SunGuide^{TM}:

Managed Lanes System Interface Control Document

SunGuide-MLS-ICD-7.0

Prepared for:

Florida Department of Transportation Traffic
Engineering and Operations Office 605
Suwannee Street, M.S. 90
Tallahassee, Florida 32399-0450
(850) 410-5600

December 19, 2017

Document Control Panel

File Name:	SunGuide-MLS-ICD-7.0		
File Location:	SunGuide CM Repository		
CDRL:			
Name		**Initial**	**Date**
Created By:	Tucker Brown	TJB	12/19/17
Reviewed By:			
Modified By:			
Completed By:			

Table of Contents

List of Figures

List of Acronyms

ATMS	Advanced Traffic Management System
DOT	Department of Transportation
FDOT	Florida Department of Transportation
ITS	Intelligent Transportation Systems
ITN	Invitation to Negotiate
SAG	SunGuide Architecture Guidelines
SDD	Software Design Document
SwRI	Southwest Research Institute
TMC	Traffic Management Center
TSS	Transportation Sensor System
XML	Extensible Markup Language

REVISION HISTORY

Revision	Date	Changes
1.0.0	December 19, 2017	Initial Release

1. Scope

1.1 Document Identification

This Interface Control Document (ICD) describes the interface between individual SunGuide^{TM} clients and the Managed Lanes Subsystem (MLS) and between MLS and the associated drivers. The general base architecture of the XML communications including connection information, byte order and base transaction classes is delineated in this document. This ICD defines Extensible Markup Language (XML) schemas upon which XML requests shall be based in communicating amongst the various processes. Refer to the SunGuide General Interface Document for details regarding data transfer.

1.2 Project Overview

The Florida Department of Transportation (FDOT) is conducting a program that is developing SunGuide software. The SunGuide software is a set of Intelligent Transportation System (ITS) software that allows the control of roadway devices as well as information exchange across a variety of transportation agencies. The goal of the SunGuide software is to have a common software base that can be deployed throughout the state of Florida. The SunGuide software development effort is based on ITS software available from the state of Texas; significant customization of the software is being performed as well as the development of new software modules. The following figure provides a graphical view of the software to be developed:

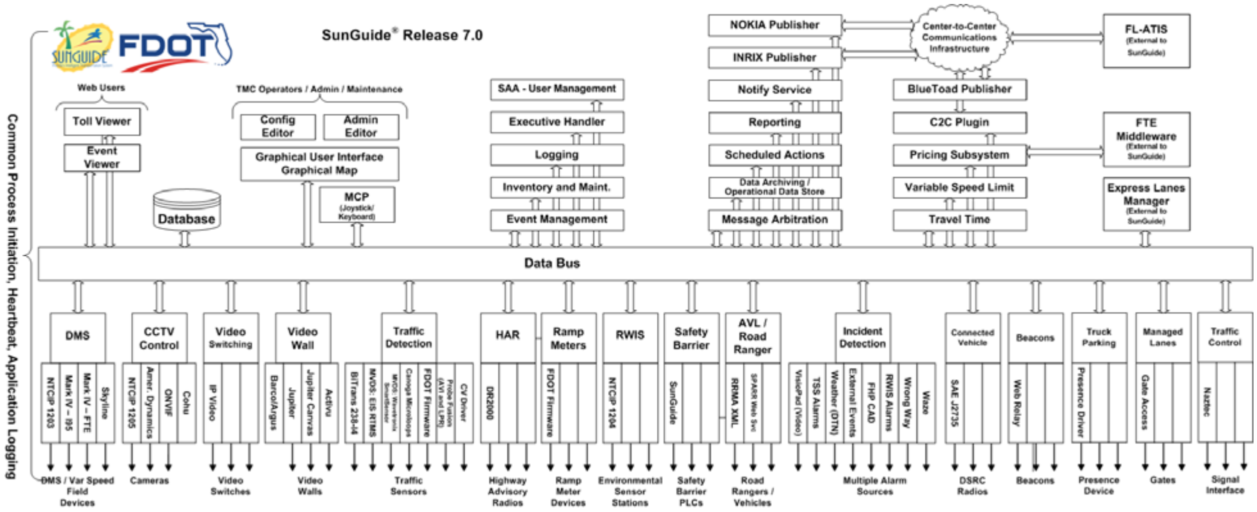

Figure 1-1. High-Level ArchitecturalConcept

1.3 How to Use This Document

The ICDs describe the specific interface between two SunGuide subsystems or between a SunGuide subsystem and a SunGuide driver. The relationship of appropriate documents is shown in the Figure 1-2.

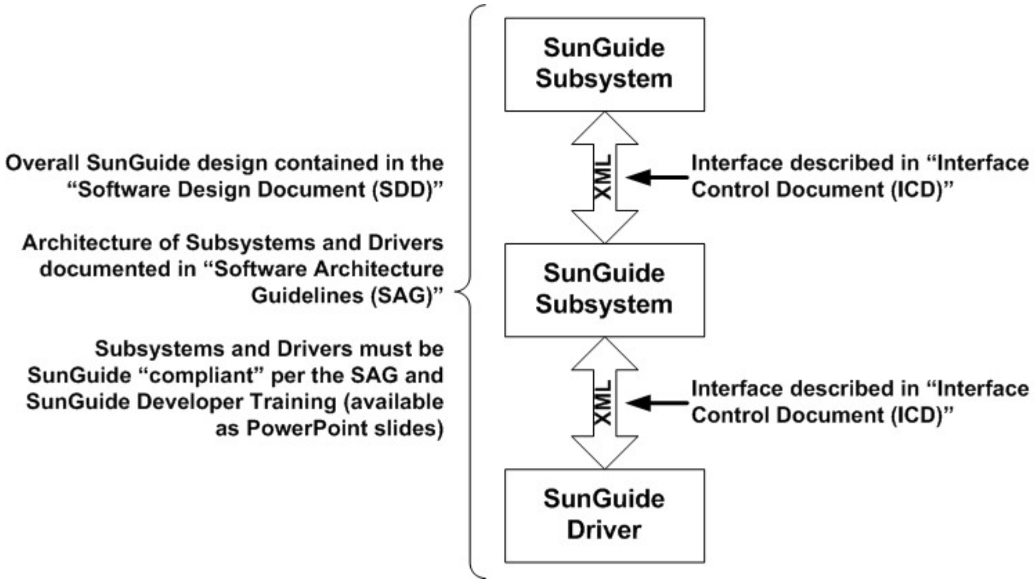
Overall SunGuide design contained in the "Software Design Document (SDD)"
Architecture of Subsystems and Drivers documented in "Software Architecture Guidelines (SAG)"
Subsystems and Drivers must be SunGuide "compliant" per the SAG and SunGuide Developer Training (available as PowerPoint slides)
SunGuide Subsystem
XML
Interface described in "Interface Control Document (ICD)"
SunGuide Subsystem
XML
Interface described in "Interface Control Document (ICD)"
SunGuide Driver

Figure 1-2. SunGuide DeveloperDocumentation

This document describes an *internal* SunGuide interface. The interface described is between two SunGuide compliant processes. The reader should review the following document to gain an understanding of how SunGuide compliant application is created (this will vary if the application is a driver or subsystem):

SunGuide Software Architecture Guidelines (SAG)

The SAG describes what needs to be included in a SunGuide application to assure that it will work cooperatively in the SunGuide environment. Once the SAG is reviewed, the following document should be reviewed:

SunGuide Software Design Document (SDD)

The SDD will provide an understanding of how individual components of SunGuide were designed. Finally the ICD, along with the associated schema should be reviewed to determine what data needs to be exchanged on the interface being defined in this document.

Additionally, a SunGuide "Developer Training" class is available that provides the students with an introduction into developing SunGuide processes. The SunGuide source code repository has a generic subsystem and a generic driver available that can be used as the basis for developing a new application.

1.4 Related Documents

The following documents were used to develop this document:

- FDOT Scope of Services: BE492, Standard Written Agreement for SunGuide Software Support, Maintenance, and Development, Exhibit A: Scope of Services. December 14, 2017.
- Notice to Proceed: Letter to Southwest Research Institute® (SwRI®) for BE492, December 14, 2017
- Letter of Authorization 001: Letter to SwRI for BE492, December 15, 2017.
- SunGuide Project website: http://sunguidesoftware.com.

1.5 Contacts

The following are contact persons for the SunGuide software project:

- Fred Heery, ITS Section, TSM&O, Central Office
  fred.heery@dot.state.fl.us, 850-410-5606
- Derek Vollmer, ITS Section, TSM&O, Central Office
  Derek.Vollmer@dot.state.fl.us, 850-410-5615
- Mark Dunthorn, AECOM Project Manager,
  mark.dunthron@dot.state.fl.us, 850-410-5623
- Tucker Brown, SwRI Project Manager,
  tbrown@swri.com, 210-522-3035
- Roger Strain, SwRI Software Project Manager,
  rstrain@swri.org, 210-522-6295

2. Data

The following sections detail the XML transactions that can be exchanged between client and server applications.

2.1 Schema

The schemas for these transactions may be located in the Schemas directory. The objects directory contains common data schemas that are used by the various request, messages, and responses. Schemas are organized in the following tree structure:

- messages
  - actionListDeletedMsg.xsd
  - actionListStatusUpdateMsg.xsd
  - actionStatusUpdateMsg.xsd
  - controllerLastCommUpdateMsg.xsd
  - controllerStatusUpdateMsg.xsd
  - gateStatusUpdateMsg.xsd
  - rampStatusUpdateMsg.xsd
  - segmentStatusUpdateMsg.xsd
- objects
  - action.xsd
  - actionList.xsd
  - actionListTemplate.xsd
  - actionResolution.xsd
  - actionTemplate.xsd
  - cameraVerification.xsd
  - controller.xsd
  - controllerConfig.xsd
  - controllerStatus.xsd
  - deviceActionTemplate.xsd
  - deviceData.xsd
  - deviceRequisite.xsd
  - directionStatus.xsd
  - dmsActionTemplate.xsd
  - dmsRequisite.xsd
  - gate.xsd
  - gateStatus.xsd
  - managedRoad.xsd
  - mlsActionTemplate.xsd
  - mlsRequisite.xsd
  - mlsState.xsd
  - prereqActionTemplate.xsd
  - ramp.xsd
  - rampStatus.xsd
  - segment.xsd

  - segmentStatus.xsd
  - statusListItem.xsd
  - templateData.xsd
  - user.xsd
- requests
  - abortActionListReq.xsd
  - addActionListTemplateReq.xsd
  - addActionTemplateReq.xsd
  - addControllerReq.xsd
  - addGateReq.xsd
  - addManagedRoadReq.xsd
  - addRampReq.xsd
  - addSegmentReq.xsd
  - assignActionListReq.xsd
  - controllerStatusReq.xsd
  - deleteActionListTemplateReq.xsd
  - deleteActionTemplateReq.xsd
  - deleteControllerReq.xsd
  - deleteGateReq.xsd
  - deleteManagedRoadReq.xsd
  - deleteRampReq.xsd
  - deleteSegmentReq.xsd
  - executeActionListReq.xsd
  - executeActionReq.xsd
  - executeMlsActionReq.xsd
  - modifyActionListTemplateReq.xsd
  - modifyActionTemplateReq.xsd
  - modifyControllerReq.xsd
  - modifyGateReq.xsd
  - modifyManagedRoadReq.xsd
  - modifyRampReq.xsd
  - modifySegmentReq.xsd
  - overrideActionReq.xsd
  - retrieveDataReq.xsd
  - setOpStatusReq.xsd
  - subscribeReq.xsd
  - verifyActionReq.xsd
- responses
  - abortActionListResp.xsd
  - addActionListTemplateResp.xsd
  - addActionTemplateResp.xsd
  - addControllerResp.xsd
  - addGateResp.xsd
  - addManagedRoadResp.xsd
  - addRampResp.xsd

- addSegmentResp.xsd
- assignActionListResp.xsd
- controllerStatusResp.xsd
- deleteActionListTemplateResp.xsd
- deleteActionTemplateResp.xsd
- deleteControllerResp.xsd
- deleteGateResp.xsd
- deleteManagedRoadResp.xsd
- deleteRampResp.xsd
- deleteSegmentResp.xsd
- executeActionListResp.xsd
- executeActionResp.xsd
- executeMlsActionResp.xsd
- modifyActionListTemplateResp.xsd
- modifyActionTemplateResp.xsd
- modifyControllerResp.xsd
- modifyGateResp.xsd
- modifyManagedRoadResp.xsd
- modifyRampResp.xsd
- modifySegmentResp.xsd
- overrideActionResp.xsd
- retrieveDataResp.xsd
- setOpStatusResp.xsd
- subscribeResp.xsd
- verifyActionResp.xsd

Requests may be sent from a client to a subsystem or from a subsystem to a driver. Responses may be sent from a driver to a subsystem or a subsystem to a client. A message can be sent from any process to another process.

2.1.1 Subsystem Communication

Initial communication to a subsystem is described in the general ICD. For MLS, the action lists, action list templates, gates, ramps, and segments are retrieved from the database on startup. Once a client has initiated the connection to MLS, additional configurations may be initiated and action lists can be executed.

The following table shows the various subscriptions a client may request. The last column shows the XML updates that will be received if a client has subscribed to this data.

Subscription	Description	Updates Received
deviceData	Receive updates for device information.	addControllerResp deleteControllerResp modifyControllerResp addGateResp modifyGateResp deleteGateResp addManagedRoadResp modifyManagedRoadResp deleteManaged RoadResp addRampResp modifyRampResp deleteRampResp addSegmentResp modifySegmentResp deleteSegmentResp
templateData	Receive updates to action list templates.	addActionTemplateResp modifyActionTemplateRep deleteActionTemplateResp addActionListTemplateResp modifyActionListTemplateResp deleteActionListTemplateResp
actionStatus	Receive updates to the status of action lists.	executeActionListResp actionStatusUpdateMsg actionListStatusUpdateMsg actionListDeletedMsg
deviceStatus	Receive updates to device status.	setOpStatusResp controllerStatusResp controllerLastCommUpdateMsg gateStatusUpdateMsg rampStatusUpdateMsg segmentStatusUpdateMsg controllerStatusUpdateMsg
userData	Receive updates to changes in users.	userUpdateMsg

2.1.2 Driver Communication

Initial communication from a subsystem to a driver is described in the general ICD. For MLS, an addControllerReq and addGateReq is sent to the driver. After receiving this request, the driver begins polling the controller. For each poll, the driver will send a controllerLastCommUpdateMsg to MLS.

As in Section 2.1.1, additional controllers and gates may be added.

2.2 Examples

For example, if a client wishes to add a controller to the system, the client sends an addControllerReq to the subsystem. Once the subsystem has verified this is a new controller, the request is then forwarded to the appropriate driver. The driver adds the controller and sends an addControllerResp to the subsystem. The subsystem then sends this response back to the appropriate client and to any clients who have subscribed to detector data.

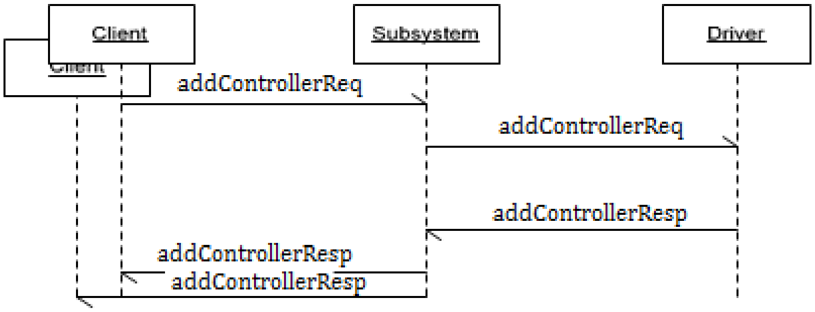

Figure 2-1. Sample Transaction

The tables below show which requests can be sent from client to subsystem and subsystem to driver. The responses sent from driver to subsystem and subsystem to client are also specified. Messages are sent when a response is not required.

2.3 Subsystem Schemas

FC (From client), TC (To client), TD (To driver), FD (From driver)

Usage Description	Requests	FC	TD	Responses	FD	TC	Messages	TD	FD	TC
Used for a client to cancel running an action list.	abortActionListReq	X		abortActionListResp		X	actionListStatusUpdateMsg			X
Used for a client to add a new action list template to the subsystem.	addActionListTemplateReq	X		addActionListTemplateResp		X				
Used for a client to add an action template to the subsystem.	addActionTemplateReq	X		addActionTemplateResp		X				
Used for a client to add a controller to the subsystem.	addControllerReq	X	X	addControllerResp	X	X				
Used for a client to add a gate to the subsystem.	addGateReq	X	X	addGateResp	X	X				
Used for a client to add a managed road to the subsystem.	addManagedRoadReq	X		addManagedRoadResp		X				
Used for a client to add a ramp to the subsystem.	addRampReq	X		addRampResp		X				

Usage Description	Requests	FC	TD	Responses	FD	TC	Messages	TD	FD	TC
Used for a client to add a segment to the subsystem.	addSegmentReq	X		addSegmentResp		X				
Used for a client to assign an action list to a user.	assignActionListReq	X		assignActionListResp		X	actionListStatusUpdateMsg			X
Used for a client to manually request the status of a controller.	controllerStatusReq	X	X	controllerStatusResp	X	X	controllerStatusUpdateMsg		X	X
Used for a client to delete an action list template from the subsystem.	deleteActionListTemplateReq	X		deleteActionListTemplateResp		X				
Used for a client to delete an action template from the subsystem	deleteActionTemplateReq	X		deleteActionTemplateResp		X				
Used for a client to delete a controller from the subsystem.	deleteControllerReq	X	X	deleteControllerResp	X	X				
Used for a client to delete a gate from the subsystem.	deleteGateReq	X	X	deleteGateResp	X	X				
Used for a client to delete a managed road from the subsystem.	deleteManagedRoadReq	X		deleteManagedRoadResp		X				

Usage Description	Requests	FC	TD	Responses	FD	TC	Messages	TD	FD	TC
Used for a client to delete a ramp from the subsystem.	deleteRampReq	X		deleteRampResp		X				
Used for a client to delete a segment from the subsystem.	deleteSegmentReq	X		deleteSegmentResp		X				
Used for a client to start the execution of an action list.	executeActionListReq	X		executeActionListResp		X				
Used for a client to execute a single action.	executeActionReq	X		executeActionResp		X	actionStatusUpdateMsg			X
Used for a client to modify an action list template in the subsystem.	modifyActionListTemplateReq	X		modifyActionListTemplateResp		X				
Used for a client to modify an action template in the subsystem.	modifyActionTemplateReq	X		modifyActionTemplateResp		X				
Used for a client to modify a controller in the subsystem.	modifyControllerReq	X	X	modifyControllerResp	X	X				
Used for a client to modify a gate in the subsystem.	modifyGateReq	X	X	modifyGateResp	X	X				
Used for a client to modify a managed road in the subsystem.	modifyManagedRoadReq	X		modifyManagedRoadResp		X				

Usage Description	Requests	FC	TD	Responses	FD	TC	Messages	TD	FD	TC
Used for a client to modify a ramp in the subsystem.	modifyRampReq	X		modifyRampResp		X				
Used for a client to modify a segment from the subsystem.	modifySegmentReq	X		modifySegmentResp		X				
Used for a client to override an action in an action list.	overrideActionReq	X		overrideActionResp		X	actionStatusUpdateMsg			X
Used for a client to retrieve the stored objects of the subsystem to generate an initial cache.	retrieveDataReq	X		retrieveDataResp		X				
Used for a client to set the operational status of a device.	setOpStatusReq	X		setOpStatusResp		X				
Used for a client to subscribe to updates for objects in the subsystem.	subscribeReq	X		subscribeResp		X				
Used for a client to verify an action within an action list has been completed.	verifyActionReq	X		verifyActionResp		X	actionStatusUpdateMsg			X
Used to send an update from the subsystem to client indicating the communication status of a controller has changed.							controllerLastCommUpdateMsg		X	X

Usage Description	Requests	FC	TD	Responses	FD	TC	Messages	TD	FD	TC
Used to send an update from the subsystem to client indicating the status of a controller has changed.							controllerStatusUpdateMsg		X	X
Used to send an update from the subsystem to client indicating the status of a gate has changed.							gateStatusUpdateMsg		X	X
Used to send an update from the subsystem to client indicating the status of a ramp has changed.							rampStatusUpdateMsg			X
Used to send an update from the subsystem to client indicating the status of a segment has changed.							segmentStatusUpdateMsg			X
Used to send an update from the subsystem to the client indicating the action list should be added to the status cache.							actionListStatusUpdateMsg			X

Usage Description	Requests	FC	TD	Responses	FD	TC	Messages	TD	FD	TC
Used to send an update from the subsystem to the client indicating the action list should be removed from the status cache.							actionListDeletedMsg			X
Used to send an update from the subsystem to the client indicating the status of an action has changed.							actionStatusUpdateMsg			X

2.4 Driver Schemas

TD (To driver), FD (From driver)

Usage Description	Requests	TD	Responses	FD	Messages	TD	FD
Used for a client to add a controller to the driver.	addControllerReq	X	addControllerResp	X			
Used for a client to delete a controller from the driver.	deleteControllerReq	X	deleteControllerResp	X			
Used for a client to modify a controller in the driver.	modifyControllerReq	X	modifyControllerResp	X			
Used to notify the subsystem that the communication status with the controller has changed.					controllerLastCommUpdateMsg		X
Used to notify the subsystem that the status of the controller has changed.					controllerStatusUpdateMsg		X
Used for a client to add a gate to the driver.	addGateReq	X	addGateResp	X			

Used for a client to modify a gate in the driver.	modifyGateReq	X	modifyGateResp	X			
Used for a client to delete a gate from the driver.	deleteGateReq	X	deleteGateResp	X			
Used to notify the subsystem that the status of the gate has changed.					gateStatusUpdateMsg		X
Used for a client to request execution of a gate action.	executeMlsActionReq	X	executeMlsActionResp	X			

3. Notes

Information about XML and schemas can be found at the World Wide Web Consortium (W3) website at http://www.w3.org.