Technical Memorandum

SunGuide® Software and Lonestar Software

Comparison and Analysis

Version 1.1

May 31, 2013

Prepared for:

Florida Department of Transportation
Intelligent Transportation Systems Program
605 Suwannee Street, M.S. 90
Tallahassee, Florida 32399-0450
(850) 410-5600

AND

Texas Department of Transportation
TxDOT Development, Integration, Implementation, and Maintenance Services Program
9500 N. Lake Creek Parkway, Bldg 51
Austin, Texas 78717
(512) 506-5000

Table of Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

ATMS ..... Advanced Traffic Management System
ATP ..... Acceptance Test Plan
AVL ..... Automatic Vehicle Location
AWARD ..... Advance Warning to Avoid Railroad Delays
C2C ..... Center-to-Center
CCTV ..... Closed-Circuit Television
CDW ..... Central Data Warehouse
CMB ..... Change Management Board
ConOps ..... Concept of Operations
CR ..... Change Request
CSD ..... Command and Status Distribution
DBDD ..... Database Design Document
DBMS ..... Database Management System
DIIMS ..... Development, Integration, Implementation, and Maintenance Services
DMS ..... Dynamic Message Sign
DOT ..... Department of Transportation
DPA ..... Data Processing Application
ESS ..... Environmental Sensor Station
FAT ..... Factory Acceptance Test
FDOT ..... Florida Department of Transportation
FHP ..... Florida Highway Patrol
FTE ..... Florida's Turnpike Enterprise
GUI ..... Graphical User Interface
HAR ..... Highway Advisory Radio
HTML ..... HyperText Markup Language
ICD ..... Interface Control Document
IP ..... Internet Protocol
ITS ..... Intelligent Transportation Systems
IV&V ..... Independent Verification and Validation
LCS ..... Lane Control Subsystem
MAS ..... Message Arbitration Subsystem
MCP ..... Manual Control Panel
MSA ..... Message Scheduling Application
MQA ..... Message Queuing Application

PCM ..... Product Configuration Management
PTZ ..... Pan/Tilt/Zoom
RWIS ..... Road Weather Information System
SDD ..... Software Design Document
SRS ..... System Requirements Specification
TCP ..... Transmission Control Protocol
TE ..... Traffic Engineering
TMC ..... Transportation Management Center
TRF ..... Traffic Operations Division
TSS ..... Transportation Sensor Subsystem
TxDOT ..... Texas Department of Transportation
UDP ..... User Datagram Protocol
UI ..... User Interface
VSL ..... Variable Speed Limit
XML ..... eXtensible Markup Language

1 Scope

1.1 Document Identification

This document records the comparison and analysis of both the Florida Department of Transportation (FDOT) and Texas Department of Transportation (TxDOT) advanced traffic management systems (ATMS) software, SunGuide® Software and Lonestar Software. It also provides recommendations for the next steps in the process towards creating a common software platform from which both departments of transportation (DOT) will enhance, support, and deploy collaboratively and synergistically. Consolidating the software systems to a common platform will provide significant cost savings overtime for both DOTs by eliminating redundant software development and support efforts.

1.2 Background

ATMS software is needed in transportation management centers (TMC) to facilitate traffic operations, including managing intelligent transportation systems (ITS) devices, detecting and managing traffic events, and collecting and reporting related data.

FDOT and TxDOT have both built an ATMS software suite that is similar in purpose, design, and implementation; however, is not adequately compatible to directly share system components. Over the years, both agencies have made enhancements to their software systems. Some enhancements have been funded and built by one agency and then shared with the other agency. The effort involved in taking an enhancement that has been completed and deployed and making it available to another agency can be substantial. This is due to the ATMS software products maturing in slightly different directions; however, FDOT and TxDOT are both interested in maintaining a collaborative relationship to build on their successes and their assets, and achieve the capability to share system components and enhancements, and thus share the burden of development and maintenance of the software.

To continue and enhance this collaboration effort and to reduce the cost of sharing enhancements from one agency to the other, FDOT and TxDOT are investigating the possibility of consolidating the two software products into a single platform. Both agencies have tasked their consultants to work together to review and compare each other's system in detail to understand the feasibility and develop a high-level approach to fulfil their vision of consolidating the software into a single, common platform. This document captures this comparison effort and recommends an approach for the next steps in the effort. This document and any other deliverables from the investigation will be delivered to both agencies. FDOT and TxDOT will coordinate with the stakeholders of both existing systems for additional awareness, input, and subsequent decision making related to consolidating the software products together.

1.3 Approach to Consolidation

Both agencies will work together as a team on this effort. The team will start by comparing the software products as a whole, along with high-level features and other project parameters that impact how the software is built. Then the team will look into the system-wide details that permeate and affect the majority of the software product. After this initial analysis, the team will conduct a site visit to one or more TMCs running the other agency's software to get a deeper understanding of how the software is used operationally. Finally, they will analyze the details of a specific module in order to quantify a representative estimation of the level of effort to merge the software products.

1.4 References

The documents and information in Table 1.1 were referenced or otherwise relevant in the comparison and analysis reported recorded in this document. Most of these documents can be found by contacting the SunGuide software or Lonestar contacts listed in section 1.5.

Table 1.1 – Information Inventory

Subject	SunGuide Software	Lonestar
Development Scope	*SunGuide Software Support, Maintenance, and Development*, June 25, 2010, and all contract modifications. See the contract documents section on the SunGuide software project web sites at http://sunguidesoftware.com/	
History Description	Sect. 2.1; pg. 5	
Systems Engineering	*Systems Engineering Guidebook for ITS;* http://www.fhwa.dot.gov/cadiv/segb/	
System Operation Description	Sect. 2.2; pg. 5	Sect. 3.1; pg. 17
System Operation	Concept of Operations (ConOps)	ConOps (per subsystem)
Change Management	Sect. 2.4; pg. 9 Scope of services document (referenced documents) Change Management Board presentations and meeting minutes: http://www.dot.state.fl.us/trafficoperations/ITS/Projects_Deploy/CMB.shtm	Sect. 3.4; pg. 18
Plan Documents	Software Development Plan	
Plan Documents	Project Staffing Plan	

Subject	SunGuide Software	Lonestar
Plan Documents	Risk Management Plan	
Plan Documents	Configuration Management Plan	Configuration Management Plan
Plan Documents	Quality Assurance Plan	
Plan Documents	Subcontractor Management Plan	
Plan Documents	Software Security Plan	
Requirements	Software Requirements Specification (document)	Software Requirements Specifications (per subsystem)
Requirements	RequisitePro requirements database	
System Architecture	Sect.2.5; pg. 11, and Software Design Document (SDD)	Sect. 3.5; pg. 19 SDD (per subsystem)
Database	Database Design Document (DBDD)	DBDD
Interface Control Document (ICD)	ICD (per subsystem)	ICD (per subsystem)
Release	Version Description Document (entire system – includes release notes as document section), Installation Notes	Version Description Document (per subsystem), Release Notes, Upgrade Steps
Usage	Software Users Manual	*Lonestar Software Users Manual* version 4.0.0, TxDOT, May 26, 2012, Administrator's Reference Guide
Usage	Training Plan, Administrator Training (PowerPoint presentation slideshow), Users Training (PowerPoint presentation slideshow)	Lonestar Administrator Training Slides (PowerPoint presentation slideshow), Lonestar Operator Training Slides (PowerPoint presentation slideshow)
Features (Modules/ Subsystems)	Sect. 2.6; pg. 15	Sect. 4.4; pg. 28
Protocols Supported	Sect.4.5; pg. 30	Sect.4.5; pg. 30
Testing	Software Integration Procedure (Factory Acceptance Test Plan)	Lonestar Acceptance Test Plan (ATP)
Testing	Software Integration Case Procedures (Factory Acceptance Test Procedures)	Lonestar ATP
Deployment	Implementation Plan (per deployment)	Upgrade Plan (per deployment)

1.5 Contacts

Charlie Farnham
Texas Department of Transportation
DIIMS Contract Manager
9500 N. Lake Creek Parkway, Bldg 51
Austin, Texas 78717
(512) 506-5115
Charlie.Farnham@txdot.gov

Cynthia Clark
AECOM
DIIMS Software Leader
5757 Woodway Drive, Suite 101 West
Houston, Texas 77057
(281) 389-1945
cynthia.clark@aecom.com

Arun Krishnamurthy, P.E.
ITS Software and Architecture Coordinator
Florida Department of Transportation
ITS Office
605 Suwannee Street, MS90
Tallahassee, FL 32399-0450
(850) 410-5615
Arun.krishnamurthy@dot.state.fl.us

Clay P. Packard, P.E.
SunGuide Project Manager
Florida Department of Transportation
ITS Office
605 Suwannee Street, MS90
Tallahassee, FL 32399-0450
(850) 410-5623
clay.packard@dot.state.fl.us

2 SunGuide Software Overview

2.1 History

In 2001, FDOT conducted a study to determine the best method to acquire an ATMS software solution that would meet their specific needs for the entire state. One of the potential objectives was to reduce cost by eliminating duplication that would be inevitable if each FDOT District developed their own ATMS software solutions. FDOT determined that the best course of action would be to develop a tailored ATMS software for Florida. FDOT started with the Texas software as a baseline and modified and enhanced it into the system that is now known as the SunGuide software.

2.2 Stakeholders

2.2.1 Stakeholder Categories

Stakeholders are considered as anyone involved in the SunGuide software project at all levels. They are categorized as:

1. **Motorists and the general public** who benefit from (and pay taxes for) a roadway transportation system that provides safety and mobility.

2. **FDOT Central Office** coordinates and manages software that meets the needs of the entire state. They also conduct software testing and analysis of data collected by the software. They provide many support functions and conduct program management for the overall SunGuide software program.

3. **FDOT Districts** are geographical divisions within FDOT that have responsibility of managing limited-access roadways and some arterial roadways within the statewide network. They deploy and make use of the software in a production environment. These stakeholders include Districts 1 through 7, and Florida's Turnpike Enterprise (FTE) as District 8.

4. **FDOT's Traffic Engineering Research Laboratory** provides an environment where the software and ITS devices are tested independently prior to release for production use. Several roles exist within the Districts to fulfil the traffic management operations, including:

    a. **ITS engineers** manage the software deployment within their District. They represent their agency at the change management board (CMB).

    b. **TMC managers** oversee the use of the software. They also use the reporting, auditing, and other management functions of the SunGuide software.

c. **Technical administrators** support deployment and use of the software. They may setup servers and workstations, and use the configuration functions of the software to prepare it for use by TMC operators.

d. **TMC operators** use the software directly. They manage incidents and perform the day-to-day traffic operations tasks.

5. **Expressway authorities** have legislative authority to operate limited-access expressways within their jurisdictions. These stakeholders include Miami-Dade Expressway Authority and Orlando-Orange County Expressway Authority.

6. **Local agencies** have more localized jurisdiction and may enter into an agreement with FDOT to use the software for their arterial and local facilities. These stakeholders include the City of Tallahassee and Lee County.

7. **ATMS software contractor** provides development, support, and maintenance services. They are the current licensing agent between Texas and Florida for sharing the software source code between the states.

8. **ITS consultants** perform a variety of tasks including management, specification, development, testing, training, demonstrating, analysis, technical support, etc.

9. **ITS product vendors** sell detectors, cameras, DMS signs, and other products that integrate with the SunGuide software.

10. **Executive management** uses performance measures reports generated by the software to demonstrate the benefits of funding the software and other related programs to the legislature.

ITS engineers work with their TMC managers, consultants, technical administrators, and operators to determine the needs of the agency or division. They represent those needs by participating in the statewide ITS change management process. There are related projects that also represent other needs for the software, such as the 511 advanced traveller information system project manager and vendors who sell products that integrate with SunGuide software.

Users are categorized by agency, TMC, and finally by the staff and their role within the group. The TMC is a general way to refer to a user, as there are a diverse number of roles and needs within the agency's TMC operations that are considered together and executed by personnel within that agency.

2.2.2 Current Users

Florida is geographically divided into seven Districts and the FTE, which is considered as the eighth District. The Districts manage all of Florida's interstate facilities using the SunGuide

software. Some Districts have a satellite TMC with the ability to take over SunGuide software operations if a disaster occurs at the regional TMC. Additionally, two of Florida's largest expressway authorities also use the SunGuide software.

Each TMC has different requirements for SunGuide software usage and this warrants discussion and coordination at the change management board to ensure compatibility with each other and provide a seamless network of ITS functions along Florida's major transportation corridors. There are expressway authorities and local agencies in Florida that also use SunGuide software and participate; however, there are no SunGuide software users outside of the state of Florida. SunGuide software currently has 12 users including District regional TMCs, satellite TMCs, expressway authority TMCs, and local agency TMCs.

2.3 Operations

SunGuide software fulfils the role of an ATMS software and is used for a wide range of operations related to traffic management. ITS device control, traffic and incident management, and reporting and data analysis are the three main operational categories. A description of the operational categories supported by SunGuide software is provided in the following subsections.

2.3.1 ITS Device Control

SunGuide software is the central software that integrates with ITS field devices. SunGuide software communicates with these ITS devices through a transmission control protocol (TCP)/internet protocol (IP) connection or through user datagram protocol (UDP) over a roadside network. ITS devices receive commands and requests from the SunGuide software and report information such as health, status, and any operational data or responses back to the software. SunGuide software provides real-time data received from ITS devices, available on a common databus, to all SunGuide software processes. SunGuide software also has a database to configure and store pertinent information regarding device identification, location with global positioning system coordinates, and communications parameters and other details of how the software should behave. SunGuide software uses the database to archive data received from ITS devices for later use in reporting and data analysis.

FDOT uses several types of ITS devices, including traffic detection devices, which primarily provide data input to SunGuide software; and DMS devices, which apart from reporting their status, accept commands from the software to send information out to motorists. Not all detection and dissemination requires physical devices managed by SunGuide software. SunGuide software has a center-to-center (C2C) interface for external systems in lieu of field devices to provide live traffic data as well as providing a channel for the software to disseminate information.

SunGuide software integrates these devices and data interfaces in many ways that are critical to other SunGuide software operations. SunGuide software has several modules that are not device-

specific to process data and support various operations. Travel times and event management are two examples.

2.3.2 Traffic and Incident Management

Traffic and incident management is the primary purpose of the SunGuide software and the ITS devices and interfaces. TMC operators use the software to monitor devices and traffic conditions and respond to any alerts presented by the SunGuide software. If an alert reveals a real traffic incident, the TMC operator uses the software to verify, track, and record the incident, and coordinate a response to the event with other agencies.

Many SunGuide software processes utilize data available in the system to enable, automate, and track tasks that would otherwise be performed manually by a TMC operator. SunGuide software automatically updates travel times. Response plans with messages warning motorists of blocked lanes ahead are automatically suggested and sent out to DMSs, highway advisory radios (HAR), and the 511 advanced traveller information system after operator approval. Another process calculates congestion using transportation sensor subsystem (TSS) data available in SunGuide software and posts variable speed limits (VSL) to signs on the roadway; while another use of the TSS data is for automatic calculation of updated pricing for managed lanes in South Florida.

2.3.3 Reporting and Data Analysis

SunGuide software supports various types of reporting. Performance measures are heavily relied upon to verify the effectiveness and efficiency of traffic operations practices. Each activity within a traffic incident is recorded with the timestamp so that the event timeline can be established. The timeline is used as a template for consistently measuring the duration of the notification from the Florida Highway Patrol (FHP); verification by the TMC; response by FHP, service patrol, or other responder; clearance of the travel lanes; and full clearance of the incident site. These durations and other statistics are aggregated over a time period of interest and broken down by various event characteristic, including time of day, location, and other statistics.

Detection data is also stored and used in reports; it is widely used in data analysis. A separate central data warehouse (CDW) is a conceptual system that stores all of the data from regional TMCs throughout the state. The University of Florida developed a prototype and the University of Maryland has a CDW system in place that will assimilate and host the data from the entire state of Florida. This system will include TSS data and incident data.

In summary, SunGuide software uses ITS device management to facilitate traffic and incident operations in such a way that reporting and data analysis can provide feedback to verify and improve the effectiveness and efficiency on Florida's roadways, supporting FDOT's mission of safety and mobility.

2.4 Development Processes

The ITS architecture and design/build approach calls for using systems engineering principles as specified by the Federal Highway Administration. SunGuide software follows this development process with thorough documentation. This documentation is referenced by section 6.1 on page 53. The following sections highlight a couple of the steps in the systems engineering process that are relevant to the scope of this document.

2.4.1 Configuration Management

The high-level configuration of SunGuide software starts with defining or refining a ConOps. This typically is a detail related to the preceding Operations section describing the highest level concepts of ITS device management, traffic and incident management, and reporting and data analysis. This ConOps is then brought before the stakeholders for review and formal vote in the change management board meeting. Past agendas and meeting minutes are available on the internet at:

http://www.dot.state.fl.us/trafficoperations/ITS/Projects_Deploy/CMB.shtm.

Once approved, the ConOps is then implemented by the Central Office and their contractor.

2.4.2 Requirements

Requirements are developed from the user's needs. There are two phases to the development of the requirements: the system requirements and the software requirements.

The system requirements are short descriptions of individual system features from the ConOps. They are also assigned a SunGuide software ID that follows another four-tier scheme based on the initial contract scope. These system requirements are tested through integration testing procedures during the factory acceptance test (FAT) process prior to the software product delivery to FDOT. After FDOT approval at the FAT, FDOT tests the software against these system requirements through operational scenarios created in a testing environment by the independent verification and validation (IV&V) process prior to releasing the software product to the users.

The software requirements pertain to the implementation details of the system requirements and are short descriptions of individual characteristics of the software itself. They are usually tested through integration testing scenarios in a lab environment by the software development consultant internally.

FDOT is responsible for developing the ConOps and the system level requirements. The ConOps and system level requirements are provided to the development team to produce the software requirements, which are reviewed by FDOT prior to the design process.

2.4.3 Design

Once the detailed requirements are approved, the design begins. Depending on the size and complexity of the release, the design is initially provided as a slideshow presentation, presented in a meeting with the development team and FDOT. FDOT provides comments and the development team updates the design to reflect these comments. This comment and response activity may repeat if a second design review is a part of the scope of the development effort.

A formal set of design documents is also produced as a deliverable for every major release. This is provided as a set of hypertext markup language (HTML) files with embedded diagram images and is very architectural in nature as to how the components communicate.

Another design artifact that is produced and maintained is the ICD that defines how a SunGuide software system module communicates with other modules. ICDs are provided as a document describing the interface as well as a set of extensible markup language (XML) schema files that can be used to understand, parse, and validate any message against the message structure defined by the ICD.

2.4.4 Implementation

The implementation process follows the software documentation that specified how to write a driver, how to write a subsystem, and a coding style guide. This activity is done after the design phase and prior to the FAT phase and is thus the development team's responsibility to define, follow, and control this process.

2.4.5 Testing

Testing occurs in three distinct phases. Unit testing is integrated into the software implementation process. The development team uses a combination of peer code reviews, manual testing, and even some automated unit testing to ensure the quality of the units. As a part of the implementation process, this is the development team's responsibility to define, follow, and control this process.

FAT is an activity conducted by the software development team and witnessed by FDOT, usually at the software development laboratory. FDOT develops a physical configuration audit document that is used to verify that the selected location is ready to perform FAT. A software integration plan and a software integration case procedures document, which outlines the plan and procedures, respectively, for the testing activity, is produced by the developer. Once the software is accepted at FAT, the software is delivered to the IV&V team for independent testing.

The IV&V team produces independent test plan and procedures documents to be used for the independent testing. They conduct this test in FDOT's facility to verify that the product is ready for release.

If any issues are found in either of the testing, the developer corrects the issues and submits the correction for retesting. After the software is independently tested to meet all of the requirements, it is accepted by FDOT and released to the users for deployment and production use.

2.4.6 Release Cycles

Enhancements and bug fixes are organized into releases that depend on the needs of the users, the schedule of those needs, and a reasonable level of work for the contractor to accept. Typically, each year has a major release, one or two moderate releases, and one or two minor releases.

Software releases are categorized as major, moderate, and minor. Major releases involve significant change to the software, follow all steps in the software development process, and an entire set of development process documents are developed or updated. Moderate and minor releases may defer updating some of the process documents to the next major release. Patches and hot fixes are very minor and specific modifications; they do not include a formal installation package and documentation, and include minimal documentation. These patches and hot fixes are intended to meet extremely urgent needs for specific users who cannot wait for an entire release cycle.

2.5 System Architecture

2.5.1 Modular Relationships

SunGuide software follows a distributed, message-oriented architecture, with loosely coupled modules that each focus on a single ITS process and which communicate using XML carried by a message bus. Figure 2.1 is a diagram showing this architecture. Subsystems, also called data providers, provide a server interface to other clients in the software system to access information and receive command requests for the devices or information they manage. Subsystems that manage ITS devices are on the row just below the databus. Below a subsystem are one or more driver modules whose sole responsibility is to communicate to the actual device using the device's protocol while translating messages back and forth from the device to the subsystem using the SunGuide software inter-module protocol. Other modules shown above the databus do not manage ITS devices, but do interact with the rest of the software system in the same way as device subsystems. Further details can be found in the system design documentation referenced by section 6.1 on page 53.

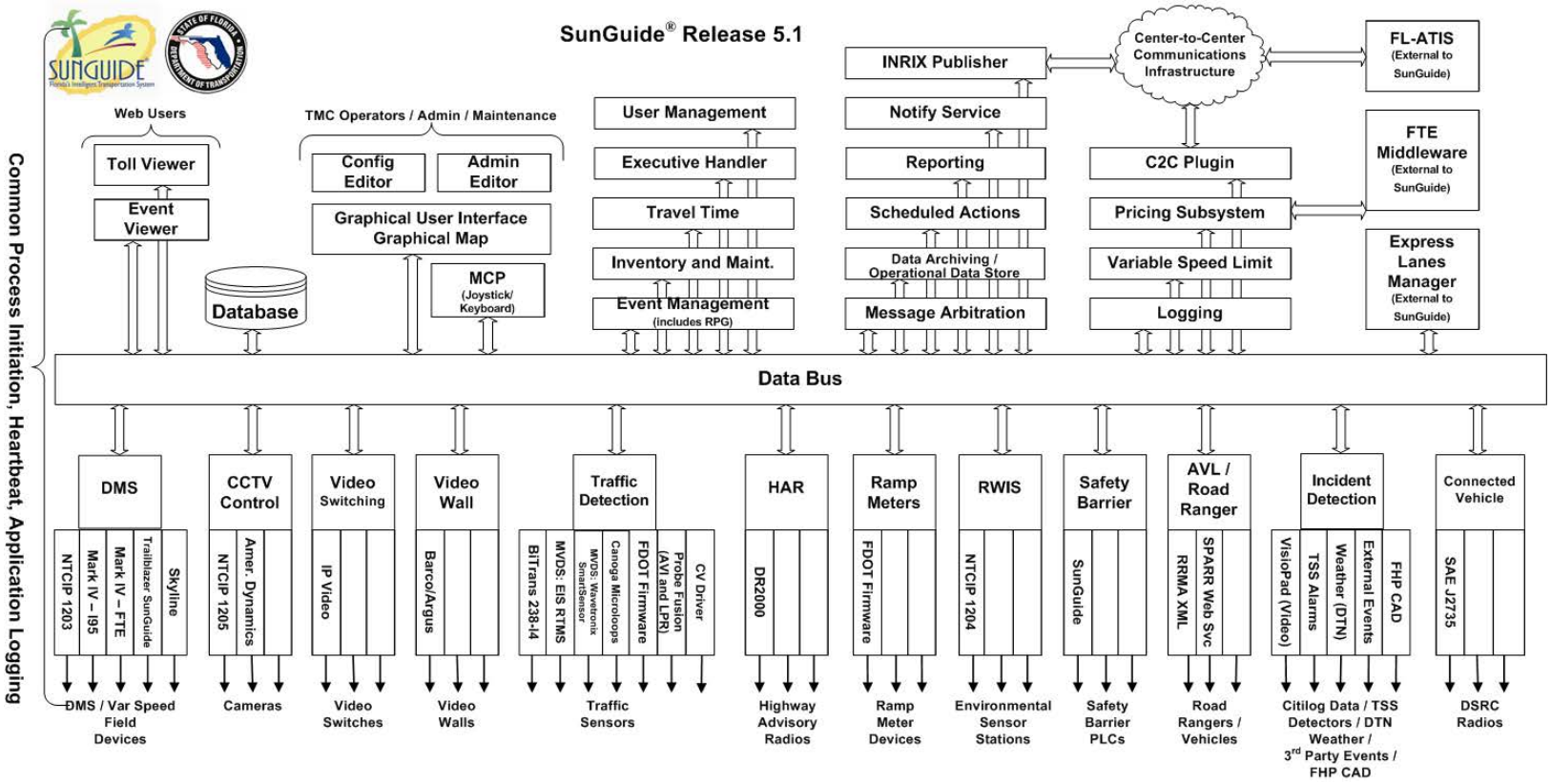

Figure 2.1: SunGuide Software Architecture Diagram

There are a few modules that provide infrastructure support, such as the database, databus, C2C infrastructure, and some user interface (UI) modules.

2.5.2 XML Protocol

SunGuide software uses an XML, client-server, TCP/IP for all internal communications between modules. The general ICD for this protocol can be found on the SunGuide software project web site and referenced in section 1.4. More specific protocols for each module can be found by navigating to the ICDs section of the SunGuide software web site.

Included with the ICD are the XML schemas that describe the detailed grammar of the XML messages in such a precise way that a machine can use the schemas to validate messages and even create sample messages. The set of schemas at the general level is available online at:

http://sunguide.datasys.swri.edu/ReadingRoom/ProjectDocuments/ICDs/SunGuide-General-ICD-3.0.1.xml.zip

While the details of the ICDs are deferred to the referenced documents, the differences with Lonestar will be explored and included later in this document.

2.5.3 Database

SunGuide software currently uses an Oracle 11G R2 database; however, support for SQL Server 2012 will be added in calendar year 2013. Each subsystem is responsible for managing their own data, including configuration and operational data, and exposing the data through an XML server interface that follows the SunGuide software XML protocol.

While each subsystem typically has tables specific to its needs, some tables are used more globally. These include tables for user authentication and permissions, a table for IDs assigned to each subsystem, etc. A detailed investigation to identify tables that have a global scope will be performed in Phase Two of this analysis.

The database design can be found on the SunGuide software project web site and is referenced in section 1.4. Starting in middle of 2013, future documentation of the database will be provided as an ERwin database model.

In 2013, all of the event management and other business logic will be removed from stored procedures and triggers from database. This will be helpful as it is more consistent with the architectural model and will make it easier to combine the systems because the logic will be better contained in the subsystem and not a combination of the subsystem and the database.

2.5.4 Databus

Databus provides a means for efficient data sharing among non-driver modules in SunGuide software. Without databus, users would have to directly authenticate and make requests to data

providers. Databus acts as a conduit to the data provider, passing authentication requests and data requests to the data provider and returning the responses to the user. By brokering this connection, fewer individual clients have to establish a network connection to the data provider.

Subscriptions are an information only, non-solicited part of the protocol that allows users to subscribe to a type of information once and continually receive updates as they are available. Databus stores a cache of these updates and distributes them to all subscribed users, rather than the data providers providing separate updates to each subscribed user individually.

2.5.5 Graphical UI

The UI is rendered within the Internet Explorer web browser that initially connects to a SunGuide software web server, which hosts UI files. The browser downloads the browser-based application and executes the application on the client machine. This application is an ActiveX control that executes .NET code. This ActiveX control directly connects and logs into databus as a SunGuide software client. Databus acts as a proxy to log into available subsystems as a more efficient data sharing platform allowing exchange of SunGuide software data.

The non-architectural details of the UI behaviour are highly dependent upon the ConOps. There are near-term plans to enhance the platform and quality of the entire SunGuide software UI.

The operator map data source is provided by NAVTEQ on a statewide contract basis to FDOT. The SunGuide software development team processes the data into image tiles to achieve significant performance gains for rendering on the user's workstation.

2.5.5.1 Configuration

Configuration is performed primarily in two locations. The config.xml file defines the static behavior of SunGuide software processes. This includes network addressing, subsystem naming, communication parameters, and various parameters that do not change while the program is running and that are behavioral in nature. Configuration of items that are more operational in nature, such as device configuration, editing location, message templates, and other items, are performed by a SunGuide software client using the XML protocol and can be updated during operation. Currently, there is a web-based tool called "Admin Editor" that facilitates this configuration task; however, SunGuide software is beginning to move more of the configuration activities into the operator map itself.

2.5.6 User Security

SunGuide software authenticates against an application-specific user database. Operators log in as a SunGuide software client to the subsystems when logging into the operator interface. This allows them to perform transactions over the SunGuide software XML interface. Modules also perform transactions in the same way. The module operating as the client logs into the module acting as a server with a SunGuide software user account.

SunGuide software also has a global set of user permissions. Permissions are associated to a privilege within a single subsystem. SunGuide software does not differentiate how a privilege is applied to any particular object or device or location in the system; thus, device-based, or county-based permissions are not possible with SunGuide software at this time. This capability has been expressed by SunGuide software users as a desired enhancement. A set of permissions can be stored as a group, and a group of permissions can be applied to a user. This action fully replaces the permissions of the user to the group's permissions. However, SunGuide software does not have the full user-group concept implemented whereby a group's permissions can be changed affecting all users in a particular group. A user is not actually in the group; this group is just an efficient way to immediately apply the current set of permissions saved to the group to a user.

2.6 System Modules

The software system processes build upon the architecture described in the previous subsection. Some of the business logic features and any deviations from the general architecture will be discussed in the following subsections. They are listed in the high-level feature comparison table in section 4.4 on page 28.

2.7 External Interfaces

2.7.1 Center-to-Center

There is a current effort to evaluate the differences between the SunGuide software and Lonestar software versions of C2C and to merge the two products back into one as a separate effort. The APPENDIX contains the results of this effort.

2.7.2 Third-Party Traffic Data Providers

FDOT receives traffic data from INRIX, a third-party data provider. The C2C component in the SunGuide software is compatible with data received from INRIX. SunGuide software interprets the provided data as transportation sensor data in a manner consistent with other traffic devices. It provides speed, but not volume or occupancy. BlueTOAD is another third party data provider integrated into SunGuide software through a C2C component.

Although there are other potential third-party data providers, they are not integrated into SunGuide software; however, the C2C schema could be used for other third-party providers to develop their own component capable of providing data to the SunGuide software.

2.7.3 Third-Party Incident Data Providers

FDOT receives weather feeds via a statewide contract with Telvent DTN. The SunGuide software incident detection system retrieves the weather data feed and integrates it into SunGuide software as weather alerts.

FHP provides FDOT with incident data from their computer aided dispatch system. This is another driver within SunGuide software's incident detection subsystem.

3 Lonestar Overview

3.1 History

3.2 Stakeholders

3.2.1 Stakeholder Categories

Stakeholders are considered as anyone affected by the Lonestar software project. They are categorized as:

1. **Motorists and the general public** who benefit from (and pay taxes for) a roadway transportation system that provides safety and mobility.

2. **TxDOT Districts** are geographical divisions within Texas that have responsibility of managing limited-access roadways and some arterial roadways within the statewide network. They deploy and make use of the software in a production environment.

3. **Local agencies** have more localized jurisdiction and may enter into an agreement with TxDOT to use the software for their arterial and local facilities.

4. **Southwest Research Institute** is the contractor that provides support, maintenance, and development services. They are the current licensing agent between Texas and Florida for sharing the software source code between the states.

5. **Traffic Operations Division (TRF)** manages the software deployment to applicable Districts.

6. **TMC managers** oversee the software use by the TMC operators. They also use the reporting, and other management functions of the Lonestar software.

7. **Technical administrators** support the deployment and use of the software. They may setup the servers and workstations, and use the configuration functions of the software to prepare it for the TMC operators to use.

8. **TMC operators** use the software directly to manage incidents and perform day-to-day traffic operations tasks.

9. **ITS contractors** perform a variety of tasks including management, specification, development, testing, training, demonstrating, analysis, technical support, etc.

Users are categorized by agency, TMC, and finally by the staff and their role within the group. The TMC is a general way to refer to a user, as there are a diverse number of roles and needs within the agency's TMC operations that are considered together and executed by personnel within that agency.

3.2.2 Current Users

Texas is geographically divided into 25 Districts. The Districts manage Texas' major roadways using the Lonestar software. Each TMC has slightly different operational needs for Lonestar usage and this warrants discussion and coordination at the change management board to ensure compatibility with each other and provide a seamless network of ITS functions along Texas' major transportation corridors. There are no out-of-state Lonestar software users.

Other users of the Lonestar software include TxDOT city and county partners as well as the media. Some media agencies have entered into an agreement with the local TxDOT district to control cameras and switching via a Lonestar client. Video for these agencies is arranged independent of Lonestar.

3.3 Operation

Lonestar is comprehensive software that fulfills the role of an ATMS software and is used for a wide range of operations related to traffic management. The following sections provide information on the operational categories that are supported by the Lonestar software.

3.3.1 ITS Device Control

Lonestar software connects to and communicates with ITS field devices. Devices report their health, status, and any operational data or responses back to the software, which sends commands and requests to the devices. Lonestar software makes real-time data received from devices available to all other Lonestar processes via the command and status distribution (CSD) process. The Lonestar database contains configuration and other pertinent information about device identification, location, and communication parameters as well as other details of how the software should behave.

There are several types of ITS devices, including detection devices, which primarily provide data input to Lonestar, and DMS devices, which apart from reporting their status, accept commands to send information out from Lonestar to motorists. Not all detection and dissemination requires physical devices managed by Lonestar. Lonestar has a C2C interface for external systems to provide live traffic data in lieu of field devices as well as providing a channel for Lonestar to send information out for dissemination.

Lonestar integrates these devices and data interfaces in many ways that are useful to other Lonestar software operations. Lonestar has several software processes that are not device-specific to process data and make it usable for various operations. Travel times and event management are two examples.

3.3.2 Traffic and Incident Management

Traffic and Incident Management is the primary purpose of the Lonestar software and the ITS devices and interfaces and is accomplished by Event Management. TMC operators use the

software to view traffic conditions and respond to traffic events, both planned and unplanned. The operator uses the software to verify, track, and record the incident, and to coordinate a response to the event with other agencies.

Many Lonestar software processes utilize the data available in the system to enable, automate, and track tasks that would otherwise be done manually by a TMC operator. Lonestar software automatically posts travel times. Response plans containing suggested Lane Control Subsystem (LCS) messages and DMS messages warning motorists of construction or blocked lanes ahead are automatically suggested and sent out to the LCSs and DMSs after operator approval.

3.3.3 Reporting and Data Analysis

Lonestar software currently supports limited reporting and data archiving; however, these are subsystems that are currently in the design phase.

3.4 Development Processes

The ITS architecture and design/build approach calls for using systems engineering principles as specified by TxDOT's TRF. Lonestar follows this process with thorough product configuration management (PCM). The following sections highlight a couple of the steps in the systems engineering process that are relevant to the scope of this document to compare SunGuide software with Lonestar for the consideration of melding the two software systems together.

3.4.1 Configuration Management

TxDOT uses a CMB for PCM. The CMB is comprised of TxDOT TRF personnel and is responsible for development of policies and procedures for products, including the statewide integration program; the CMB oversees implementation of the policies and procedures.

CMB responsibilities include reviewing and approving issues for the given TRF Traffic Engineering (TE)-ITS managed products. The CMB is responsible for reviewing, approving/rejecting, prioritizing, recommending issues be placed in a change request (CR), and reviewing approved CR for the given TE-ITS products.

The CMB monitors and manages issues that affect the product artifacts for which TRF TE-ITS is responsible. The CMB's charter is to properly maintain TRF TE-ITS products, control the changes to TRF TE-ITS products, monitor the quality of TRF TE-ITS products, and monitor the release of TRF TE-ITS products. The CMB meets on a regular basis as defined by the configuration manager. The CMB provides a periodic status update to TRF management, including a list of issues being tracked with an issues tracking system tool, a configuration status report on the TRF TE-ITS products under PCM, a report listing the man-hours used to address the issues from the previous month, and a report listing the estimated man-hours projected to be used for issues in the next month. If the configuration manager is not available, then an alternate is appointed to take action.

PCM is an integral part of the software process lifecycle. PCM is used to establish and maintain the integrity of product artifacts throughout the software lifecycle. These product artifacts include software artifacts, such as documents and source code, as well as hardware artifacts, such as drawings and physical hardware. PCM for hardware artifacts will need to be established based on the tools used to develop the drawings and other hardware artifacts.

3.4.2 Requirements

TRF TE-ITS manages the changes made to the products that it controls by requiring that issues be documented using the approved issues tracking system. For TRF TE-ITS, each issue is entered into the issues tracking system. Issues are then analyzed by the CMB and categorized. Categories include software problems, enhancement requests, or general support items. Approved issues provided to the product manager are converted to a CR, which is submitted to the TRF TM-ITS management for review and approval.

Issue tracking involves the following steps:

- Issue submission,
- Issue investigation,
- Estimate issue cost and schedule,
- Issue prioritization, CMB approvals, and CR Generation, and
- CR approval by TRF TE-ITS management.

If necessary, the system requirements specification (SRS) document is developed from the ConOps document. Development of requirements includes a detailed description of the requirement and a reference to the component of the ConOps to which it refers or from which it was derived. A traceability matrix is included to ensure each requirement maps to an appropriate test case. Once these documents have been created, they are provided to the CMB, which reviews them and provides comments. Necessary changes are made to the documents and the review process is repeated.

3.4.3 Design

Once the SRS and ConOps documents are approved, the design phase starts. Design documents are produced, which include several deliverables, as necessary. The deliverables include prototype screen shots, an ICD, a subsystem protocol document if needed, and a SDD.

The ICD defines how Lonestar subsystem modules communicate with one another. ICDs are provided as a document describing the interface as well as a set of XML schema files that can be used by a computer to understand, parse, and validate any message that should be compliant to the defined message structure.

The SDD is a detailed design document containing the system and subsystem components, design decisions, concept of execution, and subsystem handlers for XML requests, messages, and responses.

The design and design documents are provided to the CMB and reviewed, and comments provided. TxDOT provides comments and the development team updates the design to reflect those comments. Necessary changes are made to the documents and the review process is repeated.

3.4.4 Implementation

The implementation process is done after the design phase and prior to the ATP phase and is thus the development team's responsibility to define, follow, and control this process.

3.4.5 Testing

Testing occurs in three distinct phases. Unit testing is integrated into the software implementation process. The development team uses a combination of peer code reviews and manual testing to ensure the quality of the units. As a part of the implementation process, this is the development team's responsibility to define, follow, and control this process.

The software development team conducts a dry run test at the TRF laboratory. The developer produces an ATP document, which outlines the plan and procedures, respectively, for the testing activity. Once the dry run is complete the software is delivered to TRF for formal testing.

The formal ATP is followed for the final testing before release. This test is also performed in the TRF lab to verify the product is ready for release; it is typically conducted by TRF or an independent contractor.

If any issues are found in the any of the testing, the developer corrects the issues and submits the correction for retesting. After the software is determined to meet all of the requirements, it is accepted by TRF and ready for deployment and production use.

3.4.6 Release Cycles

Enhancements and bug fixes are organized into releases that depend on the needs of the users, the schedule of those needs, and a reasonable level of work for the contractor to accept. Typically, each year has a major release, one or two moderate releases, and one or two minor releases.

Software releases are categorized as major, moderate, and minor. Major releases involve significant change to the software, follow all steps in the software development process, and an entire set of development process documents are developed or updated. Moderate and minor releases may defer updating some of the process documents to the next major release. Patches are minor with specific modifications intended to meet urgent needs for specific users who cannot wait for an entire release cycle.

3.5 System Architecture

3.5.1 Modular Relationships

Lonestar software is made up of subsystems and applications, each with its own specific functionality. Figure 3-1 illustrates the software architecture, which is laid out such that the subsystems that communicate with ITS devices are shown below the Command & Status Distribution box. The applications are shown above the Command & Status Distribution box and are those processes that handle data requests and messages between the users and the subsystems. The shaded components denote future development.

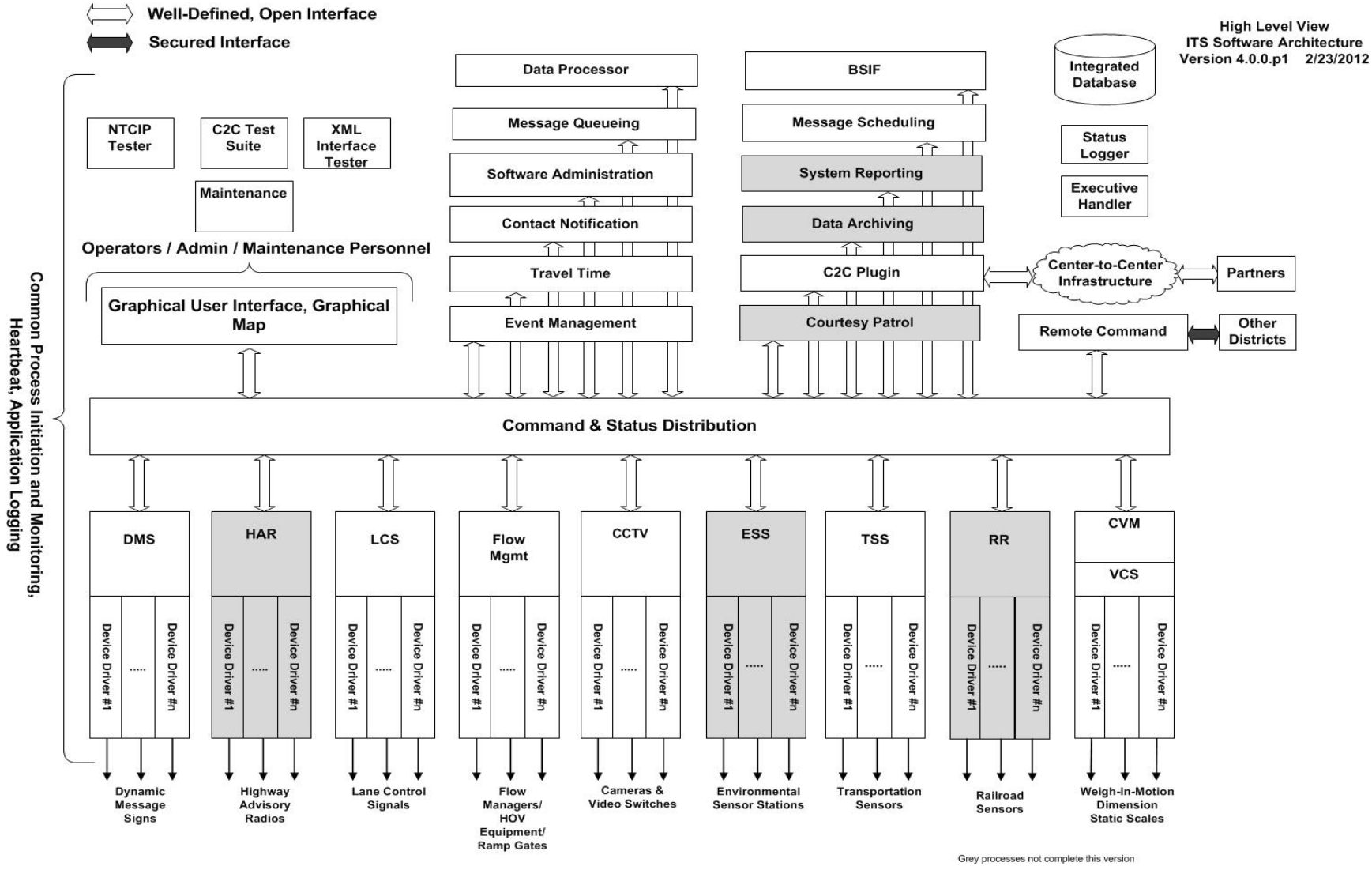

Figure 3-1: Lonestar Software Architecture Diagram

3.5.2 XML Protocol

Lonestar uses XML as its primary method of data transportation between Lonestar subsystems, applications, and the client UI via a client-server, TCP/IP. Well-defined schemas are used to request and receive data from CSD. An ICD for each subsystem or applications containing the detailed format of the XML requests, responses, and messages is available from TxDOT and is included in every product install. The XML schemas are also included in every product install in a compressed file for interested parties to access.

3.5.3 Database

Lonestar stores and retrieves configuration data, status data, and other persistent data to a Microsoft SQL Server 2008 database for all districts except Houston. Lonestar also provides an Oracle 9i database interface, for some subsystems and applications which Houston is currently using. SQL scripts are used during the Lonestar install process to create the database and populate it with some general data. District-specific data is entered either manually or through other means such as the XML Tester application.

Each subsystem or application may have tables that are specific to their requirements. Common tables are used for data that contain data with a global scope. The database design is maintained in files which can be accessed using ER/Studio.

3.5.4 CSD

CSD is the subsystem that distributes data between subsystems and applications using XML-formatted messages, requests, and responses. Requests may be for retrieving data, status, or requesting to subscribe to a particular type of data. If the specified data changes at a later time, the client will receive unsolicited responses with the updated data.

3.5.5 UI

The Lonestar client UI is the primary graphical interface to the Lonestar system for operators. The UI offers control to Lonestar's range of subsystems. From this interface TMC operators have graphical control of various devices as well as access to other ITS information. The software communicates with subsystems through plug-ins that are loaded at run time. This feature allows new subsystems to be developed and added easily.

Lonestar has a Windows-based UI that runs on a Windows XP or Windows 7 operating system. Microsoft MapPoint is utilized for drawing the base map used to display cities, roads, and landmarks as well as Lonestar equipment. The map interface has a menu bar containing all subsystems whose plug-ins have been loaded. Icons can be displayed on the map for equipment. Context sensitive menus are available for any icons displayed for command, control, and status of the equipment.

Each subsystem also provides a stand-alone UI that can be run independent of the map interface.

3.5.5.1 Configuration

Configuration parameters for system wide or subsystem-specific information are contained in the config.xml file. These include parameters, such as center location information, database information, network addressing, communication parameters, logging level, and subsystem-specific parameters that are not generally changed during the execution of the processes.

Operational configuration information is stored in the database and is updated via the Lonestar UIs. Operators are able to add, update, and/or delete device configuration, roadway information, message templates, and other information by utilizing the windows, provided they have appropriate permissions.

3.5.6 User Security

Users are assigned a unique username and password and are required to change the password on a periodic basis for security precautions. User accounts can be assigned permissions specific for each Lonestar subsystem and application. Permissions may be given to a user through a group that has already been created with specific permissions. Individual permissions may also be added and/or deleted on a per user basis. Each subsystem may also provide types of permissions as well as rights to particular pieces of equipment.

3.6 System Modules

Two Lonestar processes run as Windows services and can be configured to start automatically or manually by an administrator. These are Executive Handler and Status Logger. Other services are configured through the Executive Handler Editor. Their current status can be viewed and they can be stopped and/or started and the logging level changed through the Executive Handler Viewer.

3.7 External Interfaces

Each subsystem may support a variety of external interfaces.

3.7.1 C2C

There is a current effort to evaluate the differences between the SunGuide and Lonestar software versions of C2C and to merge the two products back into one as a separate effort. The APPENDIX contains the results of this effort.

3.7.2 Other Agencies

With proper authorization, external agencies are able to query the Lonestar software for information on devices, status of devices, and even command and control devices using the XML interface.

4 High-Level Comparison

While both the SunGuide and Lonestar software systems are very similar, their operational use, architecture, and design structure have differences that prevent enhancements from being shared. These differences are:

4.1 Operational Differences

- Both manage ITS devices, but some ITS devices are only supported by one or the other software system. Deployment architects, vendors, and technicians expect certain products to be supported by the existing system. A detailed list is provided in section 4.4 on page 30.

- Both provide incident (event) management; however, SunGuide software users rely on the software to participate in incident detection as an input for event notification. SunGuide software operators also take on the responsibility of dispatch and management of FDOT-sponsored Road Ranger service patrols, and the TMC performs extensive performance measures and reporting using the detailed, time-stamped data collected by the software and provided by the extensive reports. Lonestar operators receive information from emergency personnel, calls from travellers, and operators monitoring camera video to detect incidents and manually enter traffic events into the system. Lonestar also provides alarms and alerts via Data Processing Application (DPA) for operators. Alarm and alert rules can be created on remote events, link information (speed and occupancy), pump station information, wrong way detection, and Advance Warning to Avoid Railroad Delays (AWARD) items. TSS can also be configured to create alarms for speed and occupancy thresholds for a link. When alarms are generated from DPA via a remote event, a local event can be created or viewed. TxDOT no longer performs dispatch and management of Courtesy Patrol service patrols as it is managed by the local city entities. TxDOT Districts have varying policies on recording responder arrival and departure times and generating reports using this information.

- Another group of SunGuide software stakeholders use the archived data for research, analysis, and even training and demonstrations. Lonestar does not yet have a full fledged data archiving component or a CDW, but the data archive application is under development.

4.2 Development Process

- TxDOT's PCM and FDOT's CMB both have similar processes; however, they are comprised of different members who represent different users and stakeholders. These groups will not be able to be consolidated or combined, requiring coordination between the groups to ensure the needs of both are met.

- Independent testing done by FDOT develops verification and validation test plans and procedures from the ConOps independently from the FAT documents. TxDOT reuses a single set of ATP for their independent testing.

4.3 Architecture

4.3.1 XML Protocol

- XML is used by both software systems. The XML protocol defines the format used to encode the data into XML for transmission between components.

4.3.2 Database

The SunGuide/Lonestar software database comparison will be performed to determine the compatibility of the two ATMS databases. This analysis will examine the ability to merge the two database models into a single unified model that supports both SunGuide software and Lonestar. This analysis will perform a high-level review of the effort to merge and implement the two data models as a single data model, the changes required, and the complexity of dealing with conflicting database entities. Assumptions and recommendations for further analysis will be documented.

The current SunGuide software database management system (DBMS) is Oracle. The SunGuide software database is currently being modified to support both Oracle and SQL Server. This analysis is based on the latest version of the Oracle implementation for release 5.1.1. There will be modifications that are not reviewed for this analysis that may have an impact on the results. The LoneStar DBMS is SQL Server with some Lonestar subsystems and applications also supporting Oracle. The analysis is based on the latest deployed version of the database. The DBDD used for the analysis is IDB 3.2.0 date July 26, 2011. It was reported that there have been some minor updates to the database model, but none that would affect the results of this analysis.

Several differences in the database exist for the two ATMSs. Some modules only exist in one or the other ATMS. There will be tables for these modules that exist in the database for one ATMS, but do not exist in the database of the other ATMS (e.g., SunGuide Safety Barrier tables). Also, modules that were developed in parallel and through completely different efforts may have different tables in their respective software systems even though they may have a similar operational purpose. Finally, tables may be very similar for modules that were developed initially in one system and then ported over to the other system.

4.3.3 Data Distribution

- CSD was built based on Databus and their primary functionality is identical. However, during phase two, there will be a detailed investigation to identify any differences that have been introduced since the inception of Databus.

4.3.4 UI

- SunGuide software has a web-based platform to launch a combination of HTML dialogs, and embedded .NET application code using the Windows Presentation Foundation from Microsoft to compose its graphical user interface (GUI). The map data is licensed to FDOT as layers in shape files and is pre-processed into image tiles made available by the SunGuide software webserver. Lonestar uses a Windows forms based (non-broswer based) application for the entire GUI. TxDOT licenses seats of the Microsoft Map-Point software and Lonestar integrates the map into the GUI application.

- Both DOTs have made significant investments to their existing, licensed map data – one from a map data provider as a single, statewide license and another from a commercial off the shelf application licensed per human user. Thus, both approaches to rendering the map in the GUI shall continue to be supported in the combined system.

4.3.5 User Security

- While both software systems have support for password-protected user accounts for individual actions within each module, Lonestar additionally grants access on a per devices basis. Lonestar has a standalone shared services module that manages and provides authentication support for all other modules.

4.4 Module-by-Module Comparison

Table 4.1 shows the high-level modules included in either system. The table also specifies the module name in either system if it exists in that system.

Table 4.1 – High-Level Feature Comparison

Component	SunGuide Software	Lonestar
Secure traffic event viewer interface	Event Viewer	N/A
Toll viewer	Toll Viewer	N/A
GUI, Graphical Map	GUI, Graphical Map	Map Interface
Standalone UI	N/A	Standalone UIs for each subsystem
CSD system	Databus	CSD
Configuration editor	Config Editor Application	N/A
Operational configuration (user and device, message templates, etc.)	Admin Editor	Software Administration Application
Joystick for camera	Manual Control Panel (MCP)	MCP
Handler for all server processes	Executive Handler	Executive Handler
Status and event logging	Status Logger	Status Logger
Event management	Event Management	Event Management

Component	SunGuide Software	Lonestar
Incident detection / data processing	Incident Detection	DPA
Message arbitration and queuing	Message Arbitration Subsystem (MAS)	Message Queuing Application (MQA)
Scheduling	Scheduled Actions	Message Scheduling Application
Notification process	Notify Service	Contact Notification Application
Travel time	Travel Time	Travel Times Application
C2C	C2C Plugin	CSD C2C Plugin
Reporting	Reporting	*System Report Application (in development)*
Data archiving	Data Archiving/Operational Data Store	*Data Archive Application (in development)*
Remote command	C2C Command Receiver	C2C Command Receiver and Remote Command Application
Border safety	N/A	Border Safety Inspection Facility
Pricing	Pricing Subsystem	N/A
VSL	VSL	N/A
Inventory and maintenance	Inventory and Maintenance	N/A
Dynamic message sign (DMS)	DMS	DMS
Closed-circuit television (CCTV)	CCTV Control, Video Switching, and Video Wall	Camera Manager, CCTV Master
Traffic detection	TSS	Transportation Sensors Subsystem
Environmental and weather information	Road Weather Information System (RWIS)	Environmental Sensor Stations Lite
HAR	HAR	*N/A*
Automatic vehicle location (AVL)	AVL/Road Ranger	*N/A*
Commercial vehicle management and compliance screening (for border safety)	N/A	Commercial Vehicle Management/Vehicle Compliance Screening
Flow manager (for border safety)	N/A	Flow Manager
Lane control subsystem	N/A	LCS
Ramp metering	Ramp Meters	N/A
Safety barrier	Safety Barrier	N/A

4.5 Device Protocols Supported Comparison

Table 4.2 shows the protocols supported by each system. The table also shows which system(s) supports the protocol.

Table 4.2 – Device Protocols Supported Comparison

Device Type	Device Protocol	SunGuide Software	Lonestar
DMS	NTCIP 1203	X	X
	Trailblazer SunGuide	X	
	Mark IV - FTE	X	
	Mark IV - 195	X	
	FDS 16-bit		X
	FDS 8-bit		X
	TeleSpot		X
CCTV Control	American Dynamics	X	X
	NTCIP 1205	X	
	Vicon		X
	Cohu		X
	Pelco D		X
	Axis		X
	Quest		X
Video Switching	IP Video	X	X
	ONVIF	Future	
	Analog Cybermation System 6E		X
	ATM		X
	DACS3		X
	Quest		X
Video Wall	Barco/Argus	X	
	Jupiter	X	
	Activu	X	
Traffic Detection	FDOT Firmware (for the 170 controller)	X	
	Canoga Microloops	X	X
	MVDS: Wavetronics SmartSensor	X	
	MVDS: EIS RTMS	X	X
	BiTrans 238-I4	X	
	ISS Autoscope		X
	TxDOT TSS Protocol		X

Device Type	Device Protocol	SunGuide Software	Lonestar
	Transcore's eGO 2210 RFID		X
	Wavetronix HD		X
	Wavetronix's RTMS	X	X
	ISS G4		X
	Naztec LCU		X
	Austin LCU		X
	Austin SCU		X
	Inex Zamir Zap	X	X
	Transcore Allegro	X	X
	SIRIT Flex	X	X
	Probe Fusion	X	X
RWIS	NTCIP 1204	X	
	XML Protocol		X
HAR	DR2000 - SIM	X	
AVL/Road Ranger	XML Interface	X	
Ramp Meters	FDOT Firmware	X	
Safety Barrier	SunGuide	X	
Incident Detection	FHP CAD	X	
	External Events	X	
	Weather (DTN)	X	
	TSS Alarms	X	X
	VisioPad (Video)	X	
	Pump Station		X
	Wrong Way Detection		X
	AWARD		X
LCS	FDS		X
	NTCIP		X
	SCU-1		X
	SCU-2		X
ESS-Lite			X

5 Module Comparison

The modules are compared below, starting with the global modules that are lower level and integrated into the rest of the system and followed up by the modules that provide business logic which includes a GUI component.

5.1 XML Schema and Protocol

The XML Protocols have a schema definition with a very similar organization. Most subsystems have their own folder with subfolders containing object, messages, requests, and responses. There is also a common folder containing elements shared by many other components. There are over 3,000 elements between the SunGuide software and Lonestar XML Schemas. The differences between the two sets will primarily be the existence of different modules that the other system doesn't have (for example, Connected Vehicle Subsystem) and some minor tweaks to account for feature differences in modules that were derived between one system and the other (for example, the geofence object in SunGuide software's AVL that is not present in Lonestar's Courtesy Patrol).

5.1.1 Comparable Subsystems

SunGuide software and Lonestar have several compariable subsystems whose XML schemas could be analyzed side by side to determine detailed differences. Table 5.1 lists these subsystems and a summary of differences. Each of these differences would need to be reconciled in order to merge the two systems.

Table 5.1 – Subsystems and Summary of Differences

Schema Grouping	Subsystem Names		Differences	Required Actions
	SunGuide Software	Lonestar		
Camera control and viewing	cctv, vs, vw	cctvMaster, cctvCm, cctvSnapshot	Systems separate cctv functionality differently. Lonestar separates common schemas (cctvMaster), command schemas (cctvCm), and snapshot schemas (cctvSnapshot), while SunGuide software separates control schemas (cctv), video switching schemas (vs), and video wall schemas (vw). SunGuide software currently does not support snapshot schemas.	Functionality would need to be originized in common groupings. The additional snapshot schemas could be added without negatively affecting SunGuide software.

Schema Grouping	Subsystem Names		Differences	Required Actions
	SunGuide Software	Lonestar		
C2C software, data sharing and control system	c2c	c2c, csdc2cplugin	Systems organize schemas comparitively. SunGuide software handles floodgates and additional event editing through C2C. See the C2C comparison in the appendix for details.	Differences should be reconciled.
Common schemas	common	common	Minor differences between supported schemas.	Differences should be reconciled.
Data archiving	dataArchive	dataArchiver	Minor differences between supported schemas.	Differences should be reconciled.
DMS control and status	dms	dms	Minor differences between supported schemas.	Differences should be reconciled.
Data distribution	dataBus	csd	Minor differences between supported schemas.	Differences should be reconciled.
Event management	em, incident	em, cna	Large number of differences between how events are handled and changing/updating event details. SunGuide software's incident subsystem contains several older schemas which may no longer be used. SunGuide software's response plan functionality is included in the incident subsystem. Lonestar has a more extensive contact notification subsystem which is only covered in part in the SunGuide software schemas.	Differences should be reconciled and combined into a single em subsystem. The reconcile process may be difficult due to the large number of differences in how events are handled. Deprecated schemas should be removed.
Incident detection	ids	dpa	SunGuide software's IDS and Lonestar's DPA are fairly similar in functionality	Differences should be reconciled.
Message management	mas	mqa	Minor differences between supported schemas.	Differences should be reconciled.
Notification Subsystem	ns	ns	Modules are comparible	Differences should be reconciled.
Reporting Subsystem	rs	sra	A Reporting component (SRA) is under design/development for Lonestar. This should relate	Differences should be reconciled.

Schema Grouping	Subsystem Names		Differences	Required Actions
	SunGuide Software	Lonestar		
			to SunGuide software's RS module	
Transportation sensor monitoring	Tss	tss, protocol/tss, dpa	Lonestar separates TSS alarm schemas into protocol/tss and dpa subsystems.	Differences should be reconciled, requiring either the splitting out the TSS alarming or combining them into a common tss subsystem.
Travel time calculations and management	Tvt	tta	Minor differences between supported schemas.	Differences should be reconciled.
Weather data monitoring	Rwis	ess, protocol/ess	Lonestar's environmental sensor station (ESS) handles a simpler form of weather data than SunGuide software's RWIS.	Lonestar should be expanded to include RWIS NTCIP functionality.

5.1.2 Unique SunGuide Software Subsystems

SunGuide software contains several unique subsystems. Table 5.2 lists these subsystems.

Table 5.2 – Unique SunGuide Software Subsystems

Schema Grouping	SunGuide Software Subsystem Names	Required Actions
511 system control and monitoring	511	Subsystem is no longer used in SunGuide software.
AVL and Road Ranger management	avlrr	Need to add to Lonestar.
Connected vehicle control and monitoring	Cvs	Need to add to Lonestar.
FHP data handling	Fhp	Need to add to Lonestar.
GUI data handling	Gui	Much of this functionality exists in Lonestar, but without the use of GUI-specific schemas. Need to add to Lonestar. Much of the SunGuide software GUI Preference Manager functionality is a part of the Lonestar SAA subsystem.
HAR device management	Har	Need to add to Lonestar.
Inventory management	Ims	Need to add to Lonestar.
INRIX data monitoring	Inrix	Need to add to Lonestar.
Ramp metering	Rms	Need to add to Lonestar.

Schema Grouping	SunGuide Software Subsystem Names	Required Actions
Road Ranger management	avlrr	Need to add to Lonestar.
Safety barrier	sb	Need to add to Lonestar.
Scheduled actions, especially for camera control	Sas	Need to add to Lonestar.
Variable pricing management	Ps	Need to add to Lonestar.
VSL	vsl	Need to add to Lonestar.

5.1.3 Unique Lonestar Subsystems

Lonestar contains several unique subsystems. Table 5.3 lists these subsystems.

Table 5.3 – Unique Lonestar Subsystems

Schema Grouping	Lonestar Subsystem Names	Required Actions
Lane control	LCS	Need to add to SunGuide software.
Message scheduling (Note: this will be added to SunGuide 6.0)	Message Scheduling Application (MSA)	Functionality is included within SunGuide software, but schemas are not used because functionality is localized within DMS and not separated into its own subsystem. SunGuide software may need to be updated to separate this functionality.
System Administration	SAA	Need to add to SunGuide software

5.2 Database

The SunGuide software and Lonestar data models support their respective ATMSs. While the SunGuide software data model has its roots from the Lonestar data model, the data models have evolved somewhat independently; however, there has been more collaboration between FDOT and TxDOT on subsystem development and enhancements, thus creating the opportunity to better align the data models.

5.2.1 Assumptions

The following assumptions were made during the SunGuide software and Lonestar data model merge analysis:

1. The SunGuide software data model provided is accurate and any updates to the data model for implementing SQL Server will have no affect on the analysis or recommendations.

2. The Lonestar data model is accurate and any updates to the data model since IDB 3.2.0 will have no affect on the analysis or recommendations.
3. SunGuide 6.0 and Lonestar 5.0.0 releases may result in further database changes and divergent subsystems from the time of this document's release.

5.2.2 Data Model Analysis

Since the SunGuide software ATMS was designed and implemented from the Lonestar ATMS, one would expect the two data models to have similar underlying data model structures. Because the two ATMSs have advanced over the years, mostly independently until recently, the business logic and data models have diverged from one another. The data model design is dependent not only on how to best design and store the data for optimal use by ATMS, but also on the business logic used to implement the ATMSs. While FDOT and TxDOT both use the ATMSs to manage their roadway network, they have differing philosophies on how to implement their business rules for some of the subsystems. These philosophical differences can cause the two data models to be implemented quite differently for certain subsystems. For example, Lonestar defines all of its equipment/devices in a single set of tables while SunGuide software defines its equipment/devices in individual tables. The Lonestar tables are prefixed with an abbreviation of the applicable subsystem, which uses the table and, where there are multiple subsystems accessing a table, the prefix CT or OT is used. The SunGuide software tables do not all implement a prefix to identify the subsystem(s) that access a table.

5.2.3 User Permissions

User permissions are implemented differently in the two data models. SunGuide software has user permissions that provide the operator's ability to access the different subsystems and issue commands to different subsystems. Lonestar additionally allows groups of devices to be defined within each subsystem to which a user can access and issue commands as permissions allow. Both software products combine all user and subsystem permissions into a single set of tables.

5.2.4 Common Tables

A review of the two data models show that there are a small number of common tables. This review identified 424 tables between the two ATMSs. This number only counts common tables once. There were eight tables identified. A common table is defined as a table that has exactly the same name in both data models. Only a cursory review of the tables' columns was performed. Table 5.4 lists these common tables and the number of columns for each ATMS.

Table 5.4 – Common Database Tables

Table Name	Number of Columns	
	SunGuide Software	Lonestar
CCTV_PRESET	4	4
DMS_FONT_DATA	3	5
DMS_FONTS	6	6
DMS_SPECIAL_CHAR_LOOKUP	4	4
DMS_STATUS	22	23
DMS_SYSTEM_CONFIGURATION	14	15
TSS_DETECTOR	7	2
TSS_LANE	5	4

As can be seen from Table 5.4, even though the tables are likely used for the same purposes in both systems, there are some differences in the number of columns in the common tables that imply the business logic and data model design is implemented differently for each ATMS. In some cases, it is possible that multiple tables from one data model may be all related to a single table in the other data model.

A further examination of the CCTV_Preset was performed. Table 5.5 compares the columns defined in both tables.

Table 5.5 – Columns Defined in Both System's CCTV_Preset Tables

SunGuide Software		Lonestar	
Column Name	**Data Type**	**Column Name**	**Data Type**
REF_ID	NUMBER	**Preset_ID**	int
CAMERA_ID	NUMBER	**Preset_Number**	int
PRESET_NUM	NUMBER	**Preset_Description**	varchar(40)
PRESET_DESC	VARCHAR2(100 BYTE)	**Equipment_ID**	int

Comparing the differences in the common tables revealed the following observations:

- The columns are defined in a different order. This difference would require a minor modification.
- The data types would be handled by ERWIN when generating the database creation script. However, the SunGuide software columns PRESET_DESC and Lonestar column Preset_Description data type have to be resolved. The SunGuide software column is

defined as a VARCHAR2 (100 byte) while the Lonestar column is defined as a varchar(40).

- The columns are named differently. Common column naming would have to be implemented.
- SunGuide software column REF_ID and Lonestar column Preset_ID have to have the same meaning in both database models. CAMERA_ID has a foreign relationship to the CCTV_Equip table Resource_ID column. Preset_ID is defined as a system assigned identifier. While these two entities may serve similar functions, their implementation is not consistent between the two data models.

SunGuide software column CAMERA_ID and Lonestar column Equipment_ID have to have the same meaning in both database models. The SunGuide software device definitions will have to be moved to a common equipment table to make the SunGuide software data model more compatible with the Lonestar data model. Equipment_ID is a system-assigned value that is created in the CT_Equipment table and referenced by the CCTV_PRESET table.

5.2.5 Potential Common Tables

Further analysis of the tables identified more tables that could possibly be related. These tables do not have the exact same name, but have similar names; however the subsystem prefix is different. Each ATMS has many subsystems in common, but, in some cases, chose to use different naming conventions for these subsystems. For example, TTA is the prefix for Lonestar's Travel Time subsystem, while TVT is the prefix for SunGuide software's Travel Time subsystem. Table 5.6 provides a list of tables that likely are similar in structure and used to implement the business logic for the given subsystem.

Table 5.6 – Database Tables With Similar Structure

SunGuide Software		Lonestar	
Table Name	Number of Columns	Table Name	Number of Columns
VS_Active_Tour		**CCTV_ACTIVE_TOUR**	3
CCTV_Equip	8	**CCTV_CAMERA**	10
VS_Video_Tour	4	**CCTV_VIDEO_TOUR**	6
VS_Video_Tour_Data	3	**CCTV_VIDEO_TOUR_DATA**	4
CCTV_Equip, DMS, TSS_Detector or any table defining field devices	8 26 7	**CT_EQUIPMENT**	12
DMS_MANUFACTURERS, Manufacturers	2 2	**CT_MANUFACTURER**	2
Permission_Group	3	**CT_PERMISSION_GROUP**	3

SunGuide Software		Lonestar	
Table Name	Number of Columns	Table Name	Number of Columns
Roadway	5	**CT_ROADWAY**	3
Roadway_Direction	11	**CT_ROADWAY_DIRECTION**	3
Direction		**CT_DIRECTION**	2
Roadway_Link	7	**CT_Roadway_Link**	6
Roadway_Midpoint	5	**CT_ROADWAY_MIDPOINT**	5
Roadway_Node	4	**CT_ROADWAY_NODE**	4
Subsystem_Permissions,	4 3	**CT_SUBSYSTEM_PERMISSION**	4
APPROVED_WORDS_LIST	1	**DMS_APPROVED_WORD**	1
DMS	26	**DMS_DEVICE**	15
DMS_Font_Characters	3	**DMS_FONT_CHARS**	3
DMS_GROUP	1	**DMS_GROUP_NAME**	2
Group_DMS_Reltable	2	**DMS_GROUP_RELATION**	2
Message	8	**DMS_MESSAGE**	10
Polling_Names	1	**DMS_POLLING_NAMES**	2
Sequences	7	**DMS_SEQUENCE**	8
Sequence_Library	1	**DMS_SEQUENCE_LIBRARY**	2
Sequence_DMS_Reltable	11	**DMS_SEQUENCE_RELATION**	11
EM_Contact, **EM_Maillist,** **EM_Maillist_Contact**	17 13 6	**EM_EVENT_CONTACT**	2
EM_Event_Chrono	25	**EM_EVENT_HISTORY**	5
EM_LaneMap	18	**EM_EVENT_LANE_DATA**	5
EM_Location, **EM_Location_County**	28 6	**EM_EVENT_LOCATION**	5
EM_EventType	24	**EM_EVENT_TYPE**	2
EM_LaneType	20	**EM_LANE_TYPE**	2
RPG_Msg_Templates	13	**EM_MESSAGE_TEMPLATE**	3
RPG_Plan, **EM_Event_ResponsePlan**	2 15	**EM_RESPONSE_PLAN**	6
RPG_HAR_Plan_Item, **RPG_DMS_Plan_Item**	13 11	**EM_RESPONSE_PLAN_DMS_ITEM** EM_RESPONSE_PLAN_LCS_ITEM	12
RPG_Plan_Event_Type	2	**EM_RESPONSE_PLAN_EVENT_TYPE**	2
TSS_Driver_Data	3	**TSS_DRIVER**	1
TSS_Link_Lanes	2	**TSS_LANE_LINK**	2

SunGuide Software		Lonestar	
Table Name	Number of Columns	Table Name	Number of Columns
TVT_DESTINATIONS	2	**TTA_DESTINATIONS**	2
TVT_DEVICE_DEST	5	**TTA_DEVICE_DEST**	5
TVT_DEVICE_TEMPLATE	5	**TTA_DEVICE_TEMPLATE**	4
TVT_LINKS	7	**TTA_LINKS**	3
TVT_OPTIONS	4	**TTA_OPTIONS**	3
TVT_TEMPLATES	4	**TTA_TEMPLATES**	4
TVT_TSS_LINKS	3	**TTA_TSS_LINKS**	3
TVT_TSS_LINKS_REMOTE	3	**TTA_TSS_LINKS_REMOTE**	3

As with the common tables, even though the subsystem tables are likely used for the same purposes in both systems, there are some differences in the number of columns in the common tables that imply the business logic and data model design is implemented differently for each ATMS. In some cases, it is possible that multiple tables from one data model may be all related to a single table in the other data model.

5.2.6 Recommendations:

The following list provides recommendations for merging the two data models:

1. **DMS:** Since both ATMSs implement the basic concepts of DMS in a similar manner, it is likely feasible to merge each data models tables into a single set of tables.
2. **CCTV:** Since both ATMSs implement the basic concepts of CCTV in a similar manner, it is likely feasible to merge each data models tables into a single set of tables.
3. **Message Queuing:** Both systems have message queuing subsystems. Since the basic concept of both of these subsystems is to queue messages, it is likely that these two table sets could be merged into a single table set.
4. **Detection:** For TSS, SunGuide software and Lonestar support similar devices, implement similar functionality, and share common drivers. While further examination of how each ATMS defines links needs investigation, it is likely these two table sets could be merged into a single table set.
5. **Travel Time:** Based on the fact that the tables from both data models have tables that are very similar, it is likely that these two table sets could be merged into a single table set.
6. **User Permissions:** The FDOT CMB has approved the adoption of the Lonestar user permissions implementation.
7. **Event Management:** The Event Management tables seem to have a large gap in how the tables are organized and the level of information stored in these tables appears to be

diverse enough to not attempt to create a single subsystem. At this time, do not merge the Event Management tables.

8. Both ATMS data models have moved to a more generic approach on how they implement certain entities. For example, SunGuide software has generic implementation of messages, sequences, equipment, and approved words. While Lonestar started out as a subsystem-based implementation (tables were created specifically for the subsystem), it has migrated to a more generic approach for equipment. It is likely that these generic approaches can be implemented without a large impact to either subsystem.

5.2.7 Conclusions:

The following conclusions were derived from this analysis:

Generally, the databases have a similar enough structure and usage pattern that one database could add a set of tables from another database in order to incorporate a subsystem from the other ATMS. In fact, an initial effort could be to simply include all tables from one data model into the other data model. The table owner (e.g., SunGuide software and Lonestar) would be used to distinguish between the two table sets. Over time, selective subsystem tables could be selected for merging, rather than taking the approach of merging all the subsystem tables at once.

Where tables from the two ATMSs differ, the business logic that uses the tables, combined with the data model design, would have to be examined in concert to determine the most effective method for combining subsystem tables. Many times, business logic drives the data model design in order to accommodate certain functionality required by the system. Since FDOT and TxDOT have different philosophies and polices, examination of the business logic is required to ensure that the merged data model design could accommodate both ATMS's merged subsystems. As a starting point, the tables within a subsystem group should stay together with the baseline of the subsystem business logic and then additional tables could be added or modifications to the tables could be made in order to accomplish a merged set of features.

5.3 Data Distribution

SunGuide software and Lonestar have very similar data distribution modules named Databus and CSD, respectively. While the differences will be extremely minor, any modifications could have system-wide impacts and each difference should be investigated with this in mind.

5.4 UI

The SunGuide software UI will be going through a major overhaul. One of the things we will be discussing is how FDOT can move forward with their UI in a fashion that will ultimately include compatibility for supporting the TxDOT needs. More information is provided in Appendix B.

5.5 User Security

SunGuide software will be incorporating the TxDOT Software Administration Application and adding an additional enhancement to it in order to provide either read-only support to all devices and custom permissions to a subset of devices. This should be easily transferrable back to TxDOT for their current usage and even for their benefit from the enhancement.

5.6 Status Logger and Executive Handler

Status logger and executive handler are still very similar between SunGuide software and Lonestar. They serve the same function and may have had slight modifications since being maintained independently, but should be fairly easy to resolve these differences due to the same operational use and similar implementation.

5.7 DMSs

SunGuide software and Lonestar both have DMS subsystems. TxDOT developed the DMS subsystem initially and FDOT modified it to meet their needs in SunGuide software in the initial release. Some enhancements have been shared since the initial development, while both systems have had their own GUIs implemented. SunGuide software will further enhance their DMS subsystem to handle NTCIP version 2 with full color messaging which is a rather large effort. As a part of this effort, all DMS related GUI components will be updated to use the latest Microsoft Windows Presentation Foundation. There are a few features that Lonestar has that can be merged with SunGuide software at a minor effort. Table 5.7 shows the list of DMS features supported by each system.

Table 5.7 – DMS Features

DMS Feature	SunGuide Software	Lonestar	Resolution
Windows Presentation Foundation based GUI	In development; to be released in 2013		FDOT and TxDOT will adopt
Icons on map representing sign's location	X	X	
Icon/List colors representing sign status:	X	X	
• Operational status (no message)	X	X	
• Operational status (with message)	Small effort to add	X	FDOT will adopt
• Error status	X	X	
• Operational status with Drum (alternate route) message (or trailblazer)	X	X	
• Out of service	X	X	
• Processing (pending) status	X	X	
All DMS view		X	FDOT will adopt
Message queue	X	X	
Icon right-click access to sign and queue status	X	X	
Multi-select icons on map		X	TxDOT will adopt
Send message to sign	X	X	
Send message to group	X	X	
Font selection	Configurable; medium effort to accommodate	Fonts are automatically selected based on font priority or operator selection during send message or create operations	FDOT will adopt
Multiple phase messages	Dynamically add additional phases	Operationally, 1 or 2 phases are used. More can be used if users type in the MULTI mode.	TxDOT will adopt and not use operationally
Phase parameters (on/off durations,	X	X	
Flashing beacons	Only used with VSL, not by user. Small effort to add	X	FDOT will adopt
Message duration / indefinite	X	X	

DMS Feature	SunGuide Software	Lonestar	Resolution
Message priority	255 levels Small effort to add configurable GUI selection to match TRF recommendation	255 levels configurable, (10 levels recommended by TRF)	
DMS detailed diagnostics	X	X	
Search message library	X	X	
Multi-text editor	X	X	
Drum/trailblazer re-routing DMS	X	X	
Drum messaging does not display the queue		X	FDOT will adopt and not use
Default message displayed when user message terminated	Can be accomplished with priorities; minor effort to add	Only if you are running without MQA, which is not normal operations.	FDOT will adopt
Blank current message without blanking queue – queue is paused		X	FDOT will adopt – may need UI controls to support and preserve existing behavior
Message library	X	X	
Limited remote message status and control	X	X	
Full remote message status and control via RCA	Planned for DMS	X	FDOT will adopt
Display signs in tree hierarchy under owning agency	Small effort	X	FDOT will adopt
Disabling remote center signs	Small effort	X	FDOT will adopt
DMS detailed components status	X	X	
DMS operational/control mode	X	X	
DMS brightness mode setting	X	X	
Force poll of sign status	X	X	
Synchronize clocks	X	X	
Reset controller	X	X	
DMS Lamp Status	X	X	
DMS Pixel Status	X	X	
DMS Groups	X	X	
DMS brief status dialog	X		
DMS temperature	X	X	
Find on map	X	In development	TxDOT will adopt

DMS Feature	SunGuide Software	Lonestar	Resolution
Filter DMS by various categories	X	X	
DMS automatic polling	X	X	
Display last polling communication status	X	X	
Change sign status to blank even if terminate sign message fails	X		TxDOT will adopt
DMS power	X	X	
DMS approved words	X	X	
Override unapproved words system-wide configuration setting	X		TxDOT will adopt
Full Color DMS	In development		TxDOT will adopt
NTCIP v2	In development	In development	FDOT and TxDOT will adopt
Support for SNMP over TCP/IP		In development	FDOT will adopt
Image library management	In development		TxDOT will adopt
Message Scheduling	Development to move from DMS to SAS in Microsoft Windows Presentation Framework with a Microsoft Outlook style scheduling GUI	Uses MSA	TxDOT will adopt SAS GUI

5.8 Message Arbitration and Queuing

SunGuide software and Lonestar both have message arbitration and queuing subsystems. The Lonestar subsystem is called MQA, while the SunGuide software subsystem is called MAS. Each system implements its own GUIs. Table 5.8 shows the list of MAS/MQA features supported by each system.

Table 5.8 – MAS/MQA Features

MAS/MQA Feature	SunGuide Software	Lonestar	Resolution
Windows Presentation Foundation based GUI	In development; to be released in 2013		TxDOT will adopt
Handles DMS messages	X	X	
Handles HAR messages	X		TxDOT will adopt
Highest priority processed first	X	X	
Equal priority processing	First in, first out	Last in, first out	FDOT will adopt
Priority numbering	1 is highest	0 is the lowest	FDOT will adopt
Queue Manager dialog	X	X	
Resend message on status change		X	FDOT will adopt
Terminate message		X	FDOT will adopt
Priority editing	X	X	

5.9 CCTV

SunGuide software and Lonestar both have CCTV subsystems; in Lonestar it is called the Camera Manager, and each with its own set of associated GUIs. Table 5.9 shows the list of CCTV features supported by each subsystem.

Table 5.9 – CCTV Features

CCTV Feature	SunGuide Software	Lonestar
Icon on map represents camera location.	X	X
Color of icon represents camera status.	X	X
Set operational status	X	X
Find on map	X	
Full-motion Pan/Tilt/Zoom (PTZ) control via GUI	X	X
Nudge PTZ control via GUI	X	X
Camera control via MCP	X	X
Camera control via universal serial bus joystick	X	Deprecated
Focus and Iris control	X	X
Camera locking	X	X – multiple levels based on user priority
Numbered presets	1-99	1-99
Named presets	X	X
Advanced control (NTCIP 1205)	X	
Camera groups	X	X
Manage titlers		X – part of CCTV Master
Configure snapshots		X – part of CCTV Master

5.10 Traffic Detection

SunGuide software and Lonestar both have traffic detection subsystems, in each case called the TSS. Each subsystem implements its own GUIs. Table 5.10 shows the list of traffic detection features supported by each subsystem.

Table 5.10 – Traffic Detection Features

Traffic Detection Feature	SunGuide Software	Lonestar
Icon on map represents detector location	X	X
Color of icon represents detector status	X	X
Set operational status	X	X
Find on map	X	
Click icon to manage detector	X	X
Click link to view link	X	X
Display current speed, volume, and occupancy for a specified link	X	X
Threshold indicators	X	X
Display rolling average of speed, volume, and occupancy for a specified link	X	X
Display current speed and link traversal time for probe-based detectors	X	
Display rolling average of speed and link traversal time for probe-based detectors	X	
Display downstream detector for probe-based detectors	X	
Display vehicle count and link delay for probe-based detectors	X	
Enable/disable dynamic linking of probe-based detectors	X	
Filter detector list by roadway	X	X
Filter detector list by protocol		X
Filter detector list by status	X	X
Alarms based on traffic conditions	X	X
Synchronize clock	X	

5.11 Travel Time

SunGuide software and Lonestar both have travel time subsystems. The Lonestar subsystem is called the Travel Time Application (TTA), while the SunGuide software subsystem is called Travel Time Subsystem. Each system implements its own GUIs. Table 5.11 shows the list of travel time features supported by each system.

Table 5.11 – Travel Time Features

Travel Time Feature	SunGuide Software	Lonestar
View current travel times	X	X
View difference between current travel time and free-flow travel time	X	
Enable/disable calculation of travel time on a per-link basis	X	
View matching routes for a specified link	X	
View alternate routes for a specified link	X	
View TSS links used to calculated travel time	X	
View DMSs that display travel time of a specified link	X	
View travel time messages currently displayed on DMSs	X	X

5.12 Event Management

SunGuide software and Lonestar both have event management subsystems. While they are very similar in concept, there are many differences in implementation. Tables 5-12 and 5-13 show some of the major and minor feature differences, primarily by which features and data fields are present in SunGuide software and Lonestar, respectively.

Table 5.12 – Event Management Feature Comparison

Event Management Feature	SunGuide Software	Lonestar
Separate Planned / Unplanned Event Management screens		X
Full Audit and change tracking	X	
Start and End location for blockage		X
Full location pre-configuration	X	X
drag snapshots from CCTV		X
Responder dispatch	X	X
Responder timestamps	X	X
Event still active reminder		X
Full Incident Clearance performance measures reporting	X	

Table 5.13 – Event Management Data Field Differences Comparison

Event Management Data Field	SunGuide Software	Lonestar
Event ID	X	X
Event Description	X	X
Primary Involved Vehicle (color, make, model, year)	X	
Event Type	X	X
Blockage	X	X
Event Status with timestamp	X	X
Private flag		X
Event Severity	X	X
511 published severity	X	
Scene cleared timestamp	X	X
Delay cleared timestamp		X
Lat/long, location description, Roadway, direction, crossstreet, mile marker	X	X
Landmark near location		X
Lane map editable in event	X	X
Express Lane	X	
Entrance Exit Ramp		X
Gore	X	
Lane status: cleared, blocked, unknown	X	X
Nearest CCTV field and launch button	X	X
Event comments	X	X
Private comments		X
Catagorizable comments	X	
Parties notified		X
Notes		X
Congestion head and tail	X	

5.13 Scheduled Action

SunGuide software and Lonestar both have scheduled action subsystems, although their purposes are very different. The Lonestar subsystem is called the MSA; it allows messages to be placed on DMSs according to defined schedules. The SunGuide software subsystem is called the Scheduled Actions Subsystem and allows cameras to be controlled according to defined sequences of motion, and for these sequences to be run according to defined schedules.

Table 5.14 shows the list of scheduled action features supported by each system.

Table 5.14 – Scheduled Action Features

Scheduled Action Feature	SunGuide Software	Lonestar
Manage schedules	X	X
Add actions to schedules	Camera sequences (TVT enabling and DMS messaging in development)	DMS messages, EM response plans
Activate and suspend schedules	X (deprecated, to be removed in next release)	X
Activate sequence on specified camera	X (deprecated, to be removed in next release)	

5.14 Video Switching

SunGuide software and Lonestar both have video switching functionality. In SunGuide software, this functionality is part of the Video Switching subsystem, while in Lonestar it's part of the CCTV Master subsystem. Table 5.15 shows the list of video switching features supported by each system.

Table 5.15 – Video Switching Features

Video Switching Feature	SunGuide Software	Lonestar
Video tours	X	X
Drag and drop from video source to video destination	X – click on a list and then click on a destination, not true drag and drop	X
Camera blocking/blackout	Video cannot be switched to a restricted external destination; snapshots will not be sent via C2C to Florida's advances traveler information system.	Video can only be switched to secure destinations. Blacking out also removes snapshots.
Snapshot preview		X
Snapshot auto or manual refresh		X
Save snapshots		X

5.15 RWIS

SunGuide software and Lonestar both have RWIS subsystems. The Lonestar subsystem is called the Environmental Sensor Station Lite, while the SunGuide software subsystem is called RWIS. Table 5.16 shows the list of RWIS features supported by each system.

Table 5.16 – RWIS Features

RWIS Features	SunGuide Software	Lonestar
Icon on map represents RWIS location	X	
Color of icon represents RWIS status	X	
Set operational status	X	
Find on map	X	
Click icon to manage RWIS	X	
Filter by status and location	X	
View summary of weather data	X	
View atmospheric detail	X	
View pavement detail	X	
View precipitation detail	X	
View temperature detail	X	
View wind detail	X	
Retrieve device status		X
Retrieve water depth on roadway		X

5.16 Notify Service

SunGuide software and Lonestar both have notification subsystems. The Lonestar subsystem is called the Contact Notification Application, while the SunGuide software subsystem is called the Notify Manager. Table 5.17 shows the list of notification features supported by each system.

Table 5.17 – Notification Features

Notification Feature	SunGuide Software	Lonestar
Sends notifications via SMTP	X	X
Sends notificaitons via GUI popups		X
Notifies based on Warning and Error conditions sent to the Executive Handler by other subsystems	X	
Send a manual notification		X
Manage contacts		X
Manage groups		X
Specify notification method		X

6 Recommendations

Based on the information collected and analysis in phase One, the following recommendations can be made for several of the aspects of ATMS programs. Phase Two will further define these recommendations after receiving feedback from some of the stakeholders.

6.1 Information Inventory

A good plan for consolidating the software products can begin development with the information already collected. However, in order to further reduce the unknowns and quantify the effort, some additional information would be helpful. The references in section 1.4 on page 2 identify information recommended for collection and the status of that information for both systems.

6.2 Program Collaboration

After an initial review of the program characteristics and the high-level software systems, a recommended approach for FDOT and TXDOT to combine their software to support the collaboration of a single platform for development and implementation is described in the following sections.

6.2.1 Documentation

SunGuide software and Lonestar both have comprehensive sets of documentation. Documents that are technical in nature and applicable to the software should be merged together. Some planning documents that exist in the SunGuide software and are relevant to the combined software could be reviewed by TxDOT and possibly modified to meet their needs as a single, shared set of plans. The task of updating documents should be merged, reviewed, and modified in a manner consistent with the systems engineering process.

6.2.2 Operations Accommodations and Adjustment

Every TMC operates differently. There are TMC operational differences between local, regional, and satellite or backup TMCs, and there are differences in operations between TMCs in different states. In the component-level comparison and analysis phase of this effort, these differences will be identified and analyzed. Each difference will either be accommodated by the software through configuration options or other modifications, or the differences can be eliminated by a change in the TMC operations. In general, changes to the software to accommodate operational differences will incur cost to the software project, but will require less coordination and compromise for TMC users as their specific needs will be met without changing their operations. However, there will be benefit to consolidating and standardizing some operational characteristics of the software that could outweigh the coordination effort and operational change for the TMC. One example may be how SunGuide software will not allow an event to be closed unless all service patrols on scene have departed; furthermore, SunGuide software will not allow a service patrol to depart until at least one activity has been entered for that service patrol – even if the activity is a

'void.' This restriction has been helpful for FDOT, but TxDOT would need to consider if they had an operational need to remove the restriction, or if they wanted to consider accommodating that restriction by ensuring activities were added and departures were logged for all service patrols on scene at the incident. All operational differences will need to be considered carefully by all stakeholders as a very important early stage of the shared software development process.

6.2.3 System Development and Maintenance Planning

There are current development and maintenance activities already under contract and in execution separately for both SunGuide software and Lonestar. These programs should be continued so that the needs being met by each legacy system are supported while a new joint effort to combine the software systems is executed in parallel. This effort should use a similar approach as defined in the software development plan that has been updated, reviewed, and approved by both DOTs as a joint, shared effort. Future enhancements and maintenance will require a more extensive configuration management process before being developed, but the resulting developed product will be a benefit to both DOTs.

6.2.4 Configuration Management

6.2.4.1 Change Management

Each DOT is responsible for meeting the needs of their own users. The DOTs will also have to work together to implement change in a way that is most efficient and allows cost beneficial sharing of the code base. Any change will potentially impose impact to the shared code base and its usability to both DOTs. The DOTs will go through a process of coordination with their own users. The DOTs will determine how their needs can be balanced with any differences in the needs of the other DOT. The more operational differences that can be minimized will provide more consistency and lower software development and support costs, but may require users to accommodate limitations and less flexibility in the software. The remaining operational differences will have to be accommodated through software configuration options, which provides less consistency and higher development and support costs.

The change management process at each DOT should not need to be substantially altered. The members of the change management board should be informed of what the other state is doing with the rest of the combined software code base.

Once the operational requirements from both DOTs have been settled, an identical, combined set of operational concepts and configuration options will be approved by each DOT for implementation.

6.2.4.2 Protocol Support

The combined software system will initially benefit by supporting ITS devices and protocols that are already supported by either system. As new devices and protocols become available, drivers

may be built or enhanced to add support for new devices and protocols. The development could be done by either DOT or by their vendor(s). Both DOTs will then go through their change management and testing processes to incorporate the change in to the system. The result will be a growing list of devices and protocols supported by the combined system that will be maintained, updated, and published by both DOTs.

6.2.4.3 Product Branding

Both SunGuide software and Lonestar have built an asset in their respective brands. This branding should be preserved by a minimal change in a configuration parameter prior to building the product release distribution package. The simpler the configuration and the smaller the quantity of duplicated items to support the additional branding, the less effort it will be to ensure a consistent delivery of each brand of the product to both DOTs.

6.2.4.4 Version Control

The shared code base will be versioned in such a way as to track all releases outside of the organization maintaining the code base, including non-deployed testing versions packaged for pre-testing activities. While both DOTs may brand, package, and distribute their product differently, they will participate in the same versioning control. Each DOT takes great pride in their ATMS asset and service to their users, and has built trust and recognition in their brand names of the ATMS software system. When packaging a release, the product will be branded with their organization's ATMS brand name.

6.2.4.5 Source Code Management

Both agencies will need access to the shared source code database to make modifications. However, a single organization will need to be tasked and responsible for maintaining the integrated source code database. With either DOT's approval and both DOT's awareness, the code may be checked out to approved organizations, agencies, or vendors who may modify and use the software. Any modifications made to the software would then be dedicated back to the DOTs. Any approved and tested modifications would then be reintegrated back into the combined software code base.

6.2.5 Testing and Acceptance

Both agencies will be invited to participate and witness FATs. They will have an opportunity to review and comment on FAT plans, procedures, and test results. With both DOTs participating, a single FAT will be sufficient for each release.

Each DOT will be responsible for conducting independent testing for their own software product releases and needs; however, they could coordinate together for a single independent testing that meets both of their requirements by reviewing and approving the same independent testing plan, procedures, and resulting report as well as attending the same independent testing event.

It would be recommended to include some automated testing capability to reduce the burden of regression testing modifications for multiple configurations used by both DOTs with minimal effort.

6.2.6 Support, Maintenance, and Training

Deployment specific support and maintenance does not involve modifying any of the software concepts, design, or requirements. Users have specific needs that when met, do not necessarily benefit other users. These activities include deployment, training events, configuration, and other user requests. Other maintenance involves removing defects by making modifications to the detailed design and implementation of the code. This does not alter requirements and thus does not require formal change management approval. However, this activity is prioritized by the sponsor of the support contract based on the severity and locality of the impact to the sponsor's users. Having separate contracts for support and maintenance of the software would allow both DOTs to only be responsible for funding support and maintenance for their own users according to their own needs and priority. However, a central contract would allow the best coordination of support as more agencies and potentially other states become users of the software.

Defect removal modifications to the software will be shared back to the combined system, which is mutually beneficial to all users. Thus, even if defect removal is done through multiple contracts, there should be sufficient coordination or at least information sharing of the issues and modifications available so that once the issue is resolved other users are aware and can benefit from the solution.

6.3 Architectural Platform Design

The current two software architecture designs are similar, which should be preserved and reused to reduce rework and to facilitate the migration. The new architecture will be defined as a layered platform. Each layer will provide a service or set of resources to the components in the higher layer.

6.3.1 Data Definition Layer

The data definitions layer will reside at the bottom of the architecture and will include a shared set of data definitions required to be consistent across all components. There are three types of data definitions for data in the system: persistent storage as a database record, non-persistent storage suitable for processing as an object in memory, and non-persistent storage suitable for transmission as an XML element. The XML schema defines the format for XML elements. The DataLibrary is responsible for defining the objects in memory. The database model is responsible for defining the database and the structure of each type of record in the database.

The DataLibrary also defines functions for converting these objects in memory to XML elements and back into objects in memory. This process is called serialization and de-serialization, respectively. DataLibrary will expose high-level functions to all other modules for serialization

and de-serialization of data objects. The XML schemas are used during software development of the DataLibrary and are also needed by the DataLibrary for validation of XML objects during execution of the software. In contrast, the XML schema is a component of the ICD and will serve as the authority for the XML representation of the data elements, while the DataLibrary is a software implementation that implements the data object definitions, implements serialization and de-serialization functions that comply with the XML schema, and is used by the other software modules as a software library.

The database will be designed and defined in a database model. The subsystems will follow the definitions in the database model for interaction with the database.

6.3.2 Architecture Layer

The second layer from the data definition layer is the architecture layer. It includes the StatusLoggerClientLib, the ExecutiveHandlerClientLib, and the SharedArchitectureLibrary.

The StatusLoggerClientLib provides modules with the ability to log messages in the StatusLoggerService. The ExecutiveHandlerClient allows module processes to interface with the ExecutiveHandlerServer, which provides access to start, stop, set logging detail level, and receive operational status of each module. The ArchitectureLibrary, also known as ITSGenericLibrary by SunGuide software and Lonestar, defines a set of architectural patterns that are common to all modules.

The library provides a platform for components to easily and consistently instantiate software operational processes common to many software system modules such as: managing connections to a database, communicating to an external device, providing a server interface to queue, and processing incoming messages or transactions, or providing a client interface to connect to a server interface. These unit level design patterns as well as a style guide should be well documented for consistency in source code development.

6.3.3 Shared Services Layer

The third layer above the SharedArchitectureLibrary is the shared services layer. This layer contains executing modules, not libraries, that provide system-wide services to many other executing modules. The transaction hub (formerly databus or CSD), the status logger, and the executive handler server are included in this layer.

6.3.4 Business Logic Layer

The fourth layer, above the shared services layer, is the business logic layer. This layer includes all of modules that implement a related set of business logic. This layer includes a core set of modules that only interface with other modules in the ATMS software architecture, while there is a periphery set of modules that contain interfaces to external users and systems, including drivers, UIs, and C2C plugins.

Outside of the architectural stack are other systems that are not fully integrated into the ATMS software architecture, but may interface with it. ITS devices, third-party data feeds, C2C infrastructure, and 511 systems are some examples.

6.4 Engineering Approach

Supporting differing operations, legacy code bases, and migrating to a shared code base can be a high-cost, high-risk project. Breaking this up into manageable chunks will allow this work to be performed over time, and field-tested by fewer components initially and at any one time. The layered description of the architecture will help organize the approach to building the combined ATMS software.

The general approach will be to import shared modules into the combined software architecture one layer at time from the bottom up, such that they support both systems initially. Then iteratively merge all business layer components.

The data definitions layer will start as a superset of the corresponding definitions from both SunGuide software and Lonestar, with any conflicts resolved by providing the most detail and features that are needed by higher layer components. There will be iterative changes and fine tuning to this layer to merge any redundant definitions.

The shared architecture layer will be imported next. This should be fairly straightforward as the Status Logger Client and Executive Handler Client are already nearly identical, and the SharedArchitectureLibary's role is to implement architectural functionality for two software systems that already share the same conceptual architecture and protocol.

The shared services layer will be imported after the architecture layer is initially in place. The ExecutiveHandlerServer and the StatusLoggerServer should already be identical between SunGuide software and Lonestar, but databus and CSD may have a few differences to reconcile. The new sharedTransactionHub will be modified to provide support that covers features from both databus and CSD that will be needed in the combined software system.

The business logic layer will be imported next, and will be the bulk of the work. Modules will be selected for import based on the least dependencies, starting with core modules. For example, DMS and the TSS will be imported prior to travel times . Drivers will be imported immediately after importing the subsystems they support. Modules that have a SunGuide software and a Lonestar equivalent will be evaluated to determine which legacy system's implementation would be the least amount of work to import and add support for features required for both systems. The System Authentication Administration subsystem would be a great first module to merge. It is used by all other components and could almost be considered a shared service even through it fits in the architecture as a data provider or subsystem.

The UI, one of the most important of the periphery modules in the business logic layer, will undergo a complete overhaul. FDOT is already beginning general discussion of some technologies and consistent design and operational characteristics that the GUI will implement; such as docking panes, tabbed window pages, context menu controls, and other features available in other modern UIs. This UI merge effort will require the most coordination between the DOTs to jointly specify UI requirements since this has the most impact to the users as the features and operation of the system is most apparent in the UI.

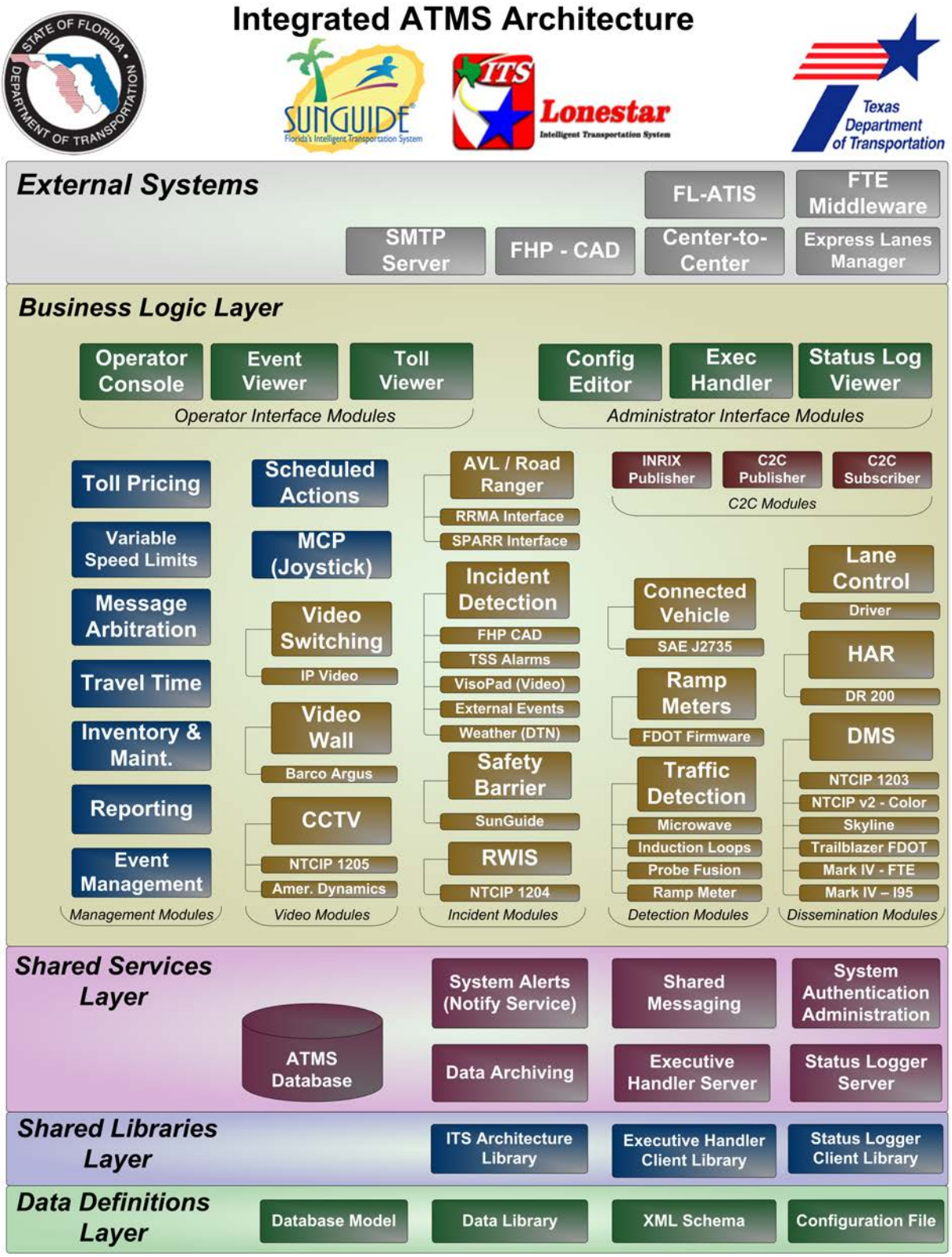

Figure 6.1: Integrated ATMS Architecture

7 Risks

7.1 Architectural Differences

The highest impact differences will exist in differences in the architecture, or architecture-related implementation that affect all modules. In the new architecture concept described in the previous sections, this would be defined in the bottom-most layers of the architecture – the data definitions layer and the architecture layer, respectively. One of the primary tasks of the second phase of this review is to address this risk, quantify these architectural differences, and refine the software consolidation approach to try to mitigate any software development impacts.

7.2 Device Functionality Differences

There may be differences not just in the protocol used by a device, but how a device is used. For example, a DMS can accept sign messages defined in the command sent from the ATMS software driver to the DMS sign, or the DMS can be commanded to display a message previously stored on the sign's internal, persistent storage. Another possible difference is if a different type of VSL sign is used that requires a speed limit schedule, not just the current speed limit, to be sent to the VSL sign. If these or other differences in how devices are used exist, there could be impacts throughout the system and all the way up to the UI. These differences will be evaluated in more detail in Phase Two of this analysis effort. Any differences between the current software systems that have a large impact throughout the rest of the design of the combined system need to be carefully considered. These differences could either be eliminated by deciding on a single design approach or accommodated by having a configuration option to allow multiple design approaches. The costs will be weighed against the benefits of each option and both DOTs will need to agree on a final decision.

7.3 UI Operational Differences

There may be differences in the UI as this is the most visible component to the users and represents how they access the system to conduct their operations. Differences in the UI may mean differences in how the TMCs operate. Adding features and support for additional devices and protocols does not have a negative impact on users as they may not use the features and new features do not alter how they use the software. However, changing the UI does have a great impact on users and to the combined software system being built. The UI may be a place where more configurable options are required while the underlying support for unused options does not impose a problem. The UI will go through a design review process with both DOTs and will include features and configuration options to meet the needs of all users.

7.4 Database Model Differences

Generally, merging of the two data models will not present serious risk issues. If the decision to merge common tables is made, the biggest risks will be ensuring the columns are of the right type and size, the primary and foreign key relationships stay intact, and views are modified

accordingly to ensure the proper data is selected. The larger risk will be to the business logic of the two ATMSs. If tables are combined between the two data models, the impact to the business logic for one or both ATMSs will have to be examined. One opportunity to reduce these risks is to start by combining both data models without changing any tables. This would allow each ATMS to utilize its own tables. After combining of the data models in this fashion, selected subsystems could be analyzed and merged as appropriate. One such example that is already approved is the user permissions. FDOT has approved modifying SunGuide software to use the Lonestar user permission schemes.

7.5 System Support

Providing support to system users across multiple states and support contracts will require increased coordination. Currently, a model of this issue exists where defects are identified and removed in a specific TMC, but not removed in all TMCs throughout a state until the next software release. Sometimes, the modification to the software may be closer to a change in requirements or design, or even just the look and feel rather than the removal of a defect to better implement the existing requirements and design. With multiple DOTs impacted by modifications, care must be made to coordinate all modifications to ensure that neither DOT is negatively impacted by any changes. Also, just as in the configuration management phase, it is important to promote high quality solutions that work well for everyone and to avoid overly excessive configuration of multiple solutions to each problem. Thus, support should be limited to defect removal, training, and deployment issues and defer other modifications to the configuration management process.

7.6 Funding Management

As the software proves versatile and can support TMCs throughout two different states, there is a potential for other agencies and states to desire to use the software. These new users may have funding for the software program, but without good coordination and configuration management, the software could again split into two separate products, defeating the overall cost savings and quality achieved by the combined system. An appropriate funding model should be offered to additional agencies desiring to use the integrated ATMS software so they get the support and enhancements they need while participating in the software program that benefits all users. This will avoid the need to duplicate effort when sharing modifications between agencies since all agencies would be funding and using the same code base.

7.7 Third Party Development

As more vendors desire to market ITS products to be included in the integrated ATMS software, drivers will have to be built to support their products. Having vendors develop their own drivers will reduce the cost burden on the agencies, but exposes a risk of a driver that is not built in a manner consistent with the architecture and coding style of the rest of the system. Providing the libraries, coding style guide, and example code will greatly reduce the risk. Code inspections prior to acceptance will also reduce this risk for the final product accepted. Full ownership of the

source code to be assimilated into the rest of the integrated ATMS code base is also important so that any modifications can be made and distribution of the product integrated into the ATMS is not impeded.

8 Next Steps

1. Discuss Event Management operational differences with TxDOT and FDOT management.

2. Discuss GUI overhaul – major effort and the place where changes to existing functionality will have the most impact to operations.

3. Determine roadmap, cost, and schedule for implementation milestones:

    a. Discuss risks;

    b. SwRI to review document, provide comments, and further discuss the roadmap; and

    c. Request cost and schedule proposal.

4. Determine program collaboration of stakeholder support, funding, and change management.

5. Build it!

Appendix A:

C2C Infrastructure TxDOT/FDOT Differences

Center-To-Center Infrastructure TxDOT/FDOT Differences

Table of Contents

1.0 Document Structure

A comparison was performed between the Center-To-Center Infrastructure (C2CI) components released with TxDOT's Lonestar™ 4.0 and FDOT's SunGuide® 5.1. This comparison focused on two areas:

1. Functional – differences that cause the two variations of C2CI to behave differently
2. Implementation – differences in how a behavior was actually implemented. These differences typically result in the same behavior, but with additional error handling and/or a cleaner implementation.

Changes were noted due to modifications made by both TxDOT and FDOT to either attempt to solve a problem or to add new functionality. These changes are described below. There are many smaller incidental differences that are not noted in this document.

2.0 Differences

2.1 Functional Differences

2.1.1 Schema Differences

Schema differences are not an actual difference in code implementation, but a difference in the defined ICD. Due to the generic nature of C2CI, adding a new command or status type does not require a corresponding change to the actual C2CI code implementation.

2.1.1.1 Common

The following changes to common C2CI schemas were found:

3. FDOT, in addition to the common Directions (North, South, etc), allows for a value of "BothDirections"
4. FDOT contains an additional "county" field in the definition for equipment locations
5. FDOT defines types for: event severity, offsetType, atisSeverityType

2.1.1.2 Commands

The following commands exist on TxDOT, but not on FDOT:

1. Various status requests (LCS, ramp meter, traffic signal, HOV, school zone, reversible lanes, dynamic lanes)
2. Device Plan Status Requests (ramp meter, traffic signal, HOV, school zone, reversible lane, dynamic lane, device group)
3. Device Timeframe requests
4. LCS signal command
5. Device Plan commands (same list as item 2).
6. Queue Message command

The following commands exist on FDOT, but not on TxDOT:

1. Floodgate command
2. Event Command

2.1.1.3 Status Data

The following status types exist on FDOT, but not on TxDOT:

1. event
2. extEvent
3. floodgate
4. locale
5. trafficSpeed

The following status types exist on TxDOT, but not on FDOT:

1. busLocation
2. busStop
3. closure
4. dynamicLane
5. emergency
6. hov
7. incident
8. lcs
9. parkAndRide
10. parkingLot
11. railLocation
12. railroad
13. railStop
14. rampMeter
15. reversibleLane
16. schoolZone
17. specialEvent
18. trafficSignal
19. vehiclePriority

The following status types exist in both systems, but have differences:

1. FDOT changes
    a. cctvStatus –added a “canPublish” flag
    b. dmsStatus –added a “duration” field and “canPublish” flag
    c. har –added a “duration” field
    d. network – added a “canPublish” flag and a “county” field for links
    e. tcd – added a trafficCondData flag for “canPublish”
    f. tvtLink –added a “centerId” attribute to each TSS link ID which makes up a TVT link; also added a “canPublish” flag

   g. tvtStatus –added a “canPublish” flag;
2. TxDOT changes
   a. ess – added a capability and alarm type attribute for water depth; also includes waterDepth and roadwayHeight fields
   b. tvtStatus—added an “avgSpeed” flag

2.1.2 Code/Behavior Differences

The following section describes actual behavioral differences between the two C2CI variations.

2.1.2.1 General

1. TxDOT – Queue sizes / counts are logged every 5 minutes
2. TxDOT – Defines Web method QueuedSubscribeNets in addition to the normal SubscribeNets
   a. Instead of returning initial status information as a response, it is provided separate from the act of subscribing as a status update
3. TxDOT – MaxMessageSize configuration value – Breaks up or combines messages based on this value. Very large messages are broken apart into a series of smaller messages when possible. Small updates received at the same time are combined into a single message.
4. TxDOT – BacklogAge – handles unresponsive subscribes so memory does not grow and cause of of memory exception
5. TxDOT – CondenserQueue – Improved management of the background worker thread to prevent an exception from causing updates to stop processing
6. TxDOT – Handles backwards compatibility to 3.x versions of C2CI

2.1.2.2 Extractor

1. TxDOT – Added error/exception handling to the CancelSubscriptions method
2. Login – Differences in setup of socket to C2C plugin (buffer size and timeout)
3. TxDOT – Added error/exception to Logout method
4. TxDOT – RequestNets / RequestNetsBlocked – Added locking of sessions collection

2.2 Implementation Differences

2.2.1 General

1. Keep Alives are implemented differently. TxDOT implementation appears cleaner and occurs on regular intervals
2. Parsing of network configuration data (“serverList”) differs
3. Logging – TxDOT fills in basic information for each log message. Both TxDOT and FDOT should probably update to use same Status Logger implementation used by other C# subsystems
4. ConnectToServer – Behavior of one C2C component connecting and subscribing to another is implemented very differently.

2.2.2 Extractor

1. Sessions are managed differently
2. FDOT – If an exception occurs during a Keep-Alive, that server is logged out

3.0 Consolidating

Combining the two versions into one version requires changes to the schemas and the code. Schema modifications would involve taking additions from both sides and would not affect the implementation of the system. From the review, the TxDOT C2C version appears to have cleaner implementations where differences exist in **how** a feature is implemented. To consolidate into one version, the TxDOT version would be used as the basis and FDOT modifications examined for inclusion. These changes would require approximately 6-8 weeks of coding and testing internally. Following the initial testing, FDOT and TxDOT data feeds would need to be available for testing compatibility.

Appendix B:

Lonestar and SunGuide Software Unification Recommendations

Lonestar and SunGuide Unification Recommendations

Merging Lonestar and SunGuide into a single, unified ATMS is an effort which would, in the long run, result in cost savings and additional software features for both Florida and Texas. These benefits could become even greater as additional states use the common software, provided all entities return contributions to the overall system and contributions are developed in a compatible manner. Reaching that goal will require short-term investment in each system as individual modules are merged, modified and deployed across the states. Additionally, each state must be willing to coordinate and compromise in their system operations to minimize the customization of the software required for any single deployment.

Phased Integration

While both ATMSs are built of several layers, attempting to merge the entire system one layer at a time would be a sizeable undertaking providing little interim, visible benefit to any stakeholder. User interfaces are designed based on the schemas presented by subsystems; schemas and database tables are driven by the functionality needed of subsystems; subsystem functionality is determined by the operational requirements of the states, districts, and users. Doing a simple merge of databases by including all tables used by both systems would result in a more cluttered, difficult to understand database. Attempting to write user interfaces which can communicate to different versions of subsystems which perform similar but not identical actions would effectively require continuing the redundant efforts this unification is attempting to avoid.

A more effective and cost-efficient approach would instead merge individual functional areas from top to bottom, evaluate the process and results of that effort, then repeat the process as necessary until all desired subsystems have been merged. This effort would include analyzing both states' needs and current functionality for a subsystem, the appropriate set of merged schemas and database tables, and a unified user interface for that functionality that would meet all stakeholder needs. With that information in hand, appropriate software updates could be performed, tested, and deployed without duplication of effort. Using this approach, stakeholders would more quickly see benefits as each area was completed. In addition, the efforts would be more efficient because the focus for the stakeholders is centered on an area of functionality.

Typical Integration Phase Steps

During the integration of a particular functional area, several common steps would be expected. These steps could be tailored in some instances, but skipping steps should be minimized to prevent conflicts occurring between planned and actual outcomes.

1. **Identification of Subsystems to Integrate**
   When integrating a subsystem, other components which interact with it should be examined. There may be sufficient dependencies between that subsystem and another that it would be inefficient to migrate one without the other. For instance, there are a number of interactions between DMS and MAS/MQA, and between TSS and TTA/TVT. For this reason, there will be cases when it will be more appropriate and cost effective to consider merging blocks of subsystems at a time.
2. **Analysis of Software Requirements**

Both states have detailed existing software requirements documented. Requirements which relate to the subsystems being integrated should be identified and extracted to assist with the comparison of the systems.

3. **Analysis of Functional Differences**
   Beyond the requirements, both systems also include a number of features and behaviors which may not be fully documented. The actual operation of the systems in the field should be observed and evaluated, along with the full extent of functionality provided by the user interfaces of the systems. This feature list will also assist with the comparison and should be added to the requirement sets.
4. **Interaction Identification**
   In addition to the behavior of the subsystem(s) being merged, other system components will likely interact with the subsystem(s). These interactions must be identified in each system to document additional work that will be required to integrate a merged module back into the main system. As more modules are migrated, this effort will lessen, as less unique code will remain in each system.
5. **Identification of Unique and Contradictory Behaviors**
   With a list of requirements and functional behaviors, both systems should be compared to identify areas where one system provides functionality not present in the other, and where each system has a different approach to the same problem. These unique and contradictory features should be itemized for review by the states' CMBs.
6. **Resolution of Differences**
   Where behaviors or requirements of the systems differ, each state's CMB should evaluate the differences and prioritize the importance of distinct behaviors to their operations. For conflicting behaviors, if one state prioritizes the behavior as important, but the other does not, deference should generally be given to the state which feels the behavior is important. If neither state feels a difference is important, the implementation of the baseline system may be left intact, or a selection may be driven by a desire to reduce the impact of interfaces with other modules in one or both systems. If both states feel strongly that a feature should be implemented differently, a configuration option may be introduced to allow either behavior. However, the states should attempt to minimize the number of differences introduced, as they will inevitably lead to higher costs in the future as different system configurations must be tested and validated with other changes.
7. **Mutual Requirement Generation**
   With the agreement of each state on a functionality set for the subsystem(s), (possibly with some minor configurable differences) a set of joint requirements should be developed and approved. These requirements may be based on existing requirements from either system where appropriate, or may be generated anew if required by decisions from the CMBs.
8. **Baseline Identification**
   With features and requirements set, a software baseline should be identified. This will typically consist of one system's subsystem(s), possibly with discrete blocks of code from the other system extracted to provide a base for specific functionality.
9. **Review Proposed Modifications**
   Once the baseline and feature set is defined, updates can be identified for any relevant changes to schemas, database tables, and user interface which may be required. These modifications should be reviewed and accepted by the appropriate stakeholders for each state prior to implementation of the changes.

10. **Subsystem Revisions**
    a. **Implementation**
       Following approval of the any design changes, modifications to the implementation of the merged subsystem(s) can begin. This task may involve personnel from one or both project teams, depending on the specific work to be performed and the organization providing funding for the effort.
    b. **Integration**
       With the merged subsystem(s) developed, other custom modules in each system may be updated to reflect any schema or functionality changes that were introduced with the merge.
    c. **Testing and Validation**
       Once implementation of the merged functional area is completed and interfaces have been updated, the system may begin the acceptance and validation testing process. This may follow existing procedures for each state, or may be performed under a joint testing process. Any joint testing must ensure that all the combined stakeholder needs are considered.
11. **Deployment**
    After acceptance and validation of the software, deployments may begin in accordance with the deployment schedule and procedure of each state.
12. **Phase Review**
    As deployment activities move forward, stakeholders should review the process that was followed for the recently completed phase to identify the source of any significant issues or difficulties (lessons learned). Improvements to the process may be proposed and approved prior to initiation of the next migration phase.

Suggested Order of Migration

To some extent, the order of migration of functional areas is arbitrary; however, certain factors may help guide a selection of appropriate subsystems in the initial phases. The following items are suggested for the first phases, to make the most efficient use of the current state of the systems and provide a slightly simpler starting point. After these phases have been completed, the states will be better positioned to verify that the effort should be continued.

Phase 0: Central Architecture Components

Before attempting to migrate subsystems from one ATMS to the other, several core differences should be resolved. A unified ITSGenericLibrary, Status Logger client, and Executive Handler client should be created to provide a common foundation. The Status Logger and Executive Handler UIs could also be merged at this time. All subsystems in both ATMSs would need to be updated to properly reference the newly merged modules. As part of this phase, a joint library for data objects can be started. As subsystems are migrated, their objects would migrate into the common library.

Another core task of this pre-phase is creating a common GUI architecture which can be used by both systems' user interface modules. This common architecture would address communication to DataBus/CSD, subsystem message exchange, subsystem startup and shutdown, dialog display, and other core, common tasks for subsystems. This phase would not directly address items like icon display and map interactions, as these tasks are specific to the current GUIs, but would provide an interface to unify how those actions are invoked. As part of this task, consideration should be given to the standalone user interfaces provided by Lonestar. These UIs do

not require a map background, and are typically limited to one subsystem. The GUI architecture would need to ensure this concept can be maintained.

The final major task of this phase is to incorporate Lonestar's SAA into SunGuide. While the SunGuide CMB has currently approved adopting SAA with a variety of enhancements, it would simplify this effort to integrate SAA as it currently exists without introducing any functionality changes. After the integration is performed, future phases could introduce new SAA functionality, but in the interests of developing a unified system, this component should initially be left intact. This change will likely require approval by the SunGuide CMB.

Phase 1: Device Messaging

With the core architectural differences between the systems resolved (or at least minimized) actual updates to full subsystems can begin. A good initial functional area to resolve would be device messaging; specifically, this includes DMS, MAS/MQA, and HAR. The processes have several operational differences between the operations of the systems in these areas which will need to be resolved; making this is an excellent starting point to define the process of reconciling differences that both states will need to make to have a truly unified system. Additionally, the user interfaces for these subsystems are being revamped with SunGuide 6.0, which means the bulk of work for UI design and development has already been completed. (Note that this work would need to be adapted to the new UI architecture developed in Phase 0, so there is still some integration to perform.) HAR is included in this block because it is part of the unified messaging interface of SunGuide 6.0, and would provide new functionality to Lonestar.

A complication of this phase is that the new UIs have been developed without input from TxDOT. TxDOT should be given an opportunity to review the UIs and provide comments, possibly introducing changes to help the dialogs better work within their operations. This would also provide a direct opportunity for the states to coordinate from a concrete starting point.

Phase 2: Traffic Detection

Following device messaging, traffic detection provides a good second block of functionality to be merged. This includes TSS and TTA/TVT, and possibly other modules. New dialogs for TVT are also part of SunGuide 6.0, but TSS dialogs have not yet been updated, which would introduce the concept of joint UI design between the states.

Phase 3 and Beyond

With these initial efforts complete, the process of identifying, analyzing, and merging subsystems should be well established, and additional work can be continued based on the priorities of the states. Some subsystems may be identified as being poor candidates for merging, notably Event Management. Each state takes a very different approach to their event management operations, and finding a common set of functionality may be difficult. Additionally, at some point all user interface actions will be routed through the common architecture introduced in Phase 0. At this point, it may be possible for one state to fully adopt the other's user interface, or for some new UI to be introduced. Either option would provide a further unified system that would introduce eventual savings in cost and development time.

Additional Considerations

With a unified system, it will be important to accurately reference an iteration of the full ATMS regardless of which state the deployment is in. Because of this, it would be beneficial to synchronize the version numbers of the systems, and reconcile the difference between the states' approach to versioning. To present a unified system and documentation set to users, administrators, and developers, it would be preferable to keep all module versions synchronized with the main system version number.

Additional consideration should be given to the branding of the systems. Lonestar and SunGuide have strong historical name recognition, and these names should continue with the unified system. However, it would also be beneficial to have a new unifying product name to present a more cohesive face to other states which might be interested in using the software. This could be referenced by describing the systems as "[Unified Name]: SunGuide" or "Lonestar, a [Unified Name] ATMS" in promotional materials. This is something the stakeholders will need to address.

Conclusion

While starting from a common source, Lonestar and SunGuide have diverged based on the technical and schedule needs of their users. This common origin provides the key point to performing a reintegration of the systems, and helps illustrate the path forward. Given support from both states, a unified system can be developed which will provide future benefits to both Florida and Texas, as well as any other states that see the value of using a common software package owned by its users.