaches that can be effective for monolithic efforts may not be successful for open-source development. There are effective models available in the open-source community. Evaluating effective models and state-of-practice for open-source project documentation would be an important future task.

MnDOT continues its own IRIS development at a rapid pace. Keeping up with MnDOT's code changes is a difficult but vital task. Handling change management is a particularly challenging area that can delay system delivery and, if not handled properly, can lead to forking of a software project, with resultant loss of collaborative benefits. Unintentional *de facto* forking can result from a lack of attention toward change management. For continuing IRIS efforts, for both maintenance and enhancements, it will be essential for the two DOTs and any of their contractors to collaborate more closely and effectively. It would be extremely useful to look at existing and previous related projects in open-source and transportation software development, identify best practices and procedures, and develop recommendations for future Caltrans collaborative efforts, including ongoing efforts such as IRIS. Improved change management procedures will enhance the results and value of Caltrans collaborative software development, and allow Caltrans to more easily and fruitfully collaborate with other agencies.

Given the discussion in the previous paragraph, it is clear that proper code management to avoid forking takes developer effort. Given the maturity of California's IRIS implementation, it is reasonable to ask whether it might make sense to fork at this time, and focus this effort into exclusive California developments. There are advantages and disadvantages to this. Most obviously, no further effort would be needed to manage the code to keep current with another agency's code. The developer time that would have been spent on this effort could be channeled into development focused exclusively on Caltrans' needs. However, a decision to permanently fork has serious implications. Caltrans would then lose the benefit of any future developments by MnDOT. In addition, Caltrans would not see any benefits from contributions to IRIS by other states, e.g. Nebraska and Wyoming, who would most likely continue participating in MnDOT's

IRIS project. In fact, it would likely be difficult to attract any other states to participate with California, given the tendency of Caltrans to develop its own field element standards. In addition, it would be less attractive to consultants to bid on California-specific IRIS development. A key current motivation for consultants to bid and develop expertise in IRIS is that they can anticipate leveraging this expertise through work with other states. In general, Caltrans would lose the known and demonstrated benefits of participating in an open-source project, were they to deliberately fork [2]. It would be instructive to quantify this issue by tracking the amount of time or money spent on code management to prevent forking vs. the benefits of developments provided by MnDOT and other participants. It must be emphasized that the level of effort experienced in the current project to get back in synch with MnDOT is far higher than what would be experienced using proper code management practices once Caltrans is back in synch with MnDOT. Our estimate is that by keeping current with MnDOT on at least a monthly basis, the effort required would be approximately one to two hours per month. Our strong belief is that Caltrans will be better served by continuing as a part of the MnDOT IRIS open-source project.

New features and enhancements are regularly requested along with ongoing maintenance and minor bug fixes. SwRI is now established as the Caltrans IRIS maintenance contractor for the next two years. Competitively bid IRIS software maintenance contracts are expected to continue into the future. In addition to ongoing maintenance, it is anticipated that Caltrans will pursue more substantial enhancements to IRIS, including potential capabilities such as Highway Advisory Radio, ramp metering, and extensive graphical user interface features. These more extensive enhancements are beyond IRIS maintenance, and will require additional resources. As such, it is anticipated that Caltrans will go to bid for development of these enhancements as they are identified and prioritized by the Caltrans IRIS team, and as funding becomes available.

REFERENCES

1. M.T. Darter, T.B. Swanston, K.S. Yen, B. Ravani, and T.A. Lasky, "Ahmct Intelligent Roadway Information System (Iris) Technical Support and Testing," Advanced Highway Maintenance and Construction Technology Research Center Rept. # UCD-ARR-11-12-31-01, 2011.
2. 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

(For releases in report period)

CA-IRIS 9.4.4, October 2014

- SV170 driver update for v4.1 compatibility (#545)
- SignScope XML feed (#535)

9.4.3, April 2014

- Axis VAPIX2 PTZ driver (#529)
- video decoder support for D1 (#530)
- Modifications to support trusted certificate (#531)
- fixed CCTV op status timestamp format (#539)
- change VDS fail-time semantics (first fail in series, not most recent)
- configurable VDS timing margins
- only update status on client for VDS failures that are new (to reduce SONAR traffic); increase max VDS timing margins
- use VAPIX2 serial API instead of OSD menu to implement camera reset
- package with trusted cert; unify district-specific build configurations into single build workflow

9.4.2, January 2014

- traffic controller failure status (#480)
- traffic controller failure timestamp (#502)
- fix Windows 7 MPEG4 WMP launch issue (#524)

9.4.1, November 2013

- PTZ communications status bar (#497)
- workaround for D2 corrupted SignView header (#516)
- client JAR manifest changes in preparation for JRE 7u51 (#527)

9.4.0, November 2013

- CCTV "return-home" feature (#417)
- CMS report export (#472)
- lens and reset control support for Pelco-D driver (#483)
- fix for D5 CCTV viewer launch issue (#515)
- PTZ Commands sent when site not connected (configurable) (#518)
- IRIS times out with error on exit (#520)
- "Current CMS Status" report: printable version (#521)
- PTZ timing issues (#522)

9.3.3, July 2013

- fix bug preventing addition of new detectors to a controller (#501)

9.3.2, June 2013

- modify TCP socket handling to avoid connection conflicts with D2 infrastructure (#513)

9.3.1, June 2013

- fix client NPE crash (#495)

9.3.0, May 2013

- CCTV UI enhancements
- CMS Monthly Usage Report
- RWIS wind gust speed/direction support
- CCTV iris, focus, and reset control
- CMS composer reliability enhancements
- external video launcher support for MPEG4 and WMV
- CMS "power cycle detection" feature
- map tooltip enhancements
- now uses OSM-based map system
- driver for Sensys AP240
- driver for RTMS G4
- Axis 292 video decoder control support
- new CMS status feed
- many bug fixes
- port IRIS distribution to SUSE/SLES 11

9.2.3, December 2011

- Purpose: Fix client login problem (#435) and CCTV read timeout error (#440).