California AHMCT Program
University of California at Davis
California Department of Transportation

DEVELOPMENT AND FIELD-OPERATIONAL TESTING OF A MOBILE REAL-TIME INFORMATION SYSTEM FOR SNOW FIGHTER SUPERVISORS*

Kin S. Yen[1], Michael T. Darter[1], Ha Nguyen[2], Bahram Ravani[1], &
Ty A. Lasky[1], Principal Investigator

AHMCT Research Report
UCD-ARR-06-12-31-07

Final Report of Contract IA65A0139 - Task Order 06-12

December 31st, 2006

Affiliations:
1. AHMCT Research Center, Department of Mechanical & Aeronautical Engineering, University of California, Davis, CA 95616-5294
2. Caltrans Division of Research & Innovation, 3337 Michelson Drive Suite CN380, Irvine, CA 92612-8894

* This report has been prepared in cooperation with the State of California, Business and Transportation Agency, Department of Transportation and is based on work supported by Contract Number IA65A0139 - Task Order 06-12 through the Advanced Highway Maintenance and Construction Technology Research Center at the University of California at Davis.

Technical Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
S/CA/RI-2006/31		
4. Title and Subtitle Development and Field-Operational Testing of a Mobile Real-Time Information System for Snow Fighter Supervisors		5. Report Date December 31st, 2006 6. Performing Organization Code
7. Author(s): Kin S. Yen, Michael T. Darter, Ha Nguyen, Bahram Ravani, & Ty A. Lasky		8. Performing Organization Report No. UCD-ARR-06-12-31-07
9. Performing Organization Name and Address AHMCT Research Center UCD Dept of Mechanical & Aeronautical Engineering Davis, California 95616-5294		10. Work Unit No. (TRAIS) 11. Contract or Grant IA65A0139 - Task Order 06-12
12. Sponsoring Agency Name and Address California Department of Transportation Division of Research and Innovation 1127 O Street Sacramento, CA 94273-0001		13. Type of Report and Period Covered Final Report July 2005 - December 2006 14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

This report documents the Mobile Real-Time Information System for Snowplow Supervisors (MRTIS) System, a research project undertaken by the Advanced Highway Maintenance & Construction Technology (AHMCT) Research Laboratory and funded by California State Department of Transportation (Caltrans) Division of Research and Innovation (DRI) at the time of the projects completion (December 2006) and before the final project demonstration in June of 2007. The MRTIS project researched, designed, built, and field tested a mobile real-time information system for snow fighter supervisors. The system goal is to provide a prototype hardware and software system that provides crucial information to snowplow supervisors while they are in the field. The availability of crucial information in the field may increase their ability to be physically present in the field while they are assisting and communicating with other snow fighters, which may improve efficiency, safety, enhance improve resource allocation decisions, and minimize environmental impact. MRTIS uses commodity hardware, Commercial-off-the-shelf (COTS) and open-source software to provide a mobile information system. Google Earth© is used as a system visualization client integrating sensor data, vehicle paths, and weather information. Wi-Fi wireless and cellular modems are used for communications between vehicles and the winter maintenance operations center. Results show the system is effective at visually integrating different types of information, collecting real-time vehicle locations and road temperatures, and presenting historical information.

17. Key Words		18. Distribution Statement	
Mobile Real-time Information System, MDSS, RWIS, ATMS, Highway maintenance, Highway winter maintenance.		No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.	
20. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	90	

Form DOT F 1700.7 (872) Reproduction of completed page authorized

(PF V2.1, 6/30/92)

Abstract

This report documents the Mobile Real-Time Information System for Snowplow Supervisors (MRTIS) System, a research project undertaken by the Advanced Highway Maintenance & Construction Technology (AHMCT) Research Laboratory and funded by California State Department of Transportation (Caltrans) Division of Research and Innovation (DRI) at the time of the projects completion (December 2006) and before the final project demonstration in June of 2007. The MRTIS project researched, designed, built, and field tested a mobile real-time information system for snow fighter supervisors. The system goal is to provide a prototype hardware and software system that provides crucial information to snowplow supervisors while they are in the field. The availability of crucial information in the field may increase their ability to be physically present in the field while they are assisting and communicating with other snow fighters, which may improve efficiency, safety, enhance improve resource allocation decisions, and minimize environmental impact. MRTIS uses commodity hardware, Commercial-off-the-shelf (COTS) and open-source software to provide a mobile information system. Google Earth© is used as a system visualization client integrating sensor data, vehicle paths, and weather information. Wi-Fi wireless and cellular modems are used for communications between vehicles and the winter maintenance operations center. Results show the system is effective at visually integrating different types of information, collecting real-time vehicle locations and road temperatures, and presenting historical information.

Table of Contents

List of Figures

List of Tables

Disclaimer/Disclosure

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 Aeronautical 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.

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, the Federal Highway Administration, or the University of California. This report does not constitute a standard, specification, or regulation.

Acronyms and Abbreviations

Acronyms used within this document are defined below. Terms and definitions specific to the MRTIS project are defined in the Definitions, Acronyms, and Abbreviations (Section 2.1.3 on page 12), within the Requirements and Goals chapter.

2V MRTIS Caltrans Secondary Winter Maintenance Vehicle

802.11 Wi-Fi IEEE wireless LAN/WAN standards

ADIS Advanced Driver Information Systems

AHMCT Advanced Highway Maintenance & Construction Technology

ATIS Advanced Traveler Information Systems

ATMS Advanced Traffic Management System

ATWIS Advanced Transportation Weather Information System

AVCS Advanced Vehicle Control System

AVCSS Advanced Vehicle Control & Safety Systems

Caltrans California State Department of Transportation

CCTV Closed-circuit Television

CHP California Highway Patrol

CMS Changeable Message Sign

CO MRTIS Caltrans Maintenance Central Office

COTS Commercial-off-the-shelf

CVO Commercial Vehicle Operations

DOT Department of Transportation

DRI Division of Research and Innovation

DSS	Decision Support System
EDGE	Enhanced Data rates for GSM Evolution
EIA	Electronic Industries Alliance
ESS	Environmental Sensor Stations
EY	MRTIS Caltrans Maintenance Equipment Yard
FHWA	Federal Highway Administration
GIS	Geographic Information System
GMT	Greenwich Mean Time
GNOME	GNU Network Object Model Environment
GNU	GNU's Not Unix
GPL	GNU General Public License
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HMI	Human-Machine Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISP	Information/Internet Service Provider
IT	Information Technology
ITS-4	4th generation ITS technologies
ITS	Intelligent Transportation Systems
J2EE	Java Platform, Enterprise Edition
JAI	Java Advanced Imaging
JMS	Java Messaging Service

JPG	Joint Photographic Experts Group
KML	Keyhole Markup Language
KMZ	Keyhole Markup Language Zipped
LAN	Local Area Network
LCD	Liquid Crystal Display
LOS	Level-of-Service
MDSS	Maintenance Decision Support System
MRTIS	Mobile Real-Time Information System for Snowplow Supervisors
NASA	National Aeronautics and Space Administration
NHS	National Highway System
NMEA-0183	National Marine Electronics Association GPS communications standard
NTCIP	National Transportation Communications for ITS Protocol
NWS	National Weather Service
OWL-DL	Web Ontology Language Description Logics
OWL	Web Ontology Language
PHP	PHP Hypertext Preprocessor
RDF	Resource Description Framework
RS-232	EIA serial communications recommended standard
RWIS	Remote Weather Information System / Roadway Weather Information System
SHRP	Strategic Highway Research Program
SOAP	Simple Object Access Protocol
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
SRS	Software Requirements Specification
STWDSR	Surface Transportation Weather Decision Support Requirements
SV	MRTIS Caltrans Winter Maintenance Supervisor Vehicle

TAG	Technical Advisory Group
TCP/IP	Transmission Control Protocol / Internet Protocol
TMC	Transportation Management Center
TPF	Transportation Pooled Fund
UCD	University of California Davis
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USB	Universal Serial Bus
USDOT	U.S. Department of Transportation
UTC	Coordinated Universal Time
VAMS	Value-Added Meteorological Services
VDS	Vehicle Detector Station
VII	Vehicle Infrastructure Integration
W3C	World Wide Web Consortium
WAN	Wide Area Network
Wi-Fi	Wireless IEEE LAN/WAN standards
WiMAX	Worldwide Interoperability for Microwave Access
WIST-DSS	Weather Information for Surface Transportation Decision Support System
WMO	World Meteorological Organization
WTI	Western Transportation Institute
x86	Commodity Intel, AMD, and compatible microprocessors
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language and Transformation

Acknowledgments

The authors thank Mr. Kin Yen for his invaluable assistance in the design and implementation of MRTIS hardware. The authors also thank the Caltrans Technical Advisory Group (TAG) for its contributions, guidance, and feedback. The TAG includes Stan Richins (Sutter Sierra Region Manager), Sean Campbell (DRI, communications), Randy Woolley, and Ha Nguyen. The authors also thank the Caltrans Equipment Service Center for their assistance in this project. Finally, we thank Caltrans DRI for their support of this project.

Chapter 1

Introduction

1.1 Report Purpose

This report summarizes and details the Mobile Real-Time Information System for Snow Fighter Supervisors project. MRTIS provides snow fighter supervisors in the field with the abundance of information previously available only in the office. Around-the-clock access to this crucial information and the supervisor's ability to stay in the field longer can improve resource allocation decisions, enhance efficiency, increase safety, minimize environmental impact, and enhance the Caltrans work environment. This improves the responsiveness of snow removal equipment and crews, thus lowering winter operational costs, and helps keep the road open.

1.2 Definition of Engineering Problem

The MRTIS project researched, designed, built, and field-tested a mobile real-time information system for snow fighter supervisors. The system goal is to provide a solution within a framework that provides crucial information to snowplow supervisors while they are in the field, so that supervisors can increase their ability to be physically present in the field while they are assisting and communicating with snow fighters.

1.3 Project Motivation

With increasingly constrained State budgets, Caltrans winter maintenance and snow removal operations must do more with less. Snow fighter supervisors are crucial in this effort. Ideally, supervisors need to be in two places at the same time: the office and the field. In the office, supervisors have access to rapidly changing weather forecasts and weather radar, traffic conditions, satellite images, phones, and, in some cases, Remote

Weather Information System / Roadway Weather Information System (RWIS) terminals. In the field, supervisors are often needed during a snowstorm to assist other snow fighter team members. The supervisor's physical presence in the field enables direct observation of fast-changing road and weather conditions which facilitates decision making. The primary goal of this research project is to provide a system which gives snow fighter supervisors in the field information previously available only in the office. Around-the-clock access to this crucial information and the supervisor's ability to stay in the field longer can improve resource allocation decisions, enhance efficiency, increase safety, minimize environmental impact, and enhance the Caltrans work environment. This can improve snow removal responsiveness, thus lowering winter operational costs and road closure times. A mobile information system for snow fighter supervisors is a component within the larger conceptual models of the Maintenance Decision Support System (MDSS) and Intelligent Transportation Systems (ITS) framework.

Efficiency in winter maintenance is important for safety and monetary reasons [13]. In terms of safety, on an annual basis, winter weather crashes[1] result in 389,000 crashes, 133,000 injured persons, and 1,500 fatalities Nationwide. Nearly one quarter of all crashes occur on slick pavement and in unfavorable weather [20]. In financial terms, state DOTs spend about 25% of their maintenance budgets on winter road maintenance. Including local agencies, this amounts to approximately $2.5 billion on an annual basis.

1.4 Background of Previous Work

The MRTIS project fits within a larger ITS context (see Figure 1.1 on the next page). This includes MDSS and RWIS. The background information summarized below strongly shaped MRTIS requirements (pg. 11) and design (pg. 25).

1.4.1 The Intelligent Transportation System Era

The National Highway System (NHS) build-out era commenced in 1956 with the National Interstate and Defense Highways Act. It initiated a 35-year $114 billion program that had a profound effect on the United States. It was mostly complete by 1991, and the era of build-out was over. In the mid to late 1980s transportation officials from Federal and State governments, the private sector, and universities began a series of informal meetings discussing the future of transportation. This included meetings held by Caltrans in October of 1986 to discuss technology applied to future advanced highways [29]. In June of 1988 in Washington, DC, the group formalized its structure and chose the name Mobility 2000, which was latter changed to ITS America [28]. Over time, ITS conceptually progressed from an automated highway concept to a comprehensive transportation Information Technology (IT) perspective. The first ITS America meeting was held in 1989 and was organized into four areas:

[1]"winter weather crashes" are those that occur on pavement that has ice, snow, sleet, or any combination

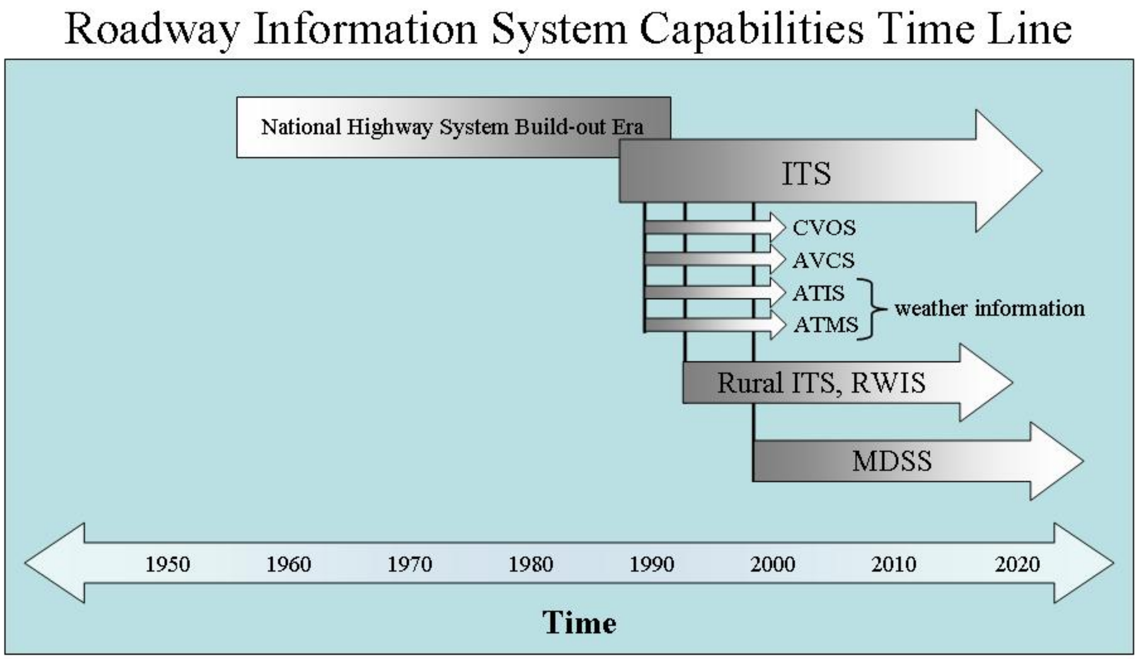

Figure 1.1: Roadway Information System capabilities time line

- Advanced Traffic Management System (ATMS)
- Advanced Driver Information Systems (ADIS), now ATIS
- Commercial Vehicle Operations (CVO)
- Advanced Vehicle Control System (AVCS), now Advanced Vehicle Control & Safety Systems (AVCSS).

1.4.2 The Roadway Weather Information System (RWIS)

The RWIS is an ITS subsystem of particular interest to winter road maintenance; it seeks to provide road condition and weather information from a variety of sources. RWIS focuses on providing multiple types of information to road maintenance mangers. These independent data sources are often proprietary, i.e. the providers may own the data and/or use proprietary communication protocols. Therefore, there is less emphasis on data fusion and analysis of multiple data sources (see Figure 1.2 on the following page). RWIS capabilities in general include [14]:

- Environmental Sensor Stations (ESS) for observing road conditions
- Additional processing for road condition prediction

- Atmospheric condition reports using Value-Added Meteorological Services (VAMS)
- Other road condition and weather products provided through Information/Internet Service Providers (ISPs).

Historically, weather information and road conditions have been important aspects of ATMS and Advanced Traveler Information Systems (ATIS) systems. A 1993 Strategic Highway Research Program (SHRP) report "Road weather information systems" discusses the RWIS approach, including VAMS, sensors such as ESS and thermal road sensors, and the effectiveness of RWIS programs [22]. RWIS development in Europe started as early as 1972, incorporating simple weather stations communicating via the telephone network [2].

The RWIS approach and systems have a number of strengths and weaknesses [14]. Their decentralized client-server architecture enables qualitative and quantitative improvements. Primary weaknesses include insufficient standards across all information sources, and insufficient research related to winter maintenance decision-making processes. Both of these problems are addressable by further research and standards.

There are many implemented RWIS systems. For example, the Advanced Transportation Weather Information System (ATWIS) system in North Dakota, South Dakota, and Minnesota; the FORETELL™projects in Iowa, Missouri, and Wisconsin; and the RWIS system in Sweden, among many others [12, 2, 14].

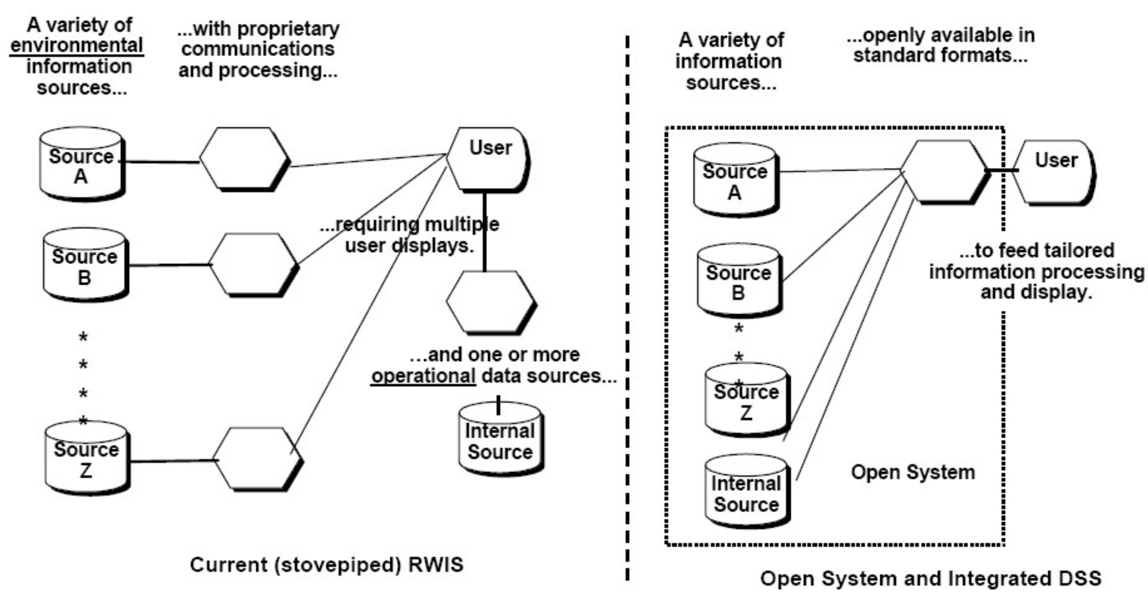

Figure 1.2: RWIS data flow vs. MDSS data flow [14]

1.4.3 The Maintenance Decision Support System (MDSS)

Conceptually, MDSS is an evolutionary refinement of the RWIS approach. In essence MDSS seeks to close the gap between available information about weather and roads and the decisions that snow fighters must make. To achieve this, MDSS seeks to provide an open-standards platform that provides data fusion capabilities. Decision support applications combine data from a variety of sources to present information to users [15], as shown in Figure 1.2. This approach builds on the RWIS foundation that provides multiple potentially proprietary data streams. MDSS also provides a conceptual framework for providing appropriate information to different actors. For example, snow fighter supervisors need different information and decision support than maintenance personnel. MDSS originated with the Federal Highway Administration (FHWA) Rural ITS Program in 1998 as the Surface Transportation Weather Decision Support Requirements (STWDSR) project. It was originally identified as the Weather Information for Surface Transportation Decision Support System (WIST-DSS). The MDSS program seeks to develop a system which [25]:

- Uses existing road and weather data sources
- Uses and improves existing data sources for decision-making tasks
- Fuses data to create an open, integrated and understandable presentation of current environmental and road conditions
- Produces diagnostic and prognostic maps of road conditions, with emphasis on the 1 to 48 hour horizon, with 48-hour historic data availability
- Provides a display capability on the state of the roadway
- Provides a Decision Support System (DSS) tool which provides maintenance recommendations
- Provides all of the above on a single platform, which provides a display of results and recommended courses of action, together with anticipated consequences of action or inaction.

The Transportation Pooled Fund (TPF) Program TPF-5(054) is an effort to further refine, develop, and deploy the MDSS system [24]. Project partners include ten states and the FHWA. Project objectives are:

- Assess needs, potential benefits, and receptivity in participating DOTs for MDSS.
- Define requirements for an operational MDSS that can assess current road and weather conditions, forecast weather effecting transportation routes, predict how road conditions will change in response to treatments, suggest optimal maintenance strategies, and evaluate the effectiveness of applied maintenance treatments.

- Build and evaluate an MDSS.
- Improve road condition forecasts in response to changing weather and applied maintenance treatments.

A benefit-cost study was performed by Western Transportation Institute (WTI) to assess MDSS [30]. The project, 'Analysis of Maintenance Decision Support System Benefits and Costs' sought to assess benefits and costs of MDSS implementation and was funded by the MDSS Pooled Fund Study (see Figure 1.3).

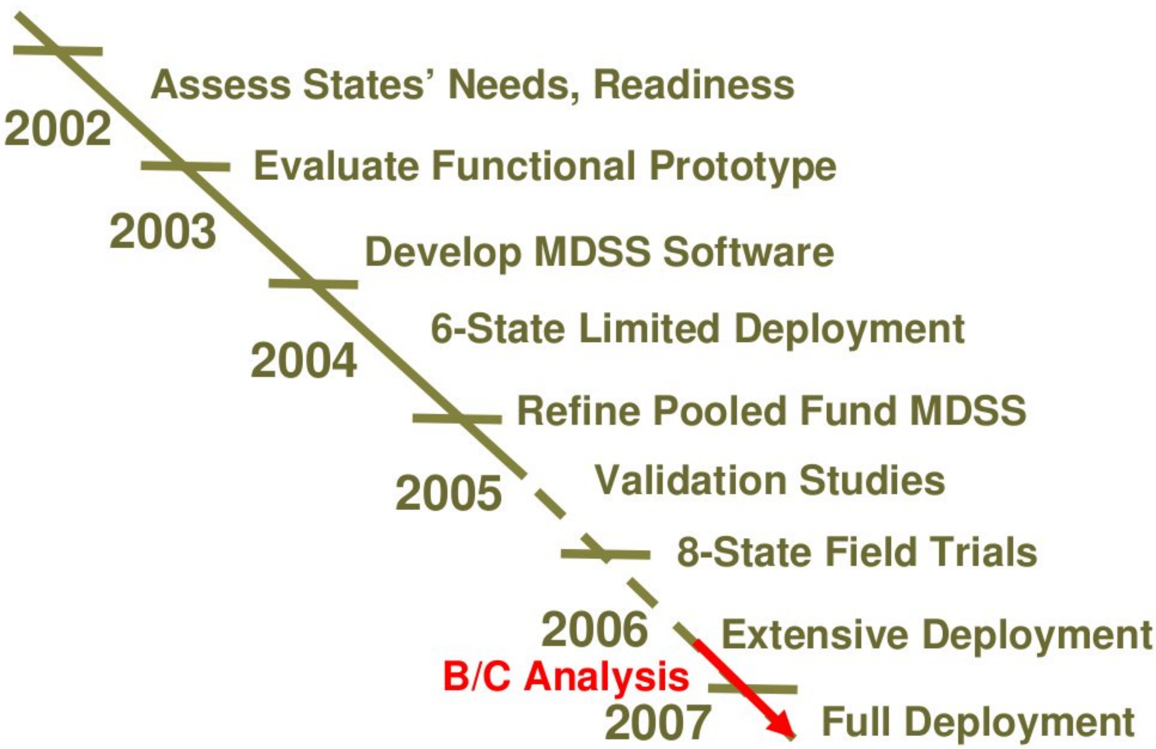

Figure 1.3: MDSS Pooled Fund development time line [30]

Anti-icing is an example of a newer technology applied in the MDSS environment that may enable improvements in efficiency, but require a more anticipatory maintenance strategy, which increases the need for accurate and timely information. In the last decade many Department of Transportations (DOTs) have experimented with anti-icing, the anticipatory spreading of chemicals. Experiments have shown application rates on the order of 4 g/m^2, as much as ten times lower than required in de-icing operations [5]. An Idaho research program reported dramatic reductions of 60 to 80 % in material use, labor and traffic accidents [8]. Anti-icing requires very accurate timing. It is found to be most effective if applied no less than one hour before the pavement starts to freeze. Timely, accessible and accurate information enables a high-quality anti-icing program. Having crucial information to make timely decisions reduces operational costs, and keeps roads safe and open, and minimizes environmental impact. The lack of integration of data sources and data complexity may impose serious limitations for efforts to improve winter maintenance efficiency.

1.4.4 The Clarus Initiative

The Clarus Initiative is a project sponsored by the U.S. Department of Transportation (USDOT) to develop a Nationwide surface transportation weather observing and forecasting system [26]. Clarus is a tier-one ITS initiative. The overall goal is to provide information to transportation managers and other users to reduce the effects of adverse weather, such as increasing safety and reducing weather-related delays. The system will build on MDSS advances and also seeks to provide an open and extensible platform that can be implemented by other organizations. The focus is on providing a National road weather data repository. Others will use the repository to generate specific weather-related information such as VAMS. The Clarus Initiative began in late 2004 and ends in 2009. System design commenced in 2004 and ended in 2006. Clarus will also attempt to better define differences between evolving World Meteorological Organization (WMO) standards and ITS weather standards.

The Clarus Initiative is closely related to the Vehicle Infrastructure Integration (VII) Initiative, also sponsored by the USDOT. VII seeks to reduce congestion and prevent accidents using vehicle-to-roadside and vehicle-to-vehicle communication [27].

1.5 Trends within ITS and MDSS

ITS and MDSS trends are being driven by a number of sources. Semiconductor process technology continues to lower power requirements, reduce chip size, and double performance about every 18 months. This is simultaneously pushing four trends which are known as 4th generation ITS technologies (ITS-4) [23]:

- Decentralization of processing intelligence
- Commoditization of sensors
- Commoditization of processors
- Commoditization of communication

These are combined to produce results in real time. Commoditization of software is also a key trend [11]. Within the MDSS conceptual framework, a number of emerging trends have been identified for different types of data[14, 10]:

- Road temperature data: display of data on routes, sharing of data with other users.

- Road temperature prediction: Geographic Information System (GIS) Graphical User Interface (GUI) based, a function of modeling, discrete time horizons.
- Environmental Sensor Stations (ESS) road conditions: GIS GUI based.
- Weather observations: GIS GUI based, integration with different types of ESS, mesonets.
- Weather prediction: increased availability of regional (meso-scale) weather products. Integration with GIS GUI.
- Other transportation information: integration with other ITS data sources, e.g. Transportation Management Centers (TMCs). Increased wireless access. Travel time predictions.
- Decision Support: integration with GIS GUI, route-specific access to information.

In addition, the AHMCT Research Center has identified trends which crosscut visualization, software engineering, and ITS [10]:

- Human Interface: there is an increased need to provide a software solution with a single GIS GUI that integrates data from many sources. For example, weather products, static and dynamic asset information, sensor data such as GPS and traffic-cams, and historic information.
- Visualization of data: increased need to include the time-scale as a GIS visualization and query dimension. This can be used across domains and is therefore an integration opportunity and challenge.
- Software engineering: increased need to work with complex distributed data (semantic) defined in non-proprietary formats (see the appendix on pg. 65).

1.6 Research Tasks and Progression

Project development progressed approximately as indicated below. Software development used an evolutionary linear process model, which followed the sequence: requirements definition, design, develop, test.

1.6.1 Task 1: Literature Review

This task consisted of a review of existing literature, product, and standards. Derived material was incorporated into the project final report and impacted requirements and design. The review included pertinent domestic and foreign documents.

1.6.2 Task 2: Study of Existing Caltrans Operations

In order to understand the snow fighter supervisors information needs, data was gathered about current Caltrans winter maintenance practices, operational methods, procedures, and equipment. In addition, Caltrans management, field supervisors, and operators were consulted to establish their information needs, which helped define system functional requirements. The goal was to gain a more detailed understanding of current Caltrans winter maintenance operations.

1.6.3 Task 3: Component Selection, System Development

System requirements were used as a basis for selection of hardware and software. The System was constructed using COTS hardware and software components. Unit, integration, and system testing was used to verify that testable requirements were met. Integration field-testing was also used.

Task 3A: System Server Integration

The System servers were designed to be installed in the MRTIS Caltrans Maintenance Central Office (CO) and connected to the Internet with a broadband Internet connection. The Internet connects MRTIS with VAMS, clients in vehicles, and other clients. Clients in vehicles may also connect through a Local Area Network (LAN) and Wi-Fi as determined by the implementation. During the course of the project, the servers were hosted in the AHMCT Research Center.

Task 3B: Client Implementation

The client computer was designed to use COTS wireless hardware to enable communication with the server. It also interfaced with air and road temperature sensors and GPS to enable thermal mapping. Client hardware and software was designed to be installed inside the MRTIS Caltrans Winter Maintenance Supervisor Vehicle (SV) and MRTIS Caltrans Secondary Winter Maintenance Vehicle (2V). Vehicle power supplies SV and 2V system power. Vehicle-borne sensors (GPS, air and road temperature) and display and input peripherals, were all connected to the client computer. Field and laboratory testing was used to validate the client system.

Task 3C: Software and Human-Machine Interface (HMI) Development

The focus of HMI efforts was on standardized User Interface (UI) elements, ease of use, and integration of information types. Inclusion of the time dimension is a key system feature. This enables both query and visualization with respect to time. HTML and Hyper-

Text Transfer Protocol (HTTP) are key system interfaces, and a standard web browser was used to access some features. Refinement of software and hardware continued into the testing phase. The final product was a functional prototype system consisting of stand-alone servers, mobile computers installed within vehicles, and client computers within the MRTIS Caltrans Maintenance Central Office (CO).

1.6.4 Task 4: Laboratory and Extended Field-testing, Upgrading and Evaluation

Unit, integration, and system testing in the laboratory were used. Integration field testing was also used. Testing verified stated requirements. Remote evaluation and monitoring of the system was performed during testing. Collected client data was archived for debugging purposes. HMI evaluation and feedback are key parts of the system demonstration. Testing under a wide range of circumstances was also a goal.

1.6.5 Task 5: System Documentation

Recommendations and research findings are documented here. Comments, feedback, and recommendations from Caltrans personnel and other stakeholders are included. Design requirements, specifications, system and software documentation, source code, and test results are also provided. Programs were provided as binary modules, along with source code.

1.7 Document Scope

This document was developed by the researchers using project documentation, research reports, software and hardware specifications, software development documentation, field and laboratory data, and guidance and information gained through collaboration with the project TAG.

1.8 Summary of Report

Chapter 2 details the project's context, requirements, and goals. The resulting design and architecture are covered in Chapter 3. The system implementation structure, languages, libraries, and screen shoots are discussed in Chapter 4. Laboratory and field testing is detailed in Chapter 5. Finally, conclusions and future work are discussed in Chapter 6 on page 57. An appendix discusses World Wide Web Consortium (W3C) semantic web standards.

Chapter 2

Requirements and Goals

2.1 Introduction

This chapter discusses MRTIS (a.k.a. the ”System”) requirements and goals. These are used in the subsequent design and implementation phases. In general, this chapter follows the Institute of Electrical and Electronics Engineers (IEEE) Software Requirements Specification (SRS) standard 830-1998 [16]. Requirements were derived from conversations with Caltrans personnel, research reports, literature reviews and ongoing development. The development process used an evolutionary linear process model, which iteratively follows the waterfall sequence of requirements definition, design, development, and testing [9].

2.1.1 Chapter Purpose

This chapter defines MRTIS features and functions which are used in the subsequent design and implementation phases.

2.1.2 Product Scope

MRTIS is a mobile real-time information system for snow fighter supervisors. The goal is to provide information needed by supervisors in the field, which enables them to be physically present in the field to assist and communicate with the snow fighter field crew. Another system requirement is to provide a software and hardware framework for a subset of an MDSS system, tailored for a prototype capability.

2.1.3 Definitions, Acronyms, and Abbreviations

MRTIS-specific terms are defined here. General acronyms are defined on page xv.

- Central office (CO): a fixed office location where snow fighters have access to information sources such as the Internet, phones, satellite weather services, etc. For this study, the Caltrans Kingvale maintenance facility along I-80 in Northern California is assumed.

- Supervisor vehicle (SV): vehicles used by snow fighter supervisors requiring access to MRTIS visualization capabilities and system information from within the vehicle. For development and testing purposes, an AHMCT test vehicle was used.

- Secondary vehicle (2V): vehicles used for winter maintenance operations such as snowplows. These vehicles contain required sensors and data collection hardware, but no visualization software—this class of vehicle is for data collection only. For development and testing purposes, an AHMCT test vehicle was used.

- Equipment Yard (EY): location where winter maintenance vehicles are parked and serviced. For this study, the Caltrans Kingvale equipment yard is assumed. Wi-Fi wireless is assumed to be available in the yard.

- KML Browser: the visualization client used by the System (Google Earth©). See Section 3.2.1 on page 27 for an explanation.

2.1.4 SRS References

The following documents are relevant for this chapter. See the References section on page 63 for further background material.

- IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998.

2.1.5 Overview

The remainder of this chapter describes System functionality, primarily in terms of functional and non-functional requirements. Requirements are grouped by category and numbered for reference elsewhere in the document.

2.2 Overall Description

2.2.1 Product Perspective

This section discusses interfaces between MRTIS and other systems. MRTIS will be used within a larger context of winter maintenance activity. Snow fighters use cellular telephones, 900 MHz radios, satellite weather services, Internet weather services, traffic-cam images, etc. Snow fighters also work with the California Highway Patrol (CHP) and 911 Emergency System. Information sources are discussed below:

- Weather forecasts: this kind of information is available through many sources, e.g. the National Weather Service (NWS). In addition, local forecasters can provide highly-accurate localized information. Snow fighters often rely on TV weather broadcasts and the Internet. Inaccurate weather forecast may force managers to revert to last-minute anti-icing strategies to avoid treatments for storms that do not develop after all [3].

- RWIS: a network of ESS placed at carefully selected locations to measure local weather and road conditions, and relay data to a TMC. Many ESS are also equipped with Closed-circuit Television (CCTV) to enable visual inspection of the road. In the last fifteen years, many DOTs have deployed these systems in locations where the data can be used to achieve a higher Level-of-Service (LOS). RWIS was first fully described in the two-volume Strategic Highway Research Program (SHRP) report "Road Weather Information Systems" [7, 6]. RWIS has quickly become a critical component in maintenance decisions and has helped reduce operating costs [6]. Unfortunately there is little standardization in RWIS equipment, and many ESS do not make local predictions [17]. There is an ongoing effort to standardize protocols in RWIS and other transportation systems, known as the National Transportation Communications for ITS Protocol (NTCIP). Although standards currently released are not mandated, they are a key factor for integrating RWIS into MDSS [18, 19] to achieve better Statewide RWIS integration [3]. In Europe, integration of RWIS with other data sources has already been shown to cut costs through accident reductions, lowered insurance premiums, improved snow removal efforts, and reduced traffic congestion [18]. A study on the use of RWIS conducted in 2002 showed that Caltrans personnel perceived RWIS as useful, especially for snow/ice control and traveler information [3].

- Thermal mapping: a method of determining local road temperatures to aid short-term (up to four hours) forecasting. Thermal mapping of the road can be achieved by driving the routes in a vehicle equipped with a GPS receiver and an infrared temperature sensor. A less accurate but faster method is to interpolate between sensor stations, if the stations have even spacing, good location, and sufficient density. Interpolation will be more challenging in locations with larger variations in altitude, sun radiation, precipitation and wind—all major influences on the road temperature.

- Now-casting techniques: the use of last-minute data to make predictions for the next one to two hours. Now-casting does not collect any new data; instead, it makes predictions based on current data. The data used includes the latest available from radar, thermal mapping, and weather stations. Now-casting essentially extrapolates current values into a short-term trend.

- Personal observation: with all of todays technology, local supervisor and crew observations are still indispensable in winter maintenance. Unfortunately the manager may not have the opportunity to make these observations and often must rely on other crew members communicating their assessments to keep informed. However, observations are subjective and may be hard to effectively communicate, and the lack of logging makes it hard to qualify, quantify, and store this type of information. RWIS tries to fulfill the need for personal observation by the use of cameras. Nevertheless, it is not a complete substitute for direct observation in the field.

- MDSS: these fuse the various aforementioned information sources to process and create more accurate local weather predictions and new applicable information for winter maintenance. Current research seeks to automate winter maintenance decision making. MDSS items focus on data fusion, improved weather prediction, and decision tools for supervisors with a strong emphasis on data fusion and road condition modeling. Federal research has been conducted to establish data fusion requirements under the Surface Transportation Weather Decision Support Requirements (STWDSR) project [17]. A project in southeastern Michigan implemented a management system for maintenance vehicles which integrates data collection, communication, plowing and spreading monitoring and even automated control [1].

System Interfaces

MRTIS will be used by Caltrans personnel. The primary interface to the System is through software applications running within each SV and within a Caltrans district Central Office.

User Interfaces

MRTIS applications with a GUI interface shall use standardized user interfaces, e.g. standard Microsoft Windows or Linux GNU Network Object Model Environment (GNOME) user interface elements. These user interface elements are common to commodity office software and familiar to the majority of users, which minimizes training expenses. Other applications may be command line oriented.

Hardware Interfaces

Hardware sensors interface to vehicle computers using EIA serial communications recommended standard (RS-232), Universal Serial Bus (USB), or other commodity hardware data buses.

Software Interfaces

Software interfaces between modules and applications use eXtensible Markup Language (XML). XML has broad industry support among tool vendors and open-source projects. XML and related standards are defined and controlled by the W3C standards body. For internal interfaces, see the Implementation chapter (pg. 33) for a list of software products used.

Interfaces between procedural software modules were provided by Simple Object Access Protocol (SOAP). SOAP is based on XML and defined by W3C. SOAP provides an ability to interface modules and applications that may be executing on a single machine or multiple machines. This capability provides some amount of architectural flexibility.

The W3C semantic Resource Description Framework (RDF) and Web Ontology Language (OWL) standards were also used for data exchange. These standards are layered on top of existing XML standards (see Figure A.1 on page 67) and provide the ability to represent complex data and relationships. This is discussed in detail in Appendix A on page 65.

The selected visualization client (see Section 2.2.2 on page 17) uses Keyhole Markup Language (KML) for communication with visualization servers. KML is based on XML, which enables existing XML development tools to manipulate and construct KML.

Communications Interfaces

Communication interfaces to MRTIS are by Transmission Control Protocol / Internet Protocol (TCP/IP), using existing wireless hot-spots via Wi-Fi and additional wireless hot-spots. It is likely that future communication options that supports TCP/IP can integrate with the existing system. Wireless coverage is expected to grow in the future. Existing cellular data coverage (as of December 2006) in the Kingvale area is shown in Figure 2.1 on the next page.

Memory

MRTIS servers and in-vehicle computers use commodity x86 hardware, persistent storage devices, and memory.

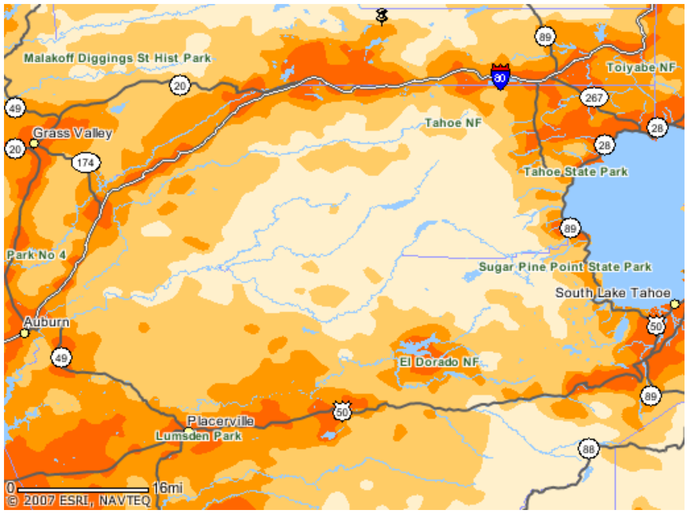

Figure 2.1: Cingular EDGE data network in the Kingvale area, 12/2006 (no coverage in light area)

Operations

MRTIS is expected to incrementally enhance ongoing Caltrans operations. The hardware and software installed within 2V will require no ongoing user interaction. The system installed within supervisor vehicles will use standard commodity software and some specialized knowledge and training is anticipated. If Caltrans hosts the CO, maintenance of the CO server and associated databases will require ongoing database and networking expertise and management consistent with ongoing Caltrans IT operations. At Caltrans' discretion, the CO could also be hosted at AHMCT; in such a scenario, AHMCT would provide ongoing database and networking expertise, maintenance, and management. The visualization component of the SV and CO uses COTS software–minimal user training is anticipated. Where needed, AHMCT could provide training under Deployment Support.

Site Adaptation Requirements

The MRTIS Caltrans Maintenance Equipment Yard (EY) may require installation of additional Wi-Fi base stations. CO software applications run on Commodity Intel, AMD, and compatible microprocessors (x86) servers with Internet connectivity.

Each existing SV will use a commodity x86 in-vehicle computer, Liquid Crystal Display (LCD), keyboard, keyboard tray, and pointing device. A GPS, antenna, and RoadWatch temperature sensor are installed in each SV. A Wi-Fi card or cellular data modem is required. If desired, cellular data services would provide a constant data connection (assuming adequate wireless data coverage) between the SV and CO. Otherwise, in-field Wi-Fi hot-spots may be used. MRTIS communication capabilities are intended to be flexible and adaptable with the use of TCP/IP. Note throughout this report that Microsoft

.Net and C# imply use of MS Windows. However, the open-source project Mono[1] provides a C# compiler and .Net support, and would facilitate migration to other platforms.

Each existing MRTIS Caltrans Secondary Winter Maintenance Vehicle (2V) such as a snowplow, needs sensors (GPS, temperature sensors) and a commodity x86 data collection computer installed. Minimally, a Wi-Fi card is installed; a cellular data modem is optional.

2.2.2 Product Functions

MRTIS functionality can be decomposed into broad categories:

- Visualization client: will place spatial information in context using road maps, terrain, road designators, geographical features (e.g. lakes, rivers), and cities. Key System data elements include Doppler radar, vehicle locations, road temperature, air temperature, traffic data from Vehicle Detector Station (VDS), and weather information (e.g. forecasts). A System goal is to place this spatial information in context to provide meaningful relationships between information types; for example, relationships between historic vehicle locations and winter storm locations provided by Doppler radar. The visualization client will be run within the CO and each SV. Within vehicles, a subset of features will be available depending on network connectivity. Visualization functionality will be available from CO computers capable of running the visualization client with network connectivity to the CO MRTIS server. See Section 3.2.1 on page 27 for information about the selected visualization client. The visualization client will provide the following functionality.
  - Real-time vehicle locations via GPS. See Figure 4.9 on page 49.
  - Historical vehicle locations with different visual queries. See Figure 4.7 on page 47.
  - Real-time air and pavement temperatures. See Figure 4.8 on page 48.
  - Historical air and pavement temperatures with different visual queries. See Figure 4.8 on page 48.
  - Real-time Doppler radar overlays. See Figure 4.4 on page 44.
  - Historical Doppler radar overlays and the ability to provide visual radar loops. See Figure 4.4 on page 44.
  - Existing VAMS Doppler radar images. See Figure 4.5 on page 45, and Figure 4.6 on page 46.
  - Fixed facility locations. See Figure 4.3 on page 43, and Figure 4.10 on page 50.
  - Real-time VDS data. See Figure 4.4 on page 44.

[1]See http://www.mono-project.com

  - RWIS station locations and HTML links (a limited number). See Figure 4.11 on page 51.
  - Existing traffic-cam locations and HTML links.

- Sensor data collection: snow supervisors use a wide variety of data for winter maintenance operations. MRTIS will use vehicle-borne sensors including GPS, road temperature, and air temperature sensors. Both vehicle types (SV, 2V) will collect, store, and forward sensor data to the CO. Data is forwarded when network connectivity is available. External data is provided through the CO Internet connectivity. This includes Doppler radar and real-time traffic.

- Sensor data storage: CO server and applications are responsible for storing collected sensor data. In addition, they provide server visualization functionality for clients.

- Communication components: both hardware and software components provide data communication between applications within each SV, 2V, and the CO. A combination of Wi-Fi and/or cellular data modems, combined with applications using reliable messaging functionality is used. For example, it is anticipated that each SV and 2V returning to the CO will forward their collected sensor data via a Wi-Fi base station installed in the EY in close proximity to vehicle parking locations.

2.2.3 User Characteristics

MRTIS users are anticipated to have typical office automation skills and familiarity with commodity desktop operating systems and office productivity software. This skill set is anticipated to be sufficient for operating the MRTIS visualization client (see Section 2.2.2 on the previous page).

2.2.4 Constraints

System constraints include:

- Cellular data network coverage: the availability of wireless hot-spots within the field, and the availability of Internet connectivity within the CO is limited. See Figure 2.1 on page 16.

- Wireless 900 MHz radios have been identified as a potential communications medium. However, the TAG has identified this medium as very slow and unusable for data transfer purposes; in addition, 900 MHz radio is not available at the main test location. On the other hand, in some environments, 900 MHz may outperform other options, such as General Packet Radio Service (GPRS), and should be considered as a viable option.

2.2.5 Assumptions and Dependencies

In general, it is assumed that wireless connectivity is available at a minimum within the EY. It is also assumed that over time (long-term), wireless connectivity will generally increase in availability and decrease in cost (see MDSS trends, pg. 7). It is also assumed that over time (long-term), sensors such as GPS will decrease in cost. These trends are driven by advances in semiconductor process technology. Real-time tracking of vehicles is possible only with constantly connected wireless services. This would most likely require installation of in-vehicle cellular data modems and the associated subscriber service.

2.3 Specific Requirements

This section lists functional and non-functional requirements, which were used in the subsequent design and implementation.

2.3.1 External Interface Requirements

This section specifies requirements specific to MRTIS interfaces with the outside world. The outside world includes existing external systems which are used by MRTIS and future systems which would be desirable to integrate MRTIS with. For example, this latter category likely includes RWIS, and perhaps additional vehicle-based sensors such as road surface friction sensors. In general the developed framework should support importing various types of images that might be provided by different VAMS.

User Interface Requirements

Requirement 1: The user interface will visually integrate data from a variety of data sources. Visual integration with a terrain model would be beneficial for visualizing potential interactions between winter maintenance factors such as weather, road temperature, elevation, etc.

Requirement 2: Software application ease-of-use is an important non-functional requirement and positively impacts initial and ongoing costs, including training. Project applications will be as easy to use as possible, balanced with System capability requirements.

Requirement 3: The System shall, to the extent possible, balanced with capability requirements, use and design applications with a consistent and common user interface (GUI), for example, Windows, GNOME, etc.

Requirement 4: The System shall use local time for display purposes.

Requirement 5: The System shall allow the user to specify the desired date and time of archived data to view.

Hardware Interface Requirements

Requirement 6: Hardware interfaces with sensors shall use standards common to commodity x86 hardware, such as RS-232, USB, Ethernet, etc.

Requirement 7: MRTIS hardware interfaces with sensors used in the future shall use standards common to commodity x86 hardware, such as RS-232, USB, Ethernet, etc.

Software Interface Requirements

Requirement 8: MRTIS will use public standards such as XML, SOAP, RDF, OWL, HTTP, to provide data and procedural functionality to existing and future applications that interface with MRTIS.

Communications Interface Requirements

Requirement 9: Data communication between MRTIS applications and applications across a network will use TCP/IP.

2.3.2 Functional Requirements

Requirement 10: The visualization application will display spatial information in context using road maps, land marks (e.g. lakes), point data, poly-lines, overlaid digital pictures (e.g. Joint Photographic Experts Group (JPG)), and HTML (including links).

Requirement 11: The visualization application will display historical vehicle path resulting from archived GPS sensor data.

Requirement 12: The visualization application will display roadway temperature.

Requirement 13: The visualization application will display weather radar images of the region of interest (here, Northern California) collected from an Internet source.

Requirement 14: Sensor data should be traceable to its source, including vehicle, time, and date.

Requirement 15: SV applications shall send collected sensor data to the CO when a connection is available.

Requirement 16: SV applications shall request new System data from the CO when a connection is available.

Requirement 17: SV applications shall provide visualization capabilities of System data elements for vehicle operators.

Requirement 18: System visualization capabilities include vehicle locations, Doppler radar, and real-time traffic.

Requirement 19: The SV shall communicate with the CO using Wi-Fi and optionally cellular data modems.

Requirement 20: The SV will contain a computer powerful enough to display visual information for the user, and perform associated processing and interfacing with sensors.

Requirement 21: The 2V shall send collected sensor data to the CO.

Requirement 22: The 2V shall communicate using Wi-Fi.

Requirement 23: The 2V shall contain a computer to collect and communicate sensor data.

Requirement 24: The CO shall forward requested System data to each SV.

Requirement 25: The System shall support multiple vehicles, of type SV or 2V.

Requirement 26: The System shall identify vehicles uniquely by name.

Requirement 27: The System shall identify vehicle-born sensor data by vehicle name.

Requirement 28: The System shall allow specification of the location of fixed Caltrans facilities within the System mapping display.

2.3.3 Performance Requirements

Requirement 29: The System shall be designed to allow for reasonable increases in the number of users, sensors, services, and vehicles.

Requirement 30: The System shall support, to the extent possible, a modular structure with the goal of module separation across different machines for load balancing.

Requirement 31: Round trip KML server response time will be less than 5 seconds and ideally less than 1 second (ignoring communication overhead).

Requirement 32: COTS hardware, especially commodity hardware, will be used wherever possible so that System costs (initial and ongoing) are minimized.

Requirement 33: COTS software, especially commodity and open-source software, will be used wherever possible so that System costs (initial and ongoing) are minimized.

2.3.4 Logical Database Requirements

Requirement 34: The selected database should support spatial indexing for spatial data (e.g. GPS). This requirement is a result of R31.

Requirement 35: The System shall archive data collected from sensors and Doppler radar images.

2.3.5 Design Constraints

Requirement 36: If a temperature sensor is installed in a vehicle, a GPS must also be installed.

Requirement 37: To the extent possible, MRTIS will provide a framework for integration of future data sources and information. For example, RWIS station data, VAMS. An example of VAMS use is the Meteorlogix weather information sources used by the existing Caltrans District 3 snow fighting operations.

Requirement 38: The System shall be designed with standard software engineering practices, tailored for a prototype capability.

Requirement 39: The System will provide a framework for vehicle-to-vehicle communication which may become a crucial system feature in the future.

Standards Compliance

Requirement 40: Hardware for data communication between the CO and vehicles shall support TCP/IP.

Requirement 41: Software responsible for data communication between the CO and vehicles shall use TCP/IP.

2.3.6 Software System Attributes

Reliability

Requirement 42: Collected sensor data should be handled in a fault-tolerant manner by messaging applications. For example, using transactions to increase reliability and decrease the probability of losing collected sensor data—hence transactions are avoided.

Requirement 43: System computers installed in the CO, SV, 2V shall have the ability to synchronize clocks with a standardized time source.

Requirement 44: The System shall internally use Coordinated Universal Time (UTC)/ Greenwich Mean Time (GMT) to represent time.

Availability

Requirement 45: The 2V installed hardware system (computers, sensors) will operate with little or no user intervention, to the extent possible.

Security

Requirement 46: In a production environment, the System design will support implementation of features that prevent unauthorized use of the System. For example, encryption of wireless data, selective Internet Protocol (IP) access to web services, etc.

Requirement 47: In a production environment, the System design and implementation will prevent it from being disabled from all access points either accidentally or intentionally.

Requirement 48: In a production environment, access to system servers shall be controlled using IP blocking.

Maintainability

Requirement 49: The System will be implemented using higher-level languages and standard software engineering practices.

Requirement 50: The System shall be implemented in English.

Portability

Requirement 51: To the extent possible, the System shall be implemented to be portable across operating system platforms. This may not always be possible due to performance and cost trade-offs.

2.3.7 Other Requirements

Requirement 52: The System shall display temperatures in Fahrenheit.

Chapter 3

Design and Architecture

This chapter describes the MRTIS design and architecture. The design uses the specified requirements of Chapter 2 on page 11 to create a blueprint for the implementation. Specific requirements are referenced where appropriate (e.g. (R1)). The implementation of the design is discussed in Chapter 4 on page 33. The general design approach was to specify a prototype system that provides broad MDSS capabilities and a number of specific features within this larger framework. A key part of the MDSS vision is integration and analysis of inbound data streams. This capability depends on providing a platform based on open standards. To support this higher-level design goal, public standards were used such as web services (SOAP), non-proprietary messaging formats that use RDF and OWL, commodity hardware, and commodity and open-source software.

The visualization client is a key system component and is the primary interface for users. The specific visualization client selected is discussed in Section 3.2.1 on page 27. General requirements for the visualization client are discussed in Section 2.2.2 on page 17. In addition, the project design integrates the following within the visualization client:

- Real-time vehicle locations via GPS (see Figure 4.9 on page 49).
- Historical vehicle locations with different visual queries (see Figure 4.7 on page 47).
- Real-time air and pavement temperatures (see Figure 4.8 on page 48).
- Historical air and pavement temperatures with different visual queries (see Figure 4.8 on page 48).
- Real-time Doppler radar overlays (see 4.4 on page 44).
- Historical Doppler radar overlays and the ability to provide visual radar loops (see 4.4 on page 44).
- Existing VAMS Doppler radar images (see Figure 4.5 on page 45, and Figure 4.6 on page 46).

- Fixed facility locations (see Figure 4.3 on page 43, and Figure 4.10 on page 50).
- Real-time VDS data (see Figure 4.4 on page 44).
- RWIS station locations and HTML links (see Figure 4.11 on page 51).
- Existing traffic-cam locations and HTML links.

3.1 Architectural Design

Relationships between major structural elements are discussed here. In the broadest sense, the system collects data from external sources and sensors; stores, integrates, and analyzes the data; and visually presents derived information to users (see Figure 3.1). MRTIS links a number of independently operating vehicles with a central office (see Figure 3.2 on the facing page). The system consists of three independent and communicating logical units: the CO, SV, and 2V (see pg. 12). These components are shown in Figures 3.3 on page 28 and 3.4 on page 29. Key design features were used in these units that influenced other design choices. These include transaction-oriented messaging, non-proprietary message encoding, a visualization client, wireless TCP/IP communication, commodity server hardware, open-source and COTS applications, commodity sensors (GPS, road temperature, and air temperature). Implementation choices often affect design choices. For example, the choice of the visualization client (Google Earth©, a KML browser) determined how historical information was presented to the user.

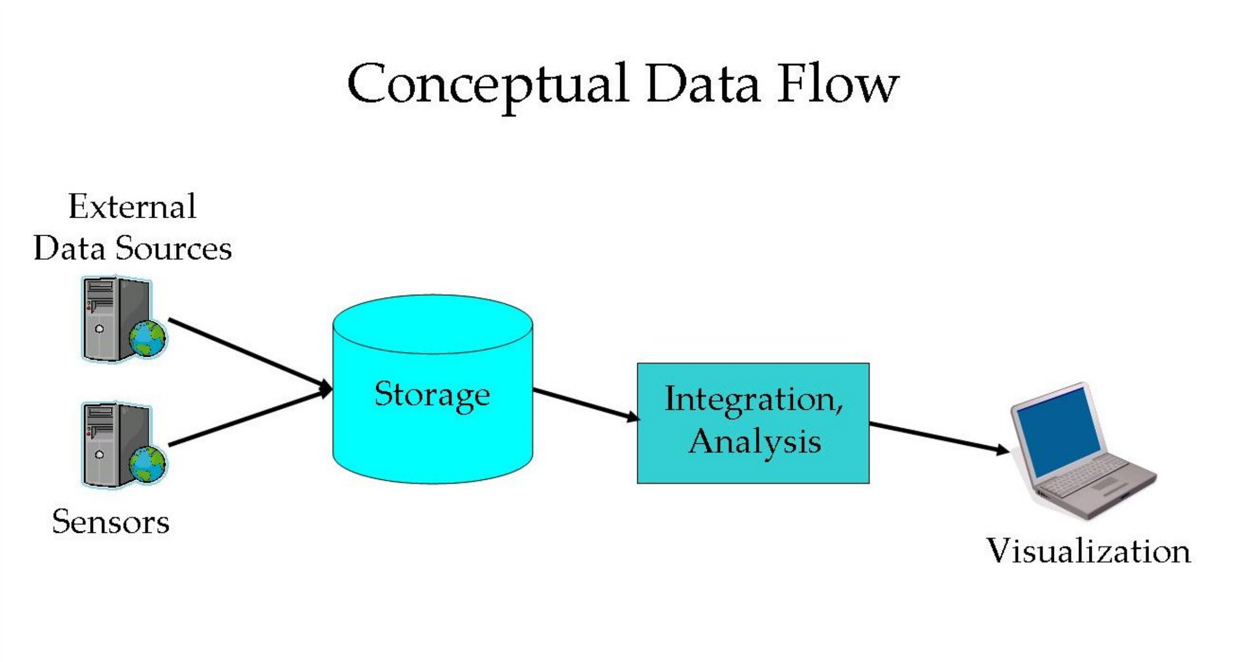

Figure 3.1: Conceptual data flow

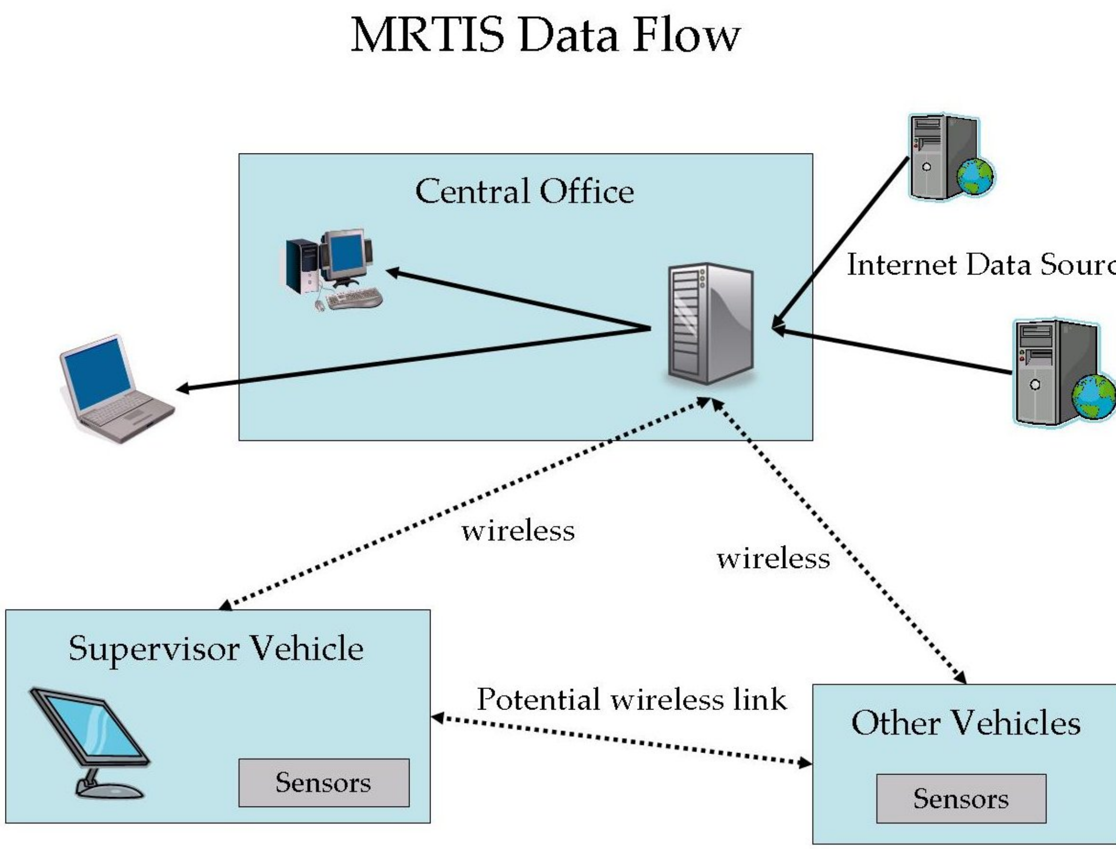

Figure 3.2: MRTIS Overall project data flow

3.2 Interface Design

This section provides a description of information flow between components, systems, and actors.

3.2.1 Human Machine Interface

The primary system interface for users is the visualization client. Google Earth©was selected as the visualization client and was identified as a new class of visualization application as distinct from traditional GIS clients, which emphasize complex spatial analysis and tools over ease of use and speed. Other applications in this new category include ESRI's ArcGIS Explorer © and NASA's World Wind. Applications in this new category can be identified as a KML browsers, reflecting an analogous relationship as that between the HTML browser and HTML server. Essentially, KML browsers enable viewing and integrating spatial data in the same way HTML browsers enabled viewing and integrating text and images within the World Wide Web. KML provides a similar data syntax stan-

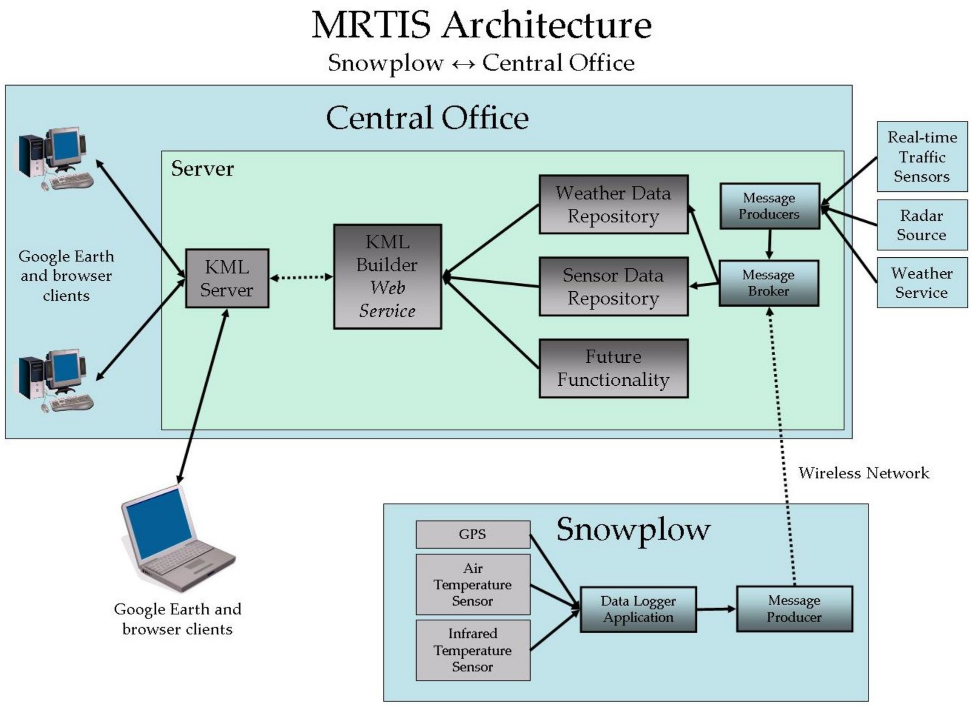

Figure 3.3: MRTIS Architecture, Central Office

dard compared to HTML. In addition, the following also motivated the use of Google Earth© as a visualization client:

- It is customizable, with the ability to add additional imagery, vector data, and layers,
- It supports HTML and integration with web pages, enabling integration of visual elements and applications across the networked servers,
- Visualization servers can be developed to provide specific customized information, both to System users and other Caltrans users,
- KML visualization servers developed by others can be integrated as desired,
- It meets specific user interface requirements (see Section 2.3.1 on page 19),
- It provides the functionality discussed in Section 2.2.2 on page 17,
- It uses an XML (in the form of KML) data format, which is used by existing tools and is generally supported by the project (see Section 2.2.1 on page 15),
- It is relatively simple to use and requires little, if any, training. The lack of spatial analysis features supported by traditional GIS products was considered a benefit.
- It is undergoing a rapid rate of evolution and enhancement. This includes both increases in image quality and availability and the addition of new features, such as the built-in time line,

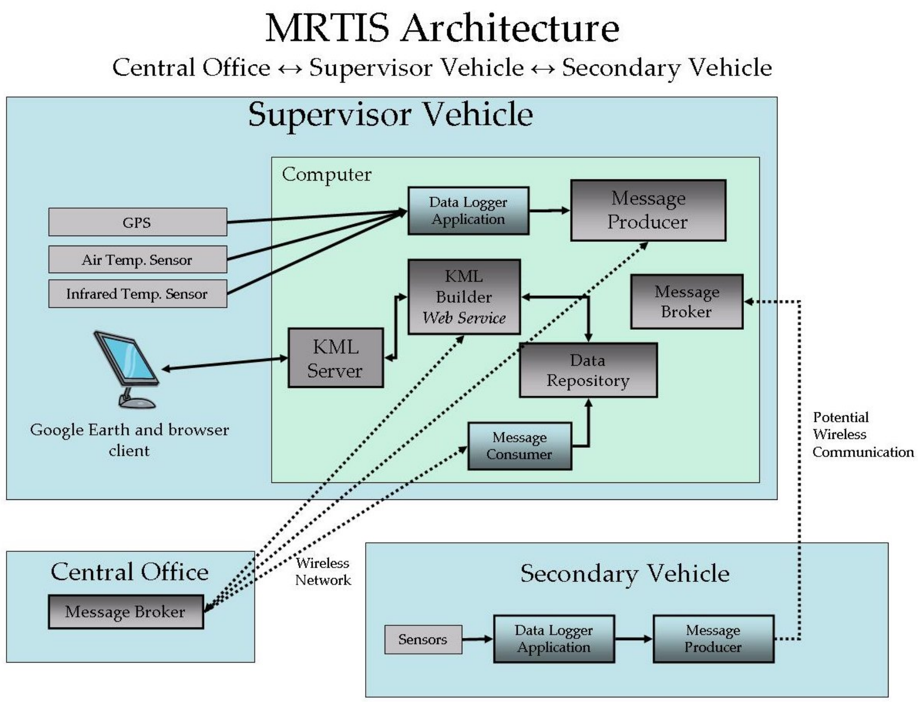

Figure 3.4: MRTIS Architecture, Supervisor Vehicle

- It is inexpensive commodity software compared with traditional GIS software,
- It includes a built-in road network, reducing development effort in this area.
- It provides a structured UI that is consistent across platforms (Linux, Microsoft Windows, etc.). For example, layers may be turned on or off, place-marks clicked, and external Internet browsers provide an HTML interface.
- It provides a time dimension control, which enables both query and visualization with respect to time.

The visualization computer mounted inside the SV uses a high-contrast LCD display with a sunlight shield. The keyboard contains an integrated pointing device. The computer starts and stops when the vehicle is turned on and off. The sensor data logging computer installed inside the 2V requires no routine user interaction.

3.2.2 Interface Between Components

Interfaces between components are of different types:

- Binary Microsoft .Net Components: used by code written in C++ and C#. Note throughout this report that Microsoft .Net and C# imply MS Windows. However, the open-source project Mono provides a C# compiler and .Net support, and would facilitate migration to other platforms.

- Binary Java Components: use by code written in Java.
- Web Services: some web services are called from PHP Hypertext Preprocessor (PHP) as a result of KML browser requests. In this case, data is exchanged via the standard SOAP interface. In general, the KML visualization server code is structured to be broken apart as web services for modularity and scalability requirements. These interfaces use web service-defined data structures.
- Hardware Interfaces: use of COTS is highly desirable and interfaces such as USB, RS-232, Ethernet, etc. are used wherever possible.

3.3 Application-Level Design

MRTIS consists of a number of cooperating applications with specific functionality. Each application is running on the central office server, within a vehicle, or both. These applications and modules are described below.

- Database Server: a database server is used within the SV and CO. The ability to handle spatial information is an important feature. Real-time and historical sensor data is stored in the database.
- Messaging Broker: a messaging broker is used with in CO. It uses transactions for reliability and stores and forwards messages to message consumers and producers. A non-proprietary message format is used: XML, and RDF (a form of XML).
- Sensor Reader Module: this module is responsible for interfacing with in-vehicle sensors, initializing the sensors, collecting the sensor data, and reliably forwarding the data over the wireless network to the central office server or other vehicles. It is used on the SV and 2V.
- Sensor Reader Configuration Module: this module provides the ability to specify sensors used by the sensor reader module. This includes configuration-related information such as ports, speeds, data formats, etc. It is used on the SV and 2V.
- Synchronization Consumer Module: this module is responsible consuming and processing messages that contain sensor data. The module reads messages from a message broker within the CO using TCP/IP. Reliable transaction-oriented messaging is used. It is used on the SV, CO, and 2V.
- Synchronization Producer Module: this module is responsible producing messages that contain sensor data. The module forwards this data to a message broker within the CO using TCP/IP. Reliable transaction-oriented messaging is used. It is used on the SV, CO, and 2V.

- Visualization Client: a KML browser (Google Earth©) is used as a visualization client. This has a number of implications. For example, it has the ability to integrate photos, HTML content, spatial elements (lines, points, surfaces), ground overlays, and screen overlays. In addition, the ability to associate each visual element with the time dimension influenced other design decisions. Google Earth© has been used elsewhere as an excellent weather visualization client [21].

- Visualization Server: a visualization server provides spatial information to the visualization clients. It is used within the CO and SV. The visualization server receives requests from the visualization client specific to each layer. Some requests are dynamic (a function of viewer position above the ground) and some requests are static. Dynamic requests are time-sensitive and must return results in a short period of time, usually around two seconds. The visualization server must query the database, format the results as KML, and return the results.

- genkmltrafficreport: this application generates a 5-minute static Keyhole Markup Language Zipped (KMZ) traffic report for the visualization server.

- httpImageCapture: this application captures an image (e.g. JPG) from a specific Uniform Resource Locator (URL) on a periodic basis and stores it in the local database. For example, Doppler Radar images are captured using this application.

- imageConsumer: this application forwards captured images to the messaging broker.

- RDFLoader: this application loads RDF and OWL objects into the database.

3.4 Library Design

The MRTIS project makes extensive use of code libraries which contain class definitions, interfaces, and static methods. Libraries are listed below:

- Common Library: this library contains classes and static methods common to all project code. For example, common classes include: Angle, AngleFilter, AppLog, CircularByteBuffer, ConfigFile, Contract, Filter, HeartBeat, MinMaxDouble, PositionFilter, Registry, SignalFilter, Speed, Temperature, among others. Many of these classes contain their own unit test methods.

- CommonActiveMQ Library: contains classes common to all messaging code.

- InfrastructurePlatform Library: contains classes common to code specific to asset management.

- jenaCommon Library: contains code and classes common to the semantic web functionality.

- Sensor Reader Library: this library contains common data structures and procedural code shared between the sensor reader application and configuration application.
- Synchronization Library: this library contains common data structures and procedural code shared between the synchronization client and server applications.
- ontology: this library contains W3C semantic web definitions used for the messaging formats and semantic objects stored in the database.

3.5 Data-Level Design

In general a highly object-oriented design was used. Classes are defined for physical quantities, e.g. Angle, Speed, and Temperature. Container classes are also used, e.g. Sensors as a group of Sensor objects. Class hierarchies are used. Containment is preferred to deep hierarchies. An example of a class hierarchy, starting from the root: Sensor, SerialDevice, GPS, NovAtel. The system design supports the addition of any number of sensors and specialization of existing sensors.

Semantic modeling is used for objects passed through the message broker. These include OWL definitions of: GroundOverlay, StationaryThing, and VDS with associated properties such as Speed, Occupancy, and Flow.

A number of web-service related data structures are defined such as: VehSynchData, VehsSynchData, VehPosition, VehPositions, VehsPositions, VehsPositions, Coordinate, GroundOverlay, and LatLonBox.

Chapter 4

Implementation

This chapter covers MRTIS implementation details. The implementation executes the design and fulfills specified requirements. The implementation consists of software and hardware components installed within vehicles and the central office. MRTIS uses commodity hardware and software (R32, R33). Note that, e.g., (R32) refers to a specific requirement defined in Section 2 on page 11.

4.1 Hardware

MRTIS hardware is responsible for:

- Providing sensor data to software applications
- Executing software applications running within vehicles and the central office
- Communicating between computers in vehicles and the Central Office

A summary of MRTIS hardware is shown in Table 4.1 on the next page. As indicated in the table, each piece of hardware is installed in one of three locations: within the central office, within supervisor vehicles, or within secondary vehicles such as snowplows.

4.1.1 Vehicle Hardware

Supervisor vehicles contain sensors, a cellular modem, and a computer used for visualization of data. Sensors consist of a GPS receiver (see Figure 4.1 on the following page) and the RoadWatch temperature sensors (see Figure 4.2 on page 35). The GPS and RoadWatch sensors use RS-232 serial communication. The GPS receiver communicates with the in-vehicle computer using National Marine Electronics Association GPS communications standard (NMEA-0183). Secondary vehicles contain sensors, a computer for data collection, and a Wi-Fi wireless card. Sensors include GPS and road and air temperature sensors. Secondary vehicles can use Wi-Fi cards and cellular modems if desired. All developed software uses TCP/IP to communicate with other machines, eliminating dependence on specific communication hardware devices. Supervisor vehicles can also use Wi-Fi if desired.

Figure 4.1: GPS sensor

4.1.2 Central Office Hardware

Central office hardware consists of commodity x86 servers. These servers provide software services using KML, HTTP, HTTP web services, Java Messaging Service (JMS), and others.

Table 4.1: Implementation hardware

#	Hardware	Location	Description
1	GPS Receiver	Vehicles	Location sensor, NMEA-0183 compatible
2	RoadWatch SS	Vehicles	Air and pavement temperature sensors
3	Computer (x86)	Secondary vehicles	Collects and forwards sensor data
4	Wireless card	Secondary vehicles	Wi-Fi wireless card
5	Computer (x86)	Supervisor vehicles	Supports visualization applications
6	Cisco mobile router	Supervisor vehicles	Mobile router bidirectional acIP
7	Cell modem	Supervisor vehicles	Wireless IP communication
8	Computer (x86)	Central office	Executes applications, collects data

Figure 4.2: RoadWatch air and pavement temperature sensor

4.2 Software

MRTIS is implemented using off-the-shelf commodity software combined with custom-developed applications. Together, these applications:

- Collect and store data over the Internet from sensors installed in vehicles
- Provide reliable communication and messaging between machines
- Provide client and server visualization functionality

4.2.1 Off-the-shelf Software

Off-the-shelf software packages were used wherever possible. These packages execute on the x86 hardware specified above and communicate with sensors.

- Application development libraries:
  - The CSharpZipLib library was used to provide Zip compression within C# applications.

  - The Sax.Net CommStudio communications library was used to communicate with sensor hardware via RS-232 within C++ and C# applications.
  - Sun's Java Advanced Imaging (JAI) was used to manipulate images within Java applications.

- Databases:
  - PostgreSQL was used by Jena (see below) for persistent object storage.
  - Microsoft SQL Server was used to store vehicle GPS, road, and temperature sensor data.

- HTML server: The Apache Foundation's Apache HTTP Server is used on the Linux server to serve static KML and web pages. Microsoft's IIS HTTP server is used on Windows XP.

- KML browser: Google Earth© 4.0 and KML 2.1 are used as a visualization client in the vehicle and within the central office. For example, see Figure 4.3 on page 43.

- Messaging platform: Apache ActiveMQ is used to provide reliable topic-based messaging between computers, both in-vehicle and the central office. Messaging reliability is a key requirement (R42).

- Operating systems:
  - Red Hat Fedora Core 6 Linux was used as a server operating system.
  - Microsoft Windows XP was used for the in-vehicle operating system and as a server operating system.

- Procedural languages and platforms:
  - Bash shell scripting was used for starting and stopping some server applications running on Linux.
  - The JBoss Application Server was used to provide Java Platform, Enterprise Edition (J2EE) functionality.
  - Microsoft's Visual Studio 2003 with C# and C++ were used on the Windows XP platform.
  - PHP was used on the Windows XP platform as an interface between the HTTP interface for the KML server and the C# applications. The PHP NuSOAP toolkit was used to call web services from within PHP code.
  - Sun's Java and the Eclipse Foundation's Eclipse development environment were used to develop applications that execute on the in-vehicle and central office computers.

- Semantic definition language: W3C Web Ontology Language Description Logics (OWL-DL) was used to define semantic classes, properties, objects, and relationships between them.

- Semantic editor: Stanford's Protégé was used to create and edit semantic definitions in OWL-DL.
- Semantic framework toolkit: Hewlett Packard's Jena was used to access and manipulate semantic objects.

4.2.2 Custom-Developed Software

The following applications and modules were developed.

Bash (Bourne-again shell, the Linux command-line shell) Scripts

- cleanJenaDB.sh: removes all classes, objects, and properties from the PostgreSQL database.
- consumeImages.sh: calls imageConsumer.
- getAccuweatherimages.sh: calls httpImageCapture.
- jb-show_deployed-files.sh: displays deployed Java server files.
- loadOWL.sh: loads OWL-DL semantic definitions into Jena semantic persistent storage.
- runjboss.sh: starts the JBossAS server.
- set-env-jboss-servlet.sh: specifies JBossAS environment variables.
- showall.sh: displays running Java servers.
- startall.sh: starts all servers, consumeImages, getAccuweatherimages, and startTraffic.
- startTraffic.sh: runs continuously, calls gettraffic.pl, LoadTraffic, GenKMLTrafficReport.
- updateBinFromDev.sh: housekeeping of binary files.
- updateBinServletsFromDev.sh: housekeeping of binary files.
- updateJBossServletsFromBin.sh: housekeeping of binary files.

Perl Scripts

- gettraffic.pl: retrieves latest real-time traffic information from traffic source and writes to a file.

C# Applications

- SensorReaderApp: initializes sensors, reads sensor data, and writes sensor data to files. On the supervisor vehicle GPS data is forwarded in real-time to a running web service. This application runs on supervisor and secondary vehicles. Configuration data is read from an XML file on start-up.

- SensorReaderConfigApp: enables editing of the associated XML configuration file which is used to specify sensor communication details, directory locations, and other program configuration data.

- SynchServerApp: reads files downloaded by MRTISVehConsumer and forwards them to the appropriate web service. This application runs on the supervisor vehicle and central office computer.

C# Shared Modules

- App: root Application class and functionality.
- Common: cross application functionality and classes.
- FastMap: 2D mapping functionality.
- MRTISCommon: functionality common to MRTIS applications.
- Sensor: base sensor class.
- SensorDL: base sensor data logging class.
- SensorReaderLib: functionality common to the sensor reader and configuration applications.
- Sensors: sensor container class.
- SensorCPP: sensors implemented in C++.
- StatusLight: indicator light.
- SynchCommon: common synchronization functionality.
- Vehicle: base vehicle class.
- VehicleCPP: vehicle functionality in C++.
- ZipLib: compression library.

Java Applications

- genkmltrafficreport: generates a KML file containing the latest real-time traffic data extracted from persistent storage. It is written in Java and runs on the central office server and supervisor's computer.
- httpImageCapture: continuously captures images from a specified URL and encodes them as XML OWL-DL objects. As a message producer, these messages are forwarded to an ActiveMQ consumer which is typically located on the central office server.
- imageConsumer: continuously performs the following:
  1. Connect to the ActiveMQ broker retrieving messages which are OWL objects of class GroundOverlay.
  2. The OWL individuals are saved persistently in the local Jena database.
  3. A KMZ file is generated which contains timestamped radar images as ground overlays.
- loadtraffic: loads a file containing real-time traffic data into the persistent database.
- MRTISVehConsumer: continuously connects to the specified JMS broker ActiveMQ message consumer, typically running on the central officer server and supervisor vehicle. Retrieved messages are stored to the local file system in a specified directory. A durable message subscription is used.
- MRTISVehProducer: a JMS message producer that runs on vehicle computers and continuously connects to the specified broker, sending files that appear in the specified directory.
- RDFLoader: loads a specified OWL file into the persistent database.

Java Shared Modules

- InfrastructurePlatform: contains common classes AssetMan, LatLonBox, Point.
- ontology: contains Jena generated class definition information.
- Common: cross application functionality and classes.
- CommonActiveMQ: common JMS messaging functionality.
- jenaCommon: common Jena functionality.

PHP Scripts

- kmlserver-doppler.php: KML server HTTP address for Doppler radar images.
- kmlserver-loc24h.php: KML server HTTP address for vehicle locations.
- kmlserver-loccur.php: KML server HTTP address for real-time vehicle locations.
- kmlserver-tmp.php: KML server HTTP address for temperature sensor.
- kmlserver-wstat.php: KML server HTTP address for weather station data.

4.3 System Installation and Maintenance

Routine maintenance is expected after installation, consistent with normal office automation hardware and software. This includes commodity database maintenance of the SV and CO systems, and potential system upgrades. System installation can be broken down by component:

- MRTIS Caltrans Winter Maintenance Supervisor Vehicle (SV): requires the following components to be installed: GPS, GPS antenna, RoadWatch road and air temperature sensor, computer, LCD, and keyboard. The computer is expected to contain the Wi-Fi IEEE wireless LAN/WAN standards (802.11) card and cellular modem.
- MRTIS Caltrans Secondary Winter Maintenance Vehicle (2V): requires the same installation as the SV, however the vehicle driver will not use the LCD, keyboard, and computer as part of operations, which may ease installation.
- MRTIS Caltrans Maintenance Central Office (CO): requires the installation of software servers connected to the Internet. These may be located anywhere and consist of commodity hardware servers running database applications.
- MRTIS Caltrans Maintenance Equipment Yard (EY): requires installation of 802.11 wireless base station within wireless range of vehicle parking areas.

4.4 Implementation and Design Issues

This section covers issues that were encountered during the design and implementation phase. The goal is to provide valuable hands-on information that may be useful for future development and design:

- Database storage of GPS data: collection of in-vehicle GPS data generates a large amount of data points rapidly. Storing GPS data for a large number of vehicles over a long period of time requires careful consideration. Implementation of policies to limit storage of GPS data to certain time boundaries, e.g. not older than two years, etc. would probably be crucial. Database indexing is also an important issue for reasonable query response time. It was found that indexing GPS data by date, location, and resolution was crucial. Spatial data is typically stored within tree data structures, which provide O(log n) performance for spatial queries. Both MySQL and PostgreSQL (with PostGIS) support spatial data. This approach is very effective when a modest number of points are within the spatial search extents. When many data points are within the search extents, other indexes must be used to eliminate points not in the result set. This is typically the case for vehicle generated data– many data points over time fall within the same roads, making spatial tree searches less effective. Indexing by time and resolution were used to improve database performance. Indexing by GPS point resolution was found to be particularly important. This involved features in both the client and server. The client (the application executing in the vehicle, collecting GPS data and forwarding it to the server) calculated the distance (in meters) from the last generated GPS point. This distance (the resolution) was stored together with the date, latitude, longitude, and other GPS point information and forwarded to the server for storage in the database. Subsequent database queries used a resolution index to eliminate points not in the result set. The resolution parameter used in the query was calculated based on the KML viewer's height above the ground and viewer size (in pixels). For example, if a vehicle sat in one position for an extended period of time, numerous GPS points would be generated for the same (or nearly the same) point, all with a very small resolution. These points would be efficiently eliminated from the query result set using a resolution parameter in the Structured Query Language (SQL) query.

- Line simplification: it was found that line simplification was crucial for displaying GPS data within the KML viewer. GPS data is both very dense and can be viewed (via the KML viewer) from low elevations (50 ft) and very high elevations (30,000 ft and more). Line simplification eliminates unnecessary vertices in poly-lines as a function of the desired accuracy in the resulting line segment. Accuracy was calculated as a function of the viewer's height above the ground and window size. Douglas-Peucker line simplification was used.

- Messaging Broker: the use of the messaging broker greatly simplified development of code related to reliable forwarding of data to vehicles and from vehicles to the central server. The durable message subscription model worked well. The support of a store-and-forward model between different vehicles and the central server is possible with the messaging broker. Apache's ActiveMQ is a young product and bugs were encountered that slowed development. The developers hope that ActiveMQ will continue to evolve and mature. There are a number of open-source messaging products available as alternatives. Testing and validation of the selected messaging product would be important for production level use.

- Real-time generation of KML: was used for KML browser updates, which require a database query of historic vehicle paths. This is computationally demanding and the server load is directly proportional to the number of simultaneous users. Increasing scalability might be important in the future. This could be achieved by caching query results as a function of viewer location, altitude, and time. However, this approach might be less useful due to the real-time dynamic nature of the collected GPS data.

- Semantic modeling: the use of semantic modeling was found to be valuable. However, there is concern about the additional complexity introduced by the use of semantic toolkit and models. This would be a particularly important issue in a production system. Integration of semantic modeling with procedural languages will be an important development and might offset the complexity burden of using current semantic modeling tools. Similar benefits may be obtained by using standard XML and additional tools. Reliability problems were uncovered with the HP Jena semantic modeling toolkit.

- Consideration of future wireless standards: the potential use of Worldwide Interoperability for Microwave Access (WiMAX) in the future would impact the direction and design of follow-up projects.

- Data file format standardization: the use of XML was helpful and valuable during development. For example, this enabled the use of XML tools such as eXtensible Stylesheet Language and Transformation (XSLT) to manipulate data with relative ease.

4.5 System Screen Shots

System screen shots are shown below and correspond to the list of desired visualization features detailed in Section 2.2.2 on page 17.

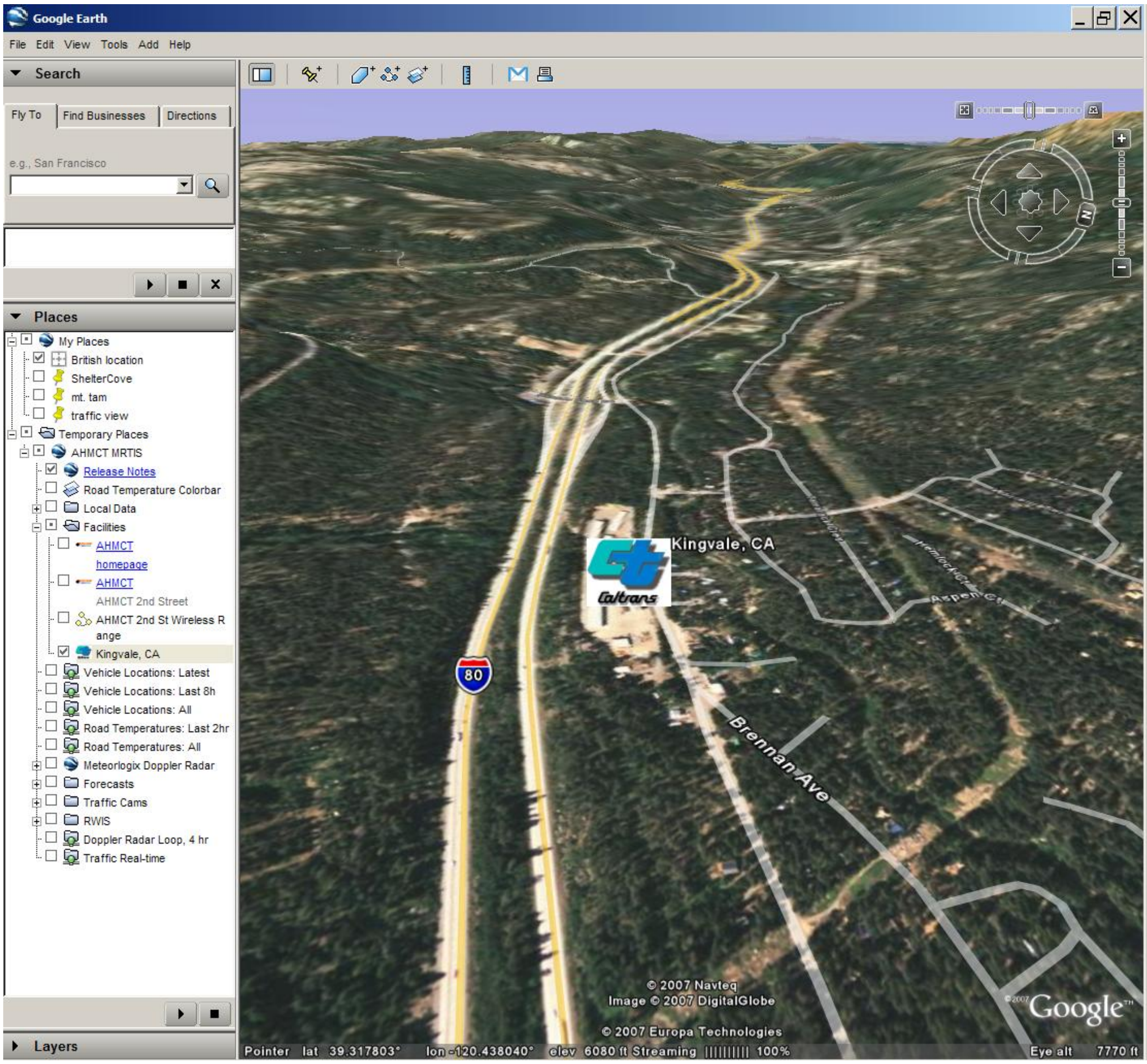

Figure 4.3: MRTIS System Screen: Kingvale facility location, static spatial data

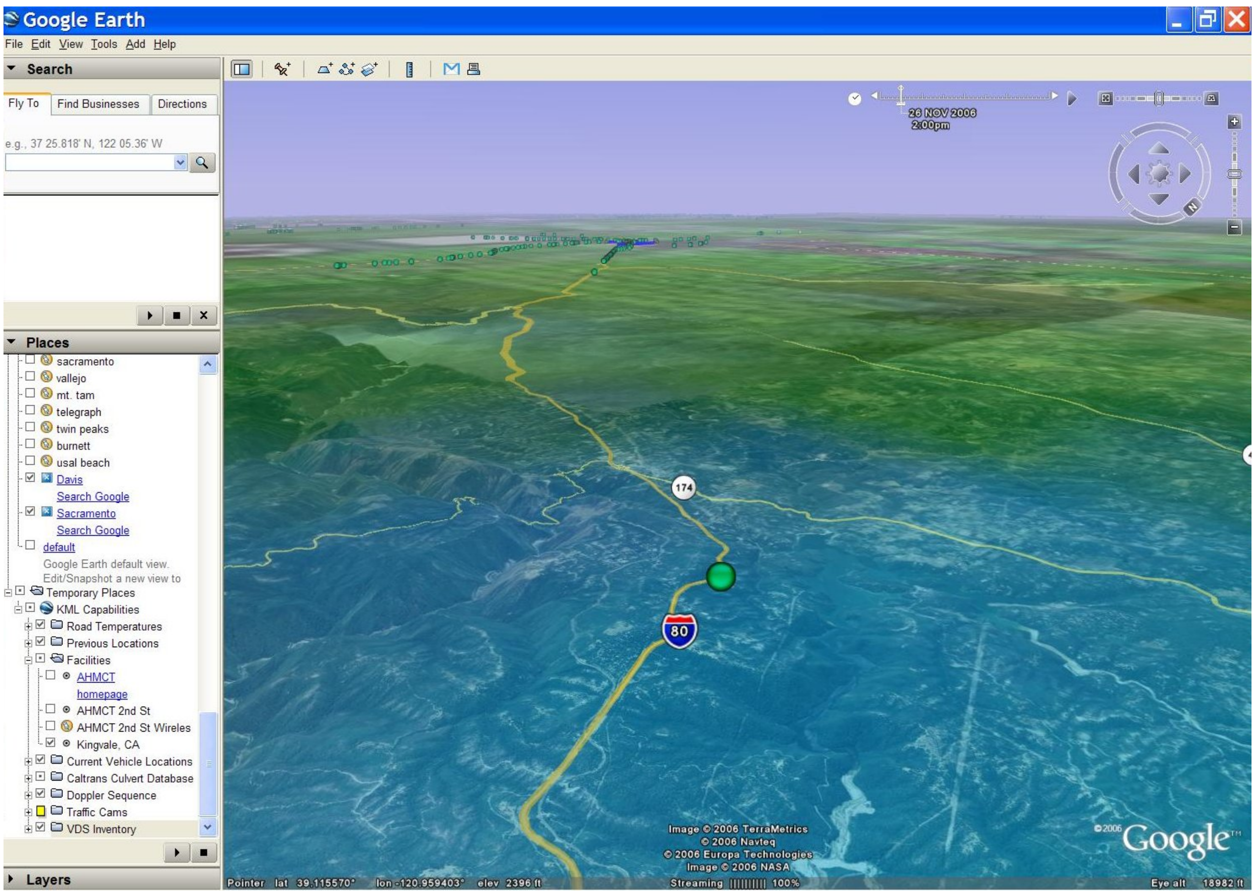

Figure 4.4: MRTIS System Screen: VAMS Doppler radar loop and VDS stations

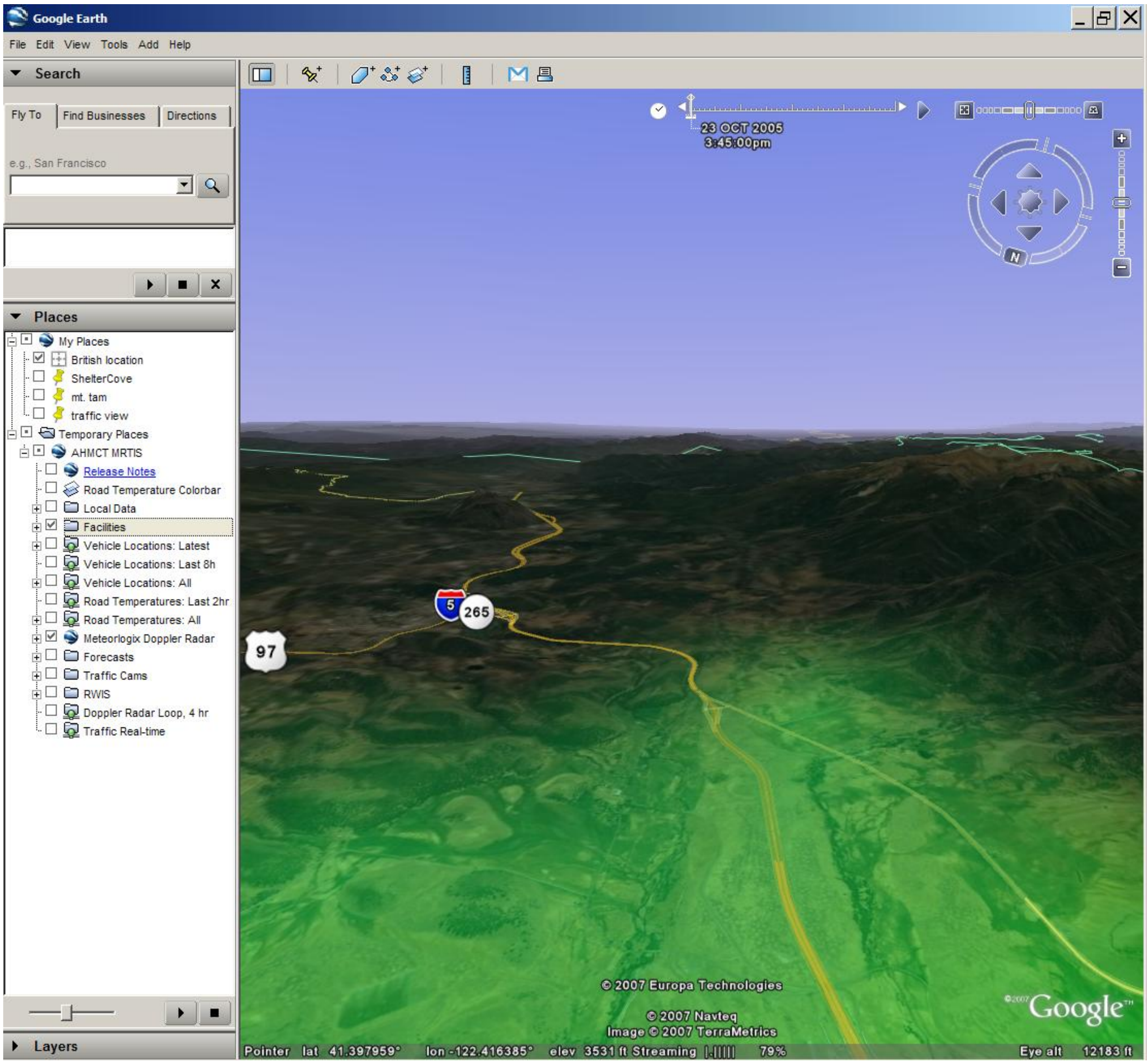

Figure 4.5: MRTIS System Screen: Doppler radar from existing Caltrans VAMS

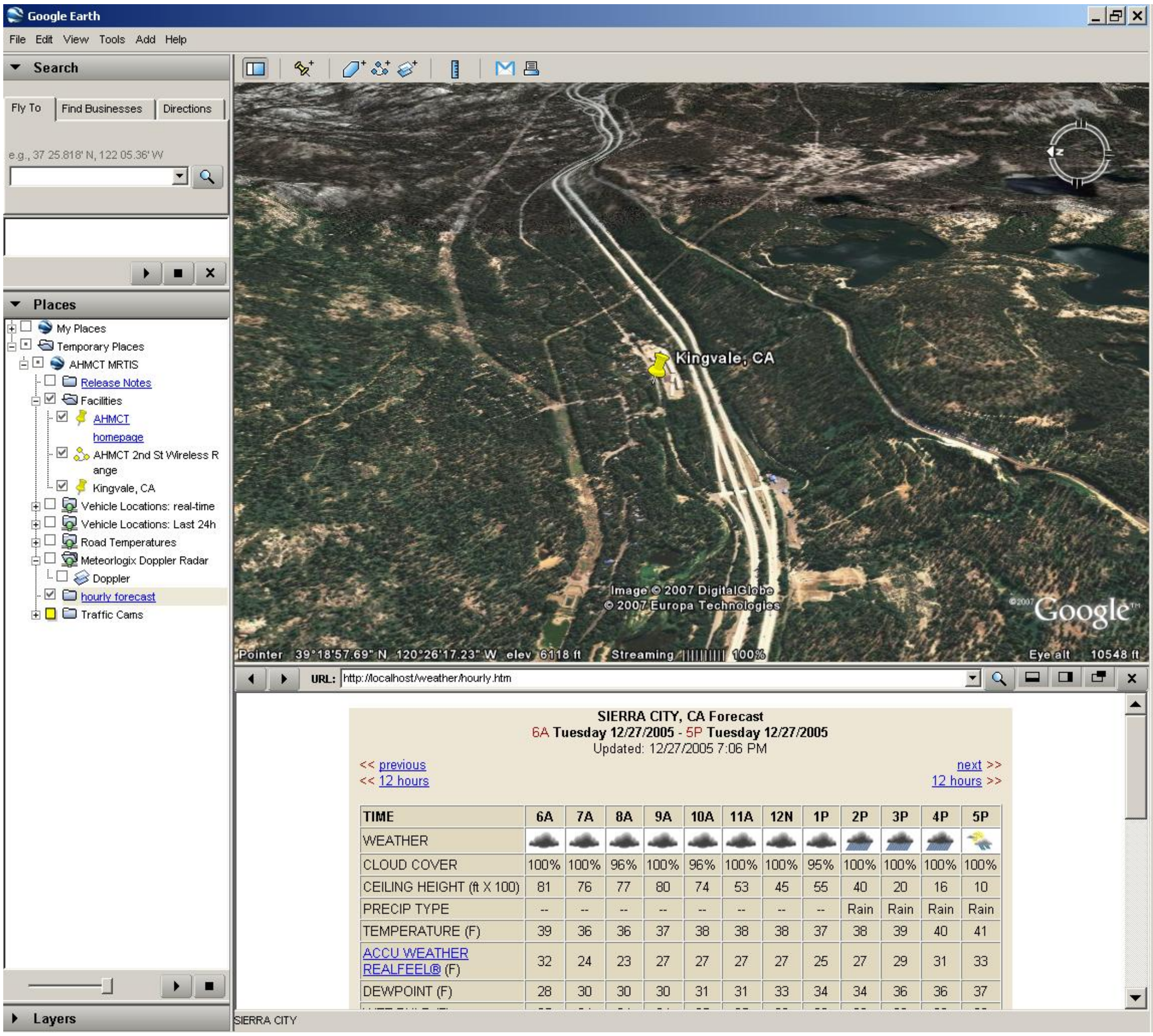

Figure 4.6: MRTIS System Screen: weather information as HTML

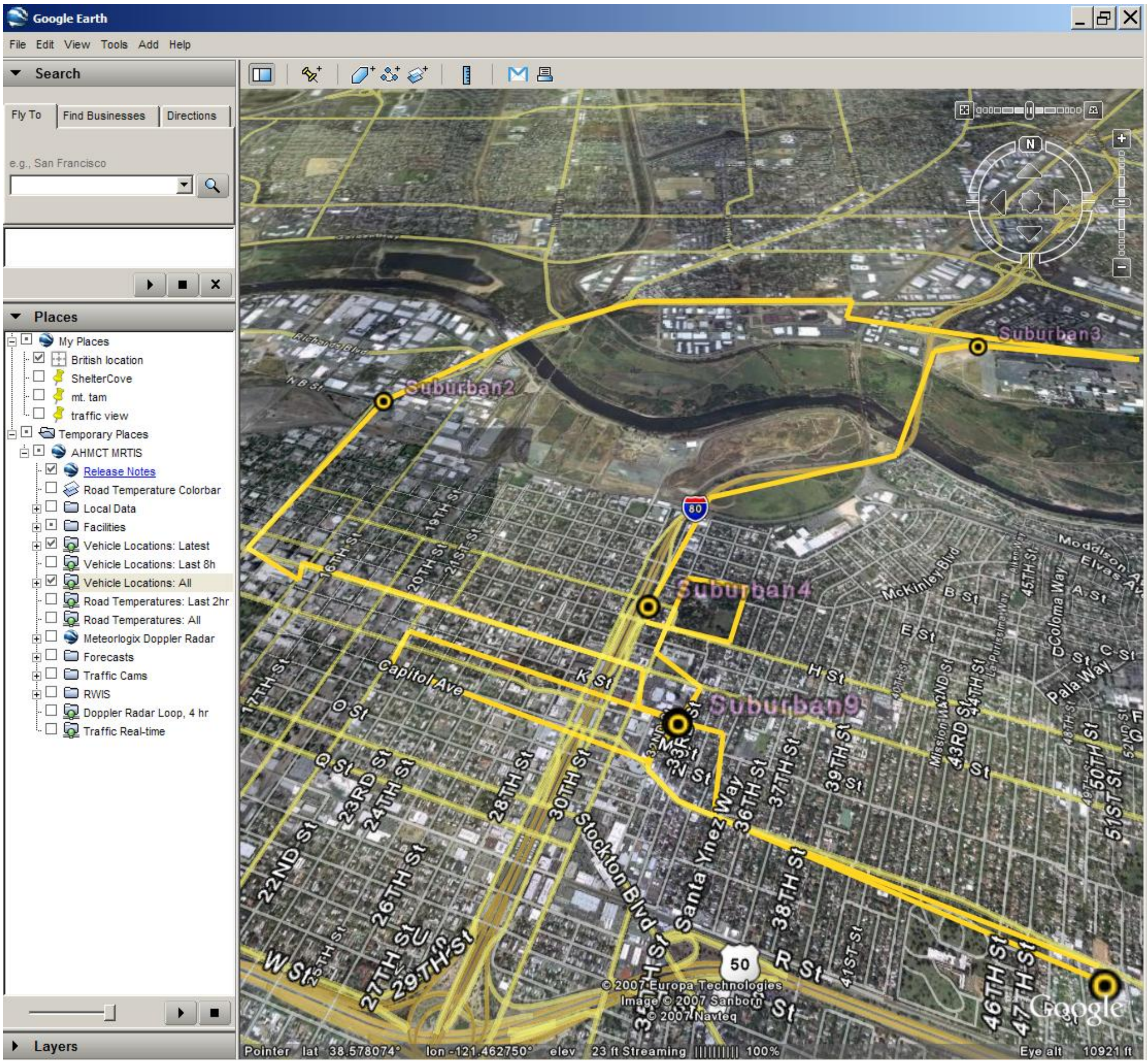

Figure 4.7: MRTIS System Screen: vehicle paths, real-time and historic

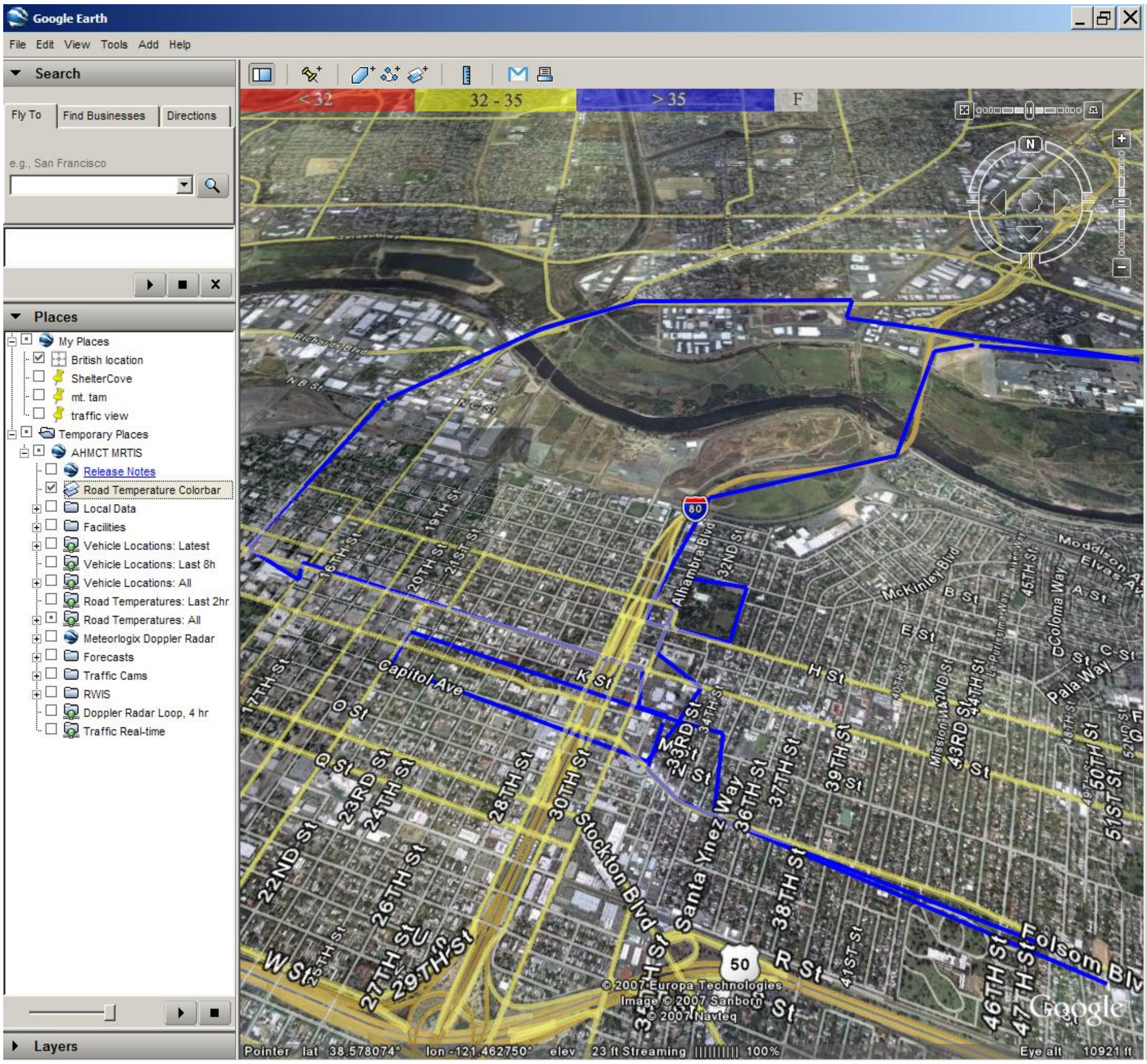

Figure 4.8: MRTIS System Screen: pavement temperatures, real-time and historic

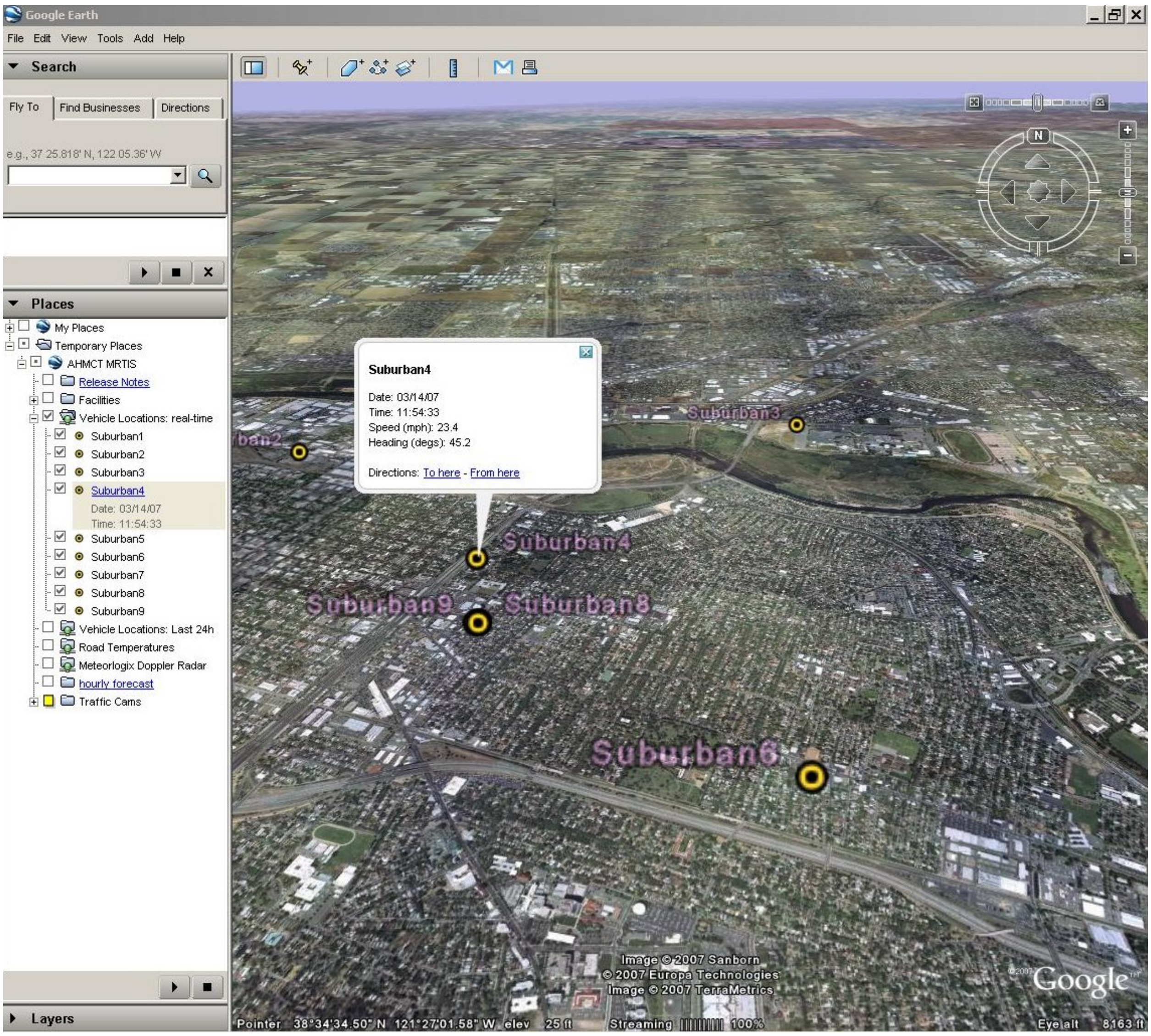

Figure 4.9: MRTIS System Screen: vehicle positions, real-time

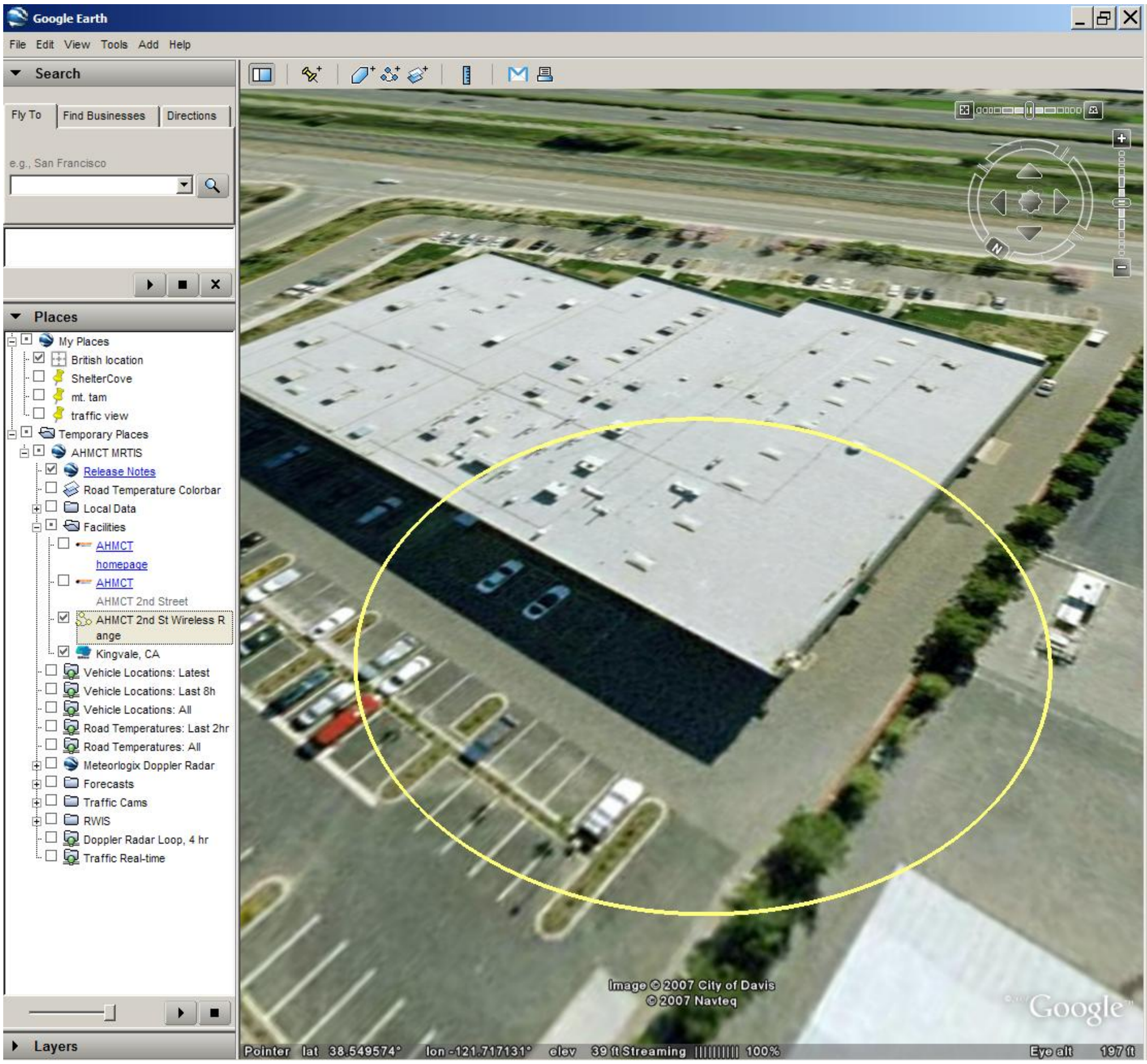

Figure 4.10: MRTIS System Screen: static Wi-Fi wireless station hot-spot location and range

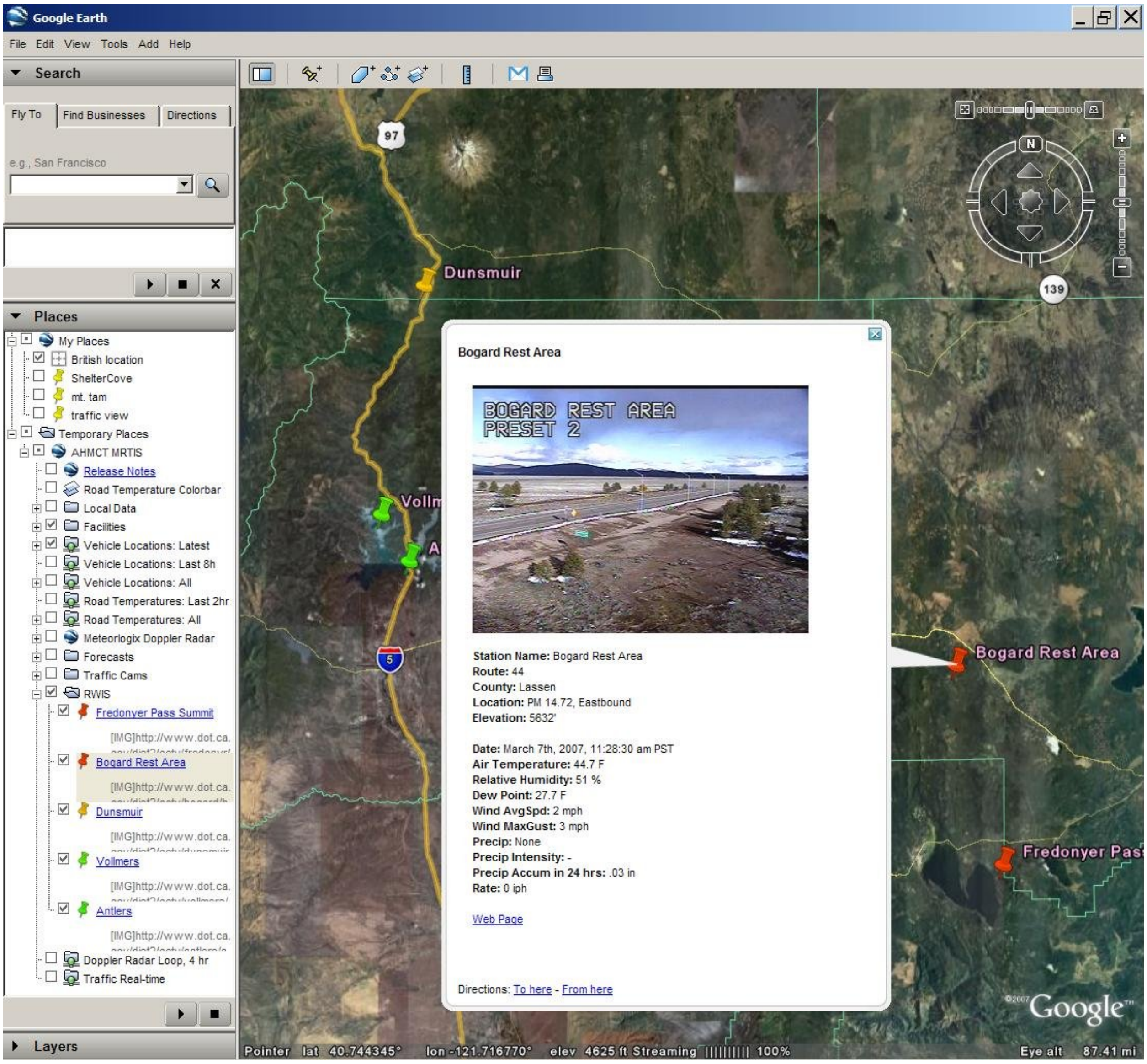

Figure 4.11: MRTIS System Screen: sample RWIS station locations and data

Chapter 5

Laboratory and Field Testing

Testing consisted of three phases: laboratory testing, field testing, and the final project demonstration.

5.1 Laboratory Testing

5.1.1 Unit Testing

Laboratory unit testing was performed during development and following development completion. During development, unit testing was performed both *ad hoc* and using test cases specific to programming classes. In general, the design by contract approach was used during development. Precondition and post-condition statements are used extensively throughout the code. Invariant statements are also used. Contract and invariant failures are logged during program execution. This facilitates detection of anomalous conditions in the code base much closer to the source of the problem, rather than as a side effect somewhere else in the code. A precondition pseudo-code fragment is shown below verifying the argument to the method SignalFilter is larger than zero.

```
double SignalFilter(int order)
{
   // preconds
   Contract.Verify("SignalFilter()",order>0);
   etc.
}
```

5.1.2 Integration Testing

Laboratory integration testing was performed during development and following development completion. To facilitate integration testing a simulator was developed to mimic sensor output within the application responsible for reading data from sensors (SensorReaderApp, see Section 4.2.2). This enabled collection of sensor data in the field (GPS, temperature data) with subsequent playback and debugging in the laboratory using the field data. This capability is particularly important for GPS data which can only be generated in the field.

5.2 Field Testing

Integration testing in the field was performed. This included wireless connections between the test vehicle computer and application servers within the CO. Scenarios defined below were tested (see Section 5.3.3 on the facing page). For example, the test vehicle was used to collect sensor data and return it to the EY.

5.3 Project Demonstration

The MRTIS project has been demonstrated to a subset of the TAG. The goal was to demonstrate the breadth of MRTIS project features and capabilities, including discussion of relevant future research. Logical components available for additional demonstration through the 2007 Calendar Year are: the SV, EY, and CO.

5.3.1 Demonstration Components

- SV: consists of a Cisco 3230 mobile router, Wi-Fi wireless card, cell modem, GPS, RoadWatch road and air temperature sensor, and computer with LCD, keyboard, and pointing device. The computer will have Google Earth©and MRTIS applications installed. The instrumented AHMCT Suburban will be used.

- EY: consists of a Wi-Fi base station installed at the 2nd Street bay door.

- CO: consists of the CO servers (Curly and Fish installed within Academic Surge at the AHMCT Research Center, University of California Davis (UCD), and CO clients. A desktop or laptop computer installed inside the 2nd street bay will be used for this component. Google Earth© will be installed on the client.

5.3.2 Demonstration Configuration

- Both the CO and SV computers will contain previously collected data for 3-6 vehicles. This includes air and road temperature data, and vehicle GPS paths.

- CO client computer: will have Google Earth© installed with the MRTIS KML menu.

5.3.3 Demonstration Scenarios

- SV idling in EY within Wi-Fi wireless range: supervisor can view local and remote KML menu items, e.g. Real-time traffic, Doppler images, etc. The system automatically downloads the latest data (e.g. Doppler radar). This is the same scenario as being out of Wi-Fi wireless range and connected with the cell-modem. The vehicle's position is visible on the screen of the SV and CO client computers.
- SV drives away from the Yard, out of Wi-Fi wireless range, with cell modem disconnected. The supervisor vehicle drives around the block, and back to the Yard. The entire time, the SV position is visible on the computer screen within the SV. When the SV is within wireless range, it automatically uploads vehicle path and temperature data to the CO server. The CO client can then view the updated historic vehicle path.
- SV out of Wi-Fi range, but connected with a cell modem: SV real-time location constantly visible from the CO client computer.

5.3.4 Demonstration and Feedback Survey Results

AHMCT has obtained feedback from a subset of the TAG, and has updated MRTIS based on this feedback. Pending further demonstration in Calendar Year 2007, any additional feedback will be addressed under Deployment Support for a reasonable period. Revisions beyond that will be reserved for a future research project.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The MRTIS project researched, designed, and built a cost-effective mobile real-time information system for snow fighter supervisors. MRTIS provides supervisors with around-the-clock access to essential information while they are in the field. Access to essential information facilitates timely and optimal decisions and simultaneously enables supervisors to be physically available to assist and communicate with snow fighters. System goals were to improve snow removal responsiveness, therefore lowering winter operational costs, and lowering road closure times. MRTIS provides an MDSS framework for integration of additional functionality in the future. The system fits within the existing conceptual ITS framework. It also provides a platform for the integration of further MDSS and RWIS functionality such as regional (meso-scale) weather products, decision support algorithms, ESS information, traffic camera integration, etc.

The System consists of three subsystems: central office servers, the wireless communication link between the central office and vehicles, and the in-vehicle systems. The in-vehicle computing system was designed and implemented for the snowplow supervisor's vehicle and secondary vehicles such as snowplows. Both vehicle types carry on-board sensors such as GPS, and air temperature and infrared pavement temperature sensors. They also carry communications hardware, and MRTIS system software. The MRTIS visualization client is used within the supervisor's vehicle (SV) and also from the winter maintenance central office (CO). The visualization client provides a single integrated three-dimensional user interface, with a time-scale, using a KML browser (Google Earth©). Information integration and sensor fusion is a primary system capability. A non-proprietary messaging platform is used with RDF-formatted messages. System information includes real-time and historic winter maintenance vehicle locations, real-time and historic road and air temperatures, real-time and historic VAMS information such as Doppler radar, real-time traffic information from VDSs, and static facility information such as wireless hot-spot locations. System screen shots are shown starting on pg. 42.

6.2 Future Work

Future work is anticipated to involve fusion of additional sensor data, and integration of other information sources and decision support. Integration of other MDSS functionality such as decision support may be important. Evolving W3C standards related to information sharing (semantic web, see pg. 65) may provide applications with powerful capabilities, such as non-proprietary rule-based intelligence, and additional digital world modeling capabilities. Continued integration of additional information and decision support into the unified KML browser interface (Google Earth©) is anticipated.

More specifically, AHMCT researchers suggest enhancing the existing system using innovations from four sources: recent developments from the FHWA/NOAA Partnership, the MDSS program, the Clarus initiative, and from innovations generated from within AHMCT. MDSS is primarily concerned with weather prediction, road condition prediction, and enhancing anti-icing and de-icing practices. Results and recommendations from an ongoing evaluation of the cost-effectiveness of MDSS would be incorporated into the enhanced winter maintenance information system. Suggested enhancements are consistent with larger trends within the winter maintenance research area, including integration of data into a single interface with the time dimension (platform provided in existing system), analysis of combined non-proprietary data streams, regional (meso-scale) weather prediction, integration with other ITS data sources such as TMC traffic-camera images, peer-to-peer weather observations, and route-specific MDSS anti-icing and de-icing recommendations.

Enhancements to existing system capabilities may include integrating real-time and historic RWIS and Environmental Sensor Station (ESS) data, sensor data sharing (e.g. road temperatures) between vehicles without an intermediary (Central Office), expanded communications options such as 802.11p and/or WiMAX, and integration of additional weather products. Additional enhancements may include regional (meso-scale) weather models, road temperature prediction models, and, in general, integration of other MDSS features that are relevant to Caltrans Winter Maintenance. Benefits are anticipated from integration with sensors to track applied anti-icing and de-icing chemical quantities, integration with other ITS data sources such as Transportation Management Centers and Changeable Message Signs (CMS). Integration with existing Caltrans systems such as scheduling may also be highly beneficial. Travel time predictions may also benefit winter maintenance operations.

Broader systematic enhancements may provide significant benefits. For example, RWIS maintenance can be difficult and tedious. Because of this difficulty, in some cases RWIS stations may be partially functional, or may generate marginal data. Future work could evaluate RWIS design specifically for maintainability, or could develop improved diagnostic and maintenance capabilities for RWIS. In addition, it may be possible to apply optimization and related methods to improve the location of RWIS stations, with the goal of enhanced information report as well as enhanced maintainability.

Additional work could evaluate the most effective communications options in the rural and semi-rural environments for which MRTIS was developed. Wireless 900 MHz radios have been identified as a potential communications medium. For example, in some environments, 900 MHz may outperform other options, such as GPRS, and should be considered as a viable option. A detailed evaluation of communications options in these areas would be of broad benefit to Caltrans.

References

[1] E. Anderson and J. Nyman. Southeast Michigan Snow and Ice Management (SEMSIM): Final evaluation at end of winter season year 2000. Technical report, Road Commission of Oakland County, 2000.

[2] Lennart Axelson. Development and use of the Swedish Road Weather Information System. Swedish National Road Administration. URL http://www.rwis.net/.

[3] L. Ballard. Winter Maintenance: Policy, Management, and Performance, Analysis of Road Weather Information System Users in California and Montana. Technical report, Sixth International Symposium on Snow Removal and Ice Control Technology, Sacramento, California, 2004.

[4] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284 (55):28–37, 2001.

[5] R. R. Blackburn, E. J. McGrane, C. C. Chappelow, D. W. Harwood, and E. J. Fleege. Development of anti-icing technology. Technical Report SHRP-H-385, National Research Council, Strategic Highway Research Program, Washington, DC, 1994.

[6] S. E. Boselly and D. Ernst. Road Weather Information Systems, volume 2 implementation guide. Technical Report SHRP-H-351, National Research Council, Strategic Highway Research Program, 1993.

[7] S. E. Boselly, J. E. Thornes, C. Ulberg, and D. Ernst. Road Weather Information Systems, volume 1: Research report. Technical Report SHRP-H-351, National Research Council, Strategic Highway Research Program, 1993.

[8] B. D. Breen. Anti-icing success fuels expansion of the program in Idaho. Technical report, Idaho Transportation Department, 2001.

[9] Caltrans Division of Research and Innovation, Version 1.1. Systems Engineering Guidebook For ITS. Technical report, Caltrans, February 2005. URL http://www.dot.ca.gov/newtech/docs/se_guidebook_ver1-12_14_05.pdf.

[10] M. T. Darter, T. A. Lasky, and B. Ravani. Transportation asset management and visualization using semantic models and Google Earth. *Transportation Research Record: Journal of the Transportation Research Board*, (TRR 2024), 2007.

[11] M.T. Darter, K.S. Yen, B. Ravani, and T.A. Lasky. Literature review of national developments in ATMS and open source software. Technical Report UCD-ARR-06-12-08-01, December 2006.

[12] P. Davies, Dean Deeter, and C. E. Bland. FORETELL™: Providing integrated weather information services across the upper midwest. In *1998 Transportation Conference Proceedings*, 1998.

[13] FHWA. Maintenance Decision Support System. Web Page, Dec 2006. URL `http://www.fhwa.dot.gov/crt/lifecycle/mdss.cfm`.

[14] FHWA Road Weather Management Program. Surface transportation weather decision support requirements operational concept description. senior project, FHWA Road Weather Management Program, July 14 2000.

[15] FHWA Weather Team. Weather information for surface transportation: A white paper on needs, issues and action. senior project, FHWA, May 15 1998.

[16] IEEE. IEEE recommended practice for software requirements specification. Technical Report ISBN: 0-7381-0332-2, IEEE Computer Society, Oct 1998.

[17] G. G. Nelson and P. Pisano. Surface Transportation Weather Decision Support Requirements, Draft Version 2.0: Preliminary Interface Requirements for Advanced-Integrated Decision Support Using Weather Information for Surface Transportation Decisions Makers. Technical report, Federal Highway Administration, Office of Transportation Operations, Road Weather Management Program, Washington, DC, 2000.

[18] NTCIP. National Transportation Communications for ITS Protocol - object definitions for Environmental Sensor Stations (ESS). Technical Report 1204:1998 v01.13, NTCIP, 1998.

[19] NTCIP. National Transportation Communications for ITS Protocol (NTCIP), the NTCIP Guide, Version V03.02b. Technical Report 9001, NTCIP, 2002.

[20] J. F. Paniati. Integrating surface transportation weather information systems - the DOT role. In *Transportation Research Board 85th Annual Meeting*. Transportation Research Board, Jan 2006. Session 494: SAFETEA-LU Road Weather Research and Development Program.

[21] T. M. Smith and V. Lakshmanan. Utilizing Google Earth as a GIS platform for weather applications. In *22nd International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*, Feb 2005.

[22] Strategic Highway Research Program. Road Weather Information Systems. Research Report SHRP-H-350, National Research Council, Washington DC, 1993.

[23] J. M. Sussman. *Perspectives on Intelligent Transportation Systems*. Springer, 2006.

[24] Transportation Pooled Fund Program. Development of Maintenance Decision Support System. Web Page, Dec 2006. URL http://www.pooledfund.org/projectdetails.asp?id=240&status=6.

[25] University Corporation for Atmospheric Research. Maintenance Decision Support System. Web Page, Dec 2006. URL http://www.rap.ucar.edu/projects/rdwx_mdss/.

[26] U.S. Department of Transportation. The clarus initiative. Web Page, Dec 2006. URL http://www.clarusinitiative.org/.

[27] U.S. Department of Transportation. The Vehicle Infrastructure Integration initiative. Web Page, Dec 2006. URL http://www.its.dot.gov/vii/.

[28] C. E. Wallace. ITS Florida–yesterday, today, and tomorrow. 2005. URL http://itsflorida.org/documents/official_doc/Founding_of_ITSA_FL.doc.

[29] C. E. Wallace. A series of e-mail messages about the founding of what we now call ITS. 2005. URL http://itsflorida.org/documents/official_doc/April-Email_Histories.doc.

[30] Western Transportation Institute. Analysis of Maintenance Decision Support System benefits and costs. Web Page, Dec 2006. URL http://www.coe.montana.edu/wti/.

Appendix A

Semantic Web Standards

This appendix briefly introduces semantic modeling–an important technology that is particularly relevant for applications that need to share structured information. ITS and ATMS applications are fundamentally concerned with sharing information between computer applications that may be located across cities, counties, and districts. Sharing information requires standards. The W3C[1] has developed a group of standards, for defining structured information; standards commonly referred to as the semantic web [4]. Figure A.1 shows how these standards are organized into a so-called semantic stack.

The term semantic web can be misleading because the associated standards are fundamentally about syntax, and their use transcends the World Wide Web. The bottom three layers of Figure A.1 identify standards for character formatting (Unicode), resource naming and location across a network (Uniform Resource Identifier (URI)), and document structure (XML, Namespaces, and XML Schema). For our purposes, the fourth and fifth layers are the most interesting–they define modeling standards used to structure data, and are discussed below. These include RDF, RDF Schema, and OWL, among others. The sixth layer defines a powerful query language known as SPARQL Protocol and RDF Query Language (SPARQL). Higher levels of the semantic stack define functionality that may be useful for rule-based systems, e.g. for ATMS.

The fourth and fifth layers of the semantic stack define standards for modeling and structuring data. These include RDF, RDF Schema, and OWL, among others. Constructed models are essentially object-oriented and enable the definition of complex classes, properties, and objects. It is important to note that these standards enable much more sophisticated modeling than is possible with traditional object-oriented programming languages (e.g. Java, C++, C#, etc.) or object relational databases. The standards support sophisticated modeling with cardinality constraints, property and class hierarchies, data constraints, and, most importantly, the ability to define and use models distributed over the network. These capabilities are particularly relevant for ITS and ATMS applications, which depend on shared data between agencies and many different types of sensors and

[1]For World Wide Web Consortium (W3C) semantic web standards see http://www.w3.org/2001/sw/

hardware.

For clarity, a practical example of semantic modeling may be helpful. Transportation assets such as VDS, Changeable Message Sign (CMS), culverts, etc., may be semantically modeled [10]. These assets share many properties such as location, highway number, county, installation date, inspection date, etc. These properties, associated classes, and objects have in the past been typically defined using a database or a programming language such as Java, PHP, or C++. A better alternative is to define these properties, classes, and objects using a standardized W3C format such as OWL, which is based on XML. For example, the code below defines a single property named countyID which is a W3C-defined string (via the URL `http://www.w3.org/2001/XMLSchema#string`), and is used by two classes: Culvert and VDS. The code below was created using the Protégé open-source ontology editor. Open-source software uses an open-source license such as the GNU General Public License (GPL) [11].

```
<owl:DatatypeProperty rdf:about="\#countyID">
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
   <rdfs:domain>
      <owl:Class>
         <owl:unionOf rdf:parseType="Collection">
            <owl:Class rdf:about="\#VDS"/>
            <owl:Class rdf:about="\#Culvert"/>
         </owl:unionOf>
      </owl:Class>
   </rdfs:domain>
</owl:DatatypeProperty>
```

The countyID property defined above is in a non-proprietary format (OWL), is independent of any particular database or language, and would therefore be easy to share with other districts, cities, and states. The countyID property can be used by instances of OWL-defined classes. The following code defines a portion of this class hierarchy:

```
<owl:Class rdf:ID="AssetThing"/>
<owl:Class rdf:ID="StationaryThing">
   <rdfs:subClassOf rdf:resource="\#AssetThing"/>
</owl:Class>
<owl:Class rdf:ID="VDS">
   <rdfs:subClassOf rdf:resource="\# StationaryThing"/>
</owl:Class>
```

The OWL code below (also developed in Protégé) shows an instance of a VDS containing an instance of the countyID property:

```
<VDS rdf:ID="vds-313618">
   <countyID rdf:datatype="http://www.w3.org/2001/XMLSchema#string">8</countyID>
   <cal_pm rdf:datatype="http://www.w3.org/2001/XMLSchema#float">80.76</cal_pm>
   <locatedIn rdf:resource="#Caltrans_8"/>
</Culvert>
```

In summary, semantic modeling of a problem domain (e.g. ATMS) uses W3C standards to define classes, properties, and their instances. Classes and properties are typically defined by domain experts and software engineers using ontology editors (e.g. Protégé). These definitions are stored in OWL or RDF Schema files, and define the content and structure of application data. They can be shared with other agencies interested in building interoperable systems. Software developers use the semantic definitions to build applications. Semantic toolkits (e.g. Jena) can be used to read semantic definitions directly, enable persistent database storage, and integrate with procedural languages like Java. Development of rule-based semantic applications is also supported. One of the key benefits of this approach is that the essence of the system–class and property definitions–is defined in a non-proprietary standardized format that is independent of any particular operating system, database product, or software vendor. This promotes interoperability, reduces lock-in, and promotes sharing of interfaces while allowing agencies to keep differentiating or sensitive Intellectual Property private.

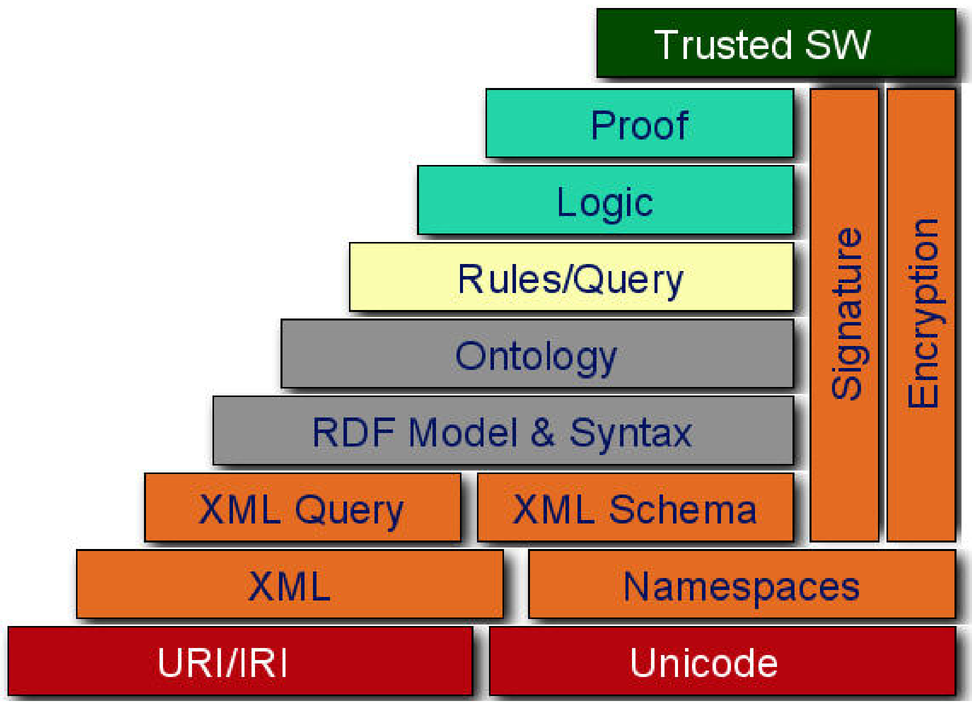

Figure A.1: W3C Semantic Web Stack (http://www.w3.org)