Title
TMDD Standards Review Technical Memorandum

Author
Peterson, Brian

Publication Date
2021-02-16

PARTNERS FOR ADVANCED TRANSPORTATION TECHNOLOGY
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Modernization of Center-to-Center Data Communication Standards
Task 3713 (65A0761)

TMDD Standards Review
Technical Memorandum

V1.1
February 16, 2021

Partners for Advanced Transportation Technology works with researchers, practitioners, and industry to implement transportation research and innovation, including products and services that improve the efficiency, safety, and security of the transportation system.

This page left blank intentionally

Primary Authors

Brian Peterson
Software Engineering Manager
California PATH
University of California, Berkeley

Editorial Review

Michelle Harrington
Technical Writer
California PATH
University of California, Berkeley

This page left blank intentionally

Table of Contents

LIST OF FIGURES

This page left blank intentionally

LIST OF TABLES

This page left blank intentionally

1. INTRODUCTION

This document presents a review of the Traffic Management Data Dictionary (TMDD) standard, specifically version 3.03d with some analysis of the newest version 3.1, released in January 2020. The intention of this document is to provide a review of this standard for transmission of data between traffic management centers (TMCs), with specific commentary on usability of the standard with specific examples based on its implementation in the Caltrans I-210 Connected Corridors program.

TMDD is published by the Institute of Transportation Engineers (ITE), an international organization of transportation engineers and the American Association of State Highway and Transportation Officials (AASHTO), an organization of highway and transportation officials in the United States with members from all 50 states, the District of Columbia, and Puerto Rico. The standard is provided in two parts, along with a reference guide, providing guidance for those developing, procuring, or implementing systems that communicate traffic and transportation device information between or within traffic management centers (TMCs). This includes information regarding system requirements, interface design, communication methods, as well as data schema and definitions.

1.1. PURPOSE OF DOCUMENT

This document is prepared as a review of the standard, based on experienced gained in the California Department of Transportation sponsored I-210 Connected Corridors Project (I-210 Project). It includes a review of the specification itself and practical knowledge gained in its implementation in the I-210 Project by the University of California, Berkeley integrating systems of the cities of Pasadena, Arcadia, Monrovia, and Duarte; LA County; the State of California, as well as the Integrated Corridor Management System developed by UC Berkeley.

The document is intended to provide a basis for recommendations for improvement to the standard, along with three additional documents:

- TMDD Modernization Software and Systems Standards Recommendations Technical Memorandum
- TMDD Modernization Current and Future Transportation Management High Level Requirements Technical Memorandum
- TMDD Modernization Gap Analysis Technical Memorandum

The recommendations for improvement developed from these four documents will propose specific modifications to the standard.

1.2. INTENDED AUDIENCE

The primary audience for this document includes:

- The Caltrans Division of Research, Innovation, and System Information.
- TMDD Steering Committee
- Caltrans Operations personnel involved in specifying, procuring, and implementing systems requiring C2C communications
- Transportation systems vendor community

1.3. DOCUMENT ORGANIZATION

The remainder of this document is organized as follows:

- **Section 2** presents a general description of TMDD and its uses.
- **Section 3** provides a review of the standard, focusing on issues identified during the implementation of Caltrans' Connected Corridors program, as well as technology related issues.
- **Section 4** provides a summary of the findings and recommendations.

2. GENERAL DESCRIPTION OF THE TRAFFIC MANAGEMENT DATA DICTIONARY (TMDD)

2.1. INTRODUCTION TO THE TRAFFIC MANAGEMENT DATA DICTIONARY

The Traffic Management Data Dictionary (TMDD)[1] is a standard provided by the Institute of Transportation Engineers (ITE) and the American Association of State Highway and Transportation Officials (AASHTO). It is meant to "assist users in the procurement process by describing the potential user needs, establishing requirements, and tracing them to data content for system interface to facilitate information exchanges among centers." [TMDD Steering Committee of ITE and AASHTO Guide to Traffic Management Data Dictionary Standard v3.1 for Center-to-Center Communications]. It provides a basis for creating interface standards between systems that will exchange traffic management information.

It is composed of two sections. Volume 1 is the Concept of Operations and Requirements volume. It provides guidelines for developing requirements for a system interface, helping those involved in specifying, procuring, designing, developing, and implementing such communication interfaces to define what is required for their specific implementation. It defines user needs and operational requirements, allowing users of the standard, when implementing the standard to select the proper dialogs and specifics of implementation suited for their project. In addition, it defines how users may customize the standard to meet the specific needs of their project and imposes limitations on those customizations.
Volume 2 is the Design Content volume. This volume details the technical design details of the communicate standard. An explanation of the standards communication mechanisms is provided, including:

- Request – Response communication method
- Publication and subscription communication method
- Types of subscriptions available
- Communication structure, including dialogs, messages, dataframes, and data elements

In addition, the specific dialogs, messages, dataframes and data elements are described in detail, as well as their relationships.

Companion XSD and WSDL files are provided for defining SOAP implementations of the service. A third publication, a "Guide to V3.1 Traffic Management Data Dictionary (TMDD) Standard for Center-to-Center Communications"[2] is also provided to assist in understanding and using the standard.

2.2. COMMUNICATION BASICS

2.2.1. SIMPLE OBJECT ACCESS PROTOCOL

TMDD utilizes the Simple Object Access Protocol (SOAP) as its communication protocol. SOAP is a type of web service first developed in 1998, with a draft version submitted to the Internet Engineering Task Force in late 1999. Version 1.1 of the specification was published in 2000, and version 1.2 in 2007. Early issues with interoperability of the specification between major software vendor implementations (Microsoft and Sun primarily) were addressed in a separate specification: Web Services Interoperability (WS-I) Basic Profile version 1.0[3] released in 2004. This addressed differences in implementation between the vendors software development tools and resulting SOAP implementations. Version 1.1[4] of the basic profile was released in 2006, and both versions 1.2[5] and 2.0[6] in 2010. TMDD uses version 1.1 of the SOAP protocol and is not WS-I compliant, as it violates two specific principles of WS-I: utilizing multi-part messages and having non-unique body parts with the document/literal style SOAP binding.

2.2.2. OWNER CENTER AND EXTERNAL CENTER

Center-to-Center Communication is, at its most fundamental level, a communication of traffic data between two Traffic Management Centers (TMCs). One center is the owner center and the other is the external center. The owner center is the source of information, generally owning and managing the traffic management resources such as intersection signals, sensors, etc. The external center is the TMC requesting the information, with no ownership or direct control of the traffic management resources. Within TMDD, the external center is receiving data regarding the status, inventory, or other related information of the owner centers traffic management assets. The external center may also request a change in the state of the owner center's assets. An example of this may be an external center managed by one city requesting a neighboring city, the owner center, to change its signal timing at a bordering intersection in order to coordinate signals and improve traffic flow between the two cities.

2.2.3. BASIC COMMUNICATION METHODS

The basic communication methods used by TMDD include request/response and subscription. These are specified by NTCIP 2306, from which TMDD derives its own specification details. Request/response is exactly what it sounds like. A message is sent from one TMC (owner or external), and a corresponding response is provided by the other. The request and response are synchronous events. Additionally, the request/response communications may be for one-time information requests or for command requests.

Subscriptions are asynchronous in nature. These work by one center (generally external) first requesting a subscription via a SOAP message. The other center acknowledges the request and, assuming the request is valid, sets up the subscription based on the parameters provided in the request. The subscription is a request for information that the receiver of the subscription request “promises” to fulfill when information that matches the request is available. There are three types of subscriptions within TMDD:

- One-time – This is a one-time update to be sent by the owner center to the external center. This is similar to the request/response mechanism, but is asynchronous in nature.
- Periodic – In this type of subscription, the external center that sends the subscription request is asking for successive updates to the information based on a specified time interval included in the subscription request. An example of this would be a periodic subscription of a specific loop-sensor, with a period of every 30 seconds. In this example, the owner center would send a message every 30 seconds (a publication) with the data for that specified center to the external center. The external center only requests once, with multiple updates from the owner center. Note that the TMDD specification does not provide guidance regarding coordination of timing between device to owner center reporting and owner center to external center subscription timing, nor does it provide guidance regarding how to address data reporting with non-synchronized timing.
- On-change – This subscription type is similar to the periodic subscription, but rather than data being returned on a set time period, the data is returned every time the source data is updated from the field or within the owner center’s systems. For the sensor example in the periodic subscription description, rather than setting up a periodic subscription, the external center could request an on-change subscription. That way, if the data is collected from the field every minute rather than every 30 seconds, the external center would receive a message approximately every minute. This has the advantage of not requiring the external center to know specific refresh rates of the source of information for the owner center’s traffic management assets.

2.2.4. DIALOGS, MESSAGES, DATAFRAMES, AND DATA ELEMENTS

TMDD has a defined structure for its communications and how the request/response and subscription communications are defined and implemented. The structure is broken down into four distinct levels:

- **Dialogs** – Dialogs are used to define a conversation between the owner center and external center. Dialogs are defined as a set of specific messages and sequence of exchange that are used to transmit information between centers. For example, a simple dialog defined within TMDD is the dlCenterActiveVerificationRequest. This dialog is used by one center (owner or external) to determine if the other center is currently active and processing requests. It is a request/response dialog that has three defined messages. The first is the MSG_CenterActiveVerificationRequest, the second the MSG_CenterActiveVerificationResponse providing the center status, and the third message, the MSG_ErrorReport, is defined for reporting any errors that may occur. For subscription dialogs, more messages are defined to complete the data exchange. In the case of providing center active information, a corresponding CenterActiveVerificationSubscription dialog is provided for subscription communication.

This dialog includes the MSG_CenterActiveVerificationSubscription message and the MSGConfirmationReceipt message use to confirm the subscription. An MSG_ErrorReport is also provided for exception handling. This dialog only establishes the subscription. An update message is used for sending the information requested. In the case of the center active subscription, the dlCenterActiveVerificationUpdate dialog defines those updates used to provide the actual center active data.

- **Messages** – Messages are the individual SOAP messages sent within one or more defined dialogs. They consist of dataframes and data elements, and transmit the required information contained in the communication. Each message is analogous to a sentence in the conversation defined by the dialog.
- **Dataframes** – Dataframes are the objects exchanged within a message. Think of them as the "nouns" within the message "sentence." For example, within the dialog dlRampMeterInventoryUpdate that provides a ramp meter inventory update to a target system as a result of a current subscription to ramp meter inventory, there is the MSG_RampMeterInventoryUpdate message. Within that message, there are dataframes that represent the ramp-meter-inventory-item and metered-lane objects. The dataframes used to represent these items is the device-inventory-header and metered-lane-inventory-header.
- **Data Elements** – Data elements are the individual data fields provided in the collection of information transmitted.

The hierarchy of how these four levels relate is illustrated below:

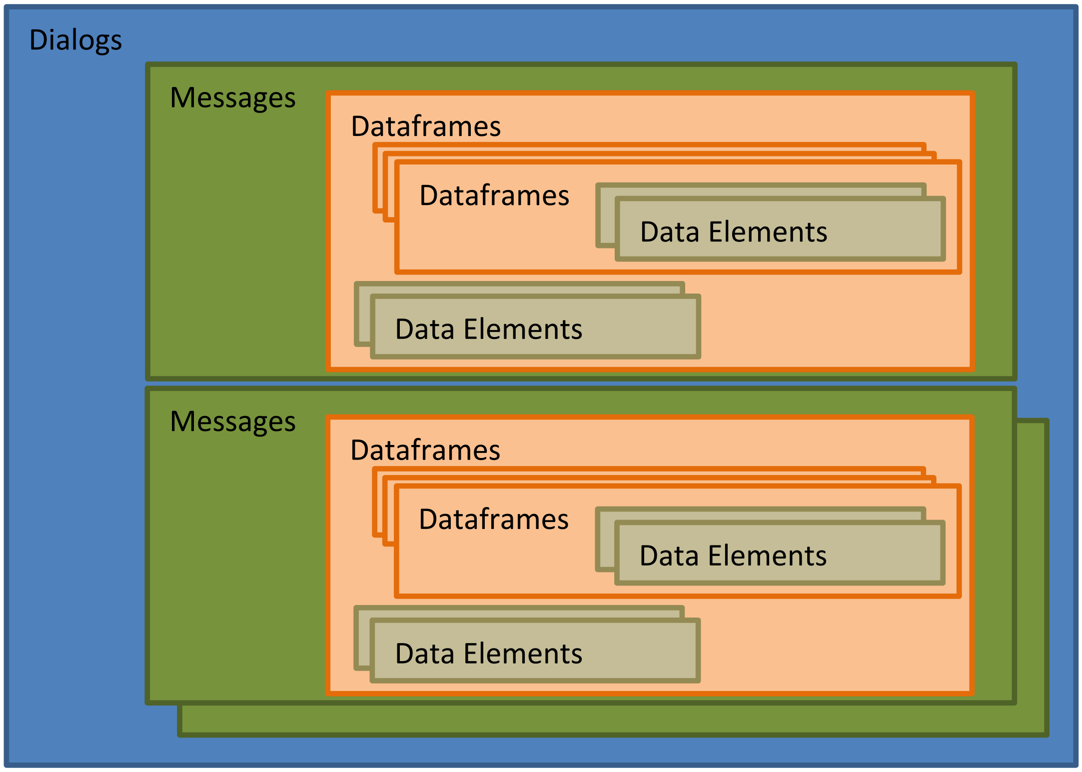

Figure 2-1 TMDD Data Exchange Hierarchy

2.3. INFORMATION EXCHANGED VIA TMDD

The information exchanged via TMDD is varied. In general, it includes information regarding transportation elements managed by TMCs. It provides some level of flexibility as well to meet specific project requirements via the optional elements allowed within the data exchanged. For example, each project implementation using TMDD can choose select dialogs to support the information desired, and not implement any remaining dialogs. In addition, many of the data elements within TMDD are optional, so if the level of detail required is less than the full specification, those optional elements can be ignored. This allows projects to determine the size of the information exchange based on the project's scope and thereby limit the cost of implementation. The tables below provide summary information regarding the information exchanges supported by TMDD (notes are provided for clarity of information exchanged when not explicitly evident from the type of information):

Table 2-1 CCTV/Video Data Exchange

CCTV/Video Type of Data Exchanged	TMDD Dialogs Available	Notes
CCTV Inventory	dlCCTVInventoryRequest dlCCTVInventoryUpdate	
CCTV Status	dlCCTVStatusRequest dlCCTVStatusUpdate	Includes elements related to pan, tilt, zoom, iris, focus
CCTV Control	dlCCTVControlRequest	

Table 2-2 Detector Data Exchange

Detectors Type of Data Exchanged	TMDD Dialogs Available	Notes
Detector Inventory	dlDetectorInventoryRequest dlDetectorInventoryUpdate	
Detector Status	dlDetectorStatusRequest dlDetectorStatusUpdate	Includes elements related to detector lane and direction of travel
Detector Data	dlDetectorDataRequest dlDetectorDataSubscription dlDetectorDataUpdate	Includes data captured by a detector. For vehicle detectors includes elements of detector data type, vehicle speed, occupancy, count, queue length and related information.

Detectors Type of Data Exchanged	**TMDD Dialogs Available**	**Notes**
Detector Maintenance	dlDetectorMaintenanceHistoryRequest	Includes information related to detector installation, calibration, and repair

Table 2-3 Multi-Device Data Exchange

Multi-device Type of Data Exchanged	**TMDD Dialogs Available**	**Notes**
Cancel Control Request	dlDeviceCancelControlRequest	Devices include CCTV, DMS, Gates, HAR, Intersection Signals, Lane Closure Systems, Ramp Meters, Intersection Signal Sections, and Video Switches
Device Control Status	dlDeviceControlStatusRequest	Devices include CCTV, DMS, Gates, HAR, Intersection Signals, Lane Closure Systems, Ramp Meters, and Video Switches
DeviceInformation	dlDeviceInformationSubscription	Information types include inventory, status, schedules, and timing plans. Devices include • CCTV (inventory and status) • DMS (inventory, status, message appearance, pattern inventory, and font table) • Detectors (inventory, status, data, and maintenance history) • Environmental Sensors (inventory, status, data, observation metadata) • Gates (inventory, status, and schedule) • HAR (inventory, status, schedule) • Intersection Signals (inventory, status, schedule, and pattern inventory)

Multi-device Type of Data Exchanged	TMDD Dialogs Available	Notes
		• Lane Closure Systems (inventory, status, and schedule) • Ramp Meters (inventory, status, schedule, and pattern inventory) • Intersection Signal Sections (status and schedule) • Video Switches (inventory and status)

Table 2-4 Dynamic Message System Data Exchange

Dynamic Message Systems Type of Data Exchanged	TMDD Dialogs Available	Notes
Inventory	dlDMSInventoryRequest dlDMSInventoryUpdate	
Status	dlDMSStatusRequest dlDMSStatusUpdate	Includes information related to current message being displayed
Message Inventory	dlDMSMessageInventoryRequest dlDMSMessageInventorySubscription dlDMSMessageInventoryUpdate	
Message Appearance	dlDMSMessageAppearanceRequest	Includes information related to dynamic message sign (DMS) capabilities such as height, width, pixels
Font Table	dlDMSFontTableRequest	
Control Request	dlDMSControlRequest	
Priority Queue	dlDMSPriorityQueueRequest	

Table 2-5 Environmental Sensor Data Exchange

Environmental Sensors Type of Data Exchanged	TMDD Dialogs Available	Notes
Inventory	dlESSInventoryRequest	

Environmental Sensors Type of Data Exchanged	TMDD Dialogs Available	Notes
	dlESSInventoryUpdate	
Status	dlESSStatusRequest dlESSStatusUpdate	
Observation Metadata	dlESSObservationMetadataRequest	
Observation Report	dlESSObservationReportRequest dlESSObservationReportUpdate	

Table 2-6 Event Data Exchange

Events Type of Data Exchanged	TMDD Dialogs Available	Notes
Event reports and updates	dlFullEventUpdateRequest dlFullEventUpdateSubscription dlFullEventUpdateUpdate	
Event Index	dlEventIndexRequest dlEventIndexSubscription dlEventIndexUpdate	Includes information listing a center’s current events
Action Log	dlActionLogRequest dlActionLogSubscription dlActionLogUpdate	Includes event descriptive information related to the event communications log

Table 2-7 Gates Data Exchange

Gates Type of Data Exchanged	TMDD Dialogs Available	Notes
Inventory	dlGateInventoryRequest dlGateInventoryUpdate	
Status	dlGateStatusRequest dlGateStatusUpdate	
Control Request	dlGateControlRequest	
Control Schedule	dlGateControlScheduleRequest dlGateControlScheduleUpdate	

Table 2-8 HAR Data Exchange

Highway Advisory Radio Type of Data Exchanged	TMDD Dialogs Available	Notes
Inventory	dlHARInventoryRequest dlHARInventoryUpdate	
Message Inventory	dlHARMessageInventoryRequest dlHARMessageInventoryUpdate	
Status	dlHARStatusRequest dlHARStatusUpdate	
Control Request	dlHARControlRequest	
Control Schedule	dlHARControlScheduleRequest dlHARControlScheduleUpdate	
Priority Queue	dlHARPriorityQueueRequest	

Table 2-9 Intersection Signal Data Exchange

Intersection Signals Type of Data Exchanged	TMDD Dialogs Available	Notes
Inventory	dlIntersectionSignalInventoryRequest dlIntersectionSignalInventoryUpdate	
Status	dlIntersectionSignalStatusRequest dlIntersectionSignalStatusUpdate	Contains detailed intersection signal state related to operational status as well as ring status
Timing Pattern Inventory	dlIntersectionSignalTimingPatternInventoryRequest dlIntersectionSignalTimingPatternInventorySubscription dlIntersectionSignalTimingPatternInventoryUpdate	
Control Request	dlIntersectionSignalControlRequest	
Control Schedule	dlIntersectionSignalControlScheduleRequest dlIntersectionSignalControlScheduleUpdate	
Priority Queue	dlIntersectionSignalPriorityQueueRequest	

Table 2-10 Intersection Signal Section Data Exchange

Intersection Signal Sections Type of Data Exchanged	TMDD Dialogs Available	Notes
Status	dlSectionStatusRequest dlSectionStatusUpdate	
Control Request	dlSectionControlRequest dlSectionControlStatusRequest	
Control Schedule	dlSectionControlScheduleRequest dlSectionControlScheduleUpdate	
Priority Queue	dlSectionPriorityQueueRequest	
Timing Pattern Inventory	dlSectionTimingPatternInventoryRequest dlSectionTimingPatternInventorySubscription dlSectionTimingPatternInventoryUpdate	

Lane Closure Systems Type of Data Exchanged	TMDD Dialogs Available	Notes
Inventory	dlLCSInventoryRequest dlLCSInventoryUpdate	
Status	dlLCSStatusRequest dlLCSStatusUpdate	
Control Request	dlLCSControlRequest	
Control Schedule	dlLCSControlScheduleRequest dlLCSControlScheduleUpdate	

Table 2-11 Road Network Data Exchange

Transportation Network Type of Data Exchanged	TMDD Dialogs Available	Notes
Link Inventory	dlLinkInventoryRequest dlLinkInventoryUpdate	Contains information related to the road network representation using links
Link Status	dlLinkStatusRequest dlLinkStatusUpdate	Contains information related to road network state at the link level, such as vehicle restrictions, level of service, lanes,

Transportation Network Type of Data Exchanged	TMDD Dialogs Available	Notes
		detection, travel time, speed limits, traffic density, speed, volume and occupancy, and capacity
Node Inventory	dlNodeInventoryRequest dlNodeInventoryUpdate	Contains information related to road network representation, particularly link connectivity via nodes
Node Status	dlNodeStatusRequest dlNodeStatusUpdate	Contains open/closed/restricted status of a road network node
Route Inventory	dlRouteInventoryRequest dlRouteInventoryUpdate	
Route Status	dlRouteStatusRequest dlRouteStatusUpdate	Contains information related to a route's status, such as road conditions, traffic conditions, delay, volume, speed, speed limits, or occupancy
Traffic Network Information	dlTrafficNetworkInformationSubscription	Contains information related to road networks, including links, nodes, and identifying information

Table 2-12 Ramp Meter Data Exchange

Ramp Meters Type of Data Exchanged	TMDD Dialogs Available	Notes
Inventory	dlRampMeterInventoryRequest dlRampMeterInventoryUpdate	
Status	dlRampMeterStatusRequest dlRampMeterStatusUpdate	Contains information related to metered rate, mainline speed and occupancy

Ramp Meters Type of Data Exchanged	TMDD Dialogs Available	Notes
Control Request	dlRampMeterControlRequest	
Control Schedule	dlRampMeterControlScheduleRequest	
	dlRampMeterControlScheduleUpdate	
Priority Queue	dlRampMeterPriorityQueueRequest	
Pattern Inventory	dlRampMeterPlanInventoryRequest	
	dlRampMeterPlanInventorySubscription	
	dlRampMeterPlanInventoryUpdate	

Table 2-13 Video Switch Data Exchange

Video Switch Type of Data Exchanged	TMDD Dialogs Available	Notes
Inventory	dlVideoSwitchInventoryRequest	
	dlVideoSwitchInventoryUpdate	
Status	dlVideoSwitchStatusRequest	
	dlVideoSwitchStatusUpdate	
Control Request	dlVideoSwitchControlRequest	

Table 2-14 Miscellaneous Elements Data Exchange

Miscellaneous Type of Data Exchanged	TMDD Dialogs Available	Notes
Archived Traffic Data	dlArchivedDataTraffic MonitoringMetadataRequest	
Archived Data Processing Documentation	dlArchivedDataProcessingDocumentation MetadataRequest	
Connection Management	dlCenterActiveVerificationRequest	Provides response when center is actively responding to requests
	dlCenterActiveVerificationSubscription	
	dlCenterActiveVerificationUpdate	
Organization Information	dlOrganizationInformationRequest	
	dlOrganizationInformationSubscription	
	dlOrganizationInformationUpdate	

3. REVIEW OF TMDD

3.1. CLASSIFICATION OF REVIEW

This review of TMDD will be largely based upon the experience gained in the I-210 Connected Corridors program funded by the California Department of Transportation and managed by the University of California, Berkeley. In the review, the following elements will be discussed:

- The Connected Corridors implementation
- Successes
- Overview of implementation issues
- Practical experience in the Connected Corridors implementation

3.2. CONNECTED CORRIDORS IMPLEMENTATION

Connected Corridors (https://connected-corridors.berkeley.edu) is an Integrated Corridor Management (ICM) program funded by the California Department of Transportation with the goal of creating a modern platform for ICM systems and deployment for California. The program is concerned primarily with non-recurrent congestion within the I-210 corridor, developing strategies to optimize the utilization of the available capacity of the entire transportation network when significant events are present within its boundaries. The program addresses:

- Development of an ICM system that can be deployed in multiple locations within the state
- Development of partnerships and cooperation between state, regional, and local jurisdictions for the implementation and maintenance of the ICM project
- Definition of ICM goals, concept of operations, and the requirements for the ICM deployment.
- Organizational changes required for the implementation and operation of an ICM project
- Identification and selection of strategies for routing traffic on alternative routes and optimizing capacity on those routes for the rerouted traffic flow while still minimizing negative impacts on cross traffic flow using changes in traffic signal timing and ramp metering rates.

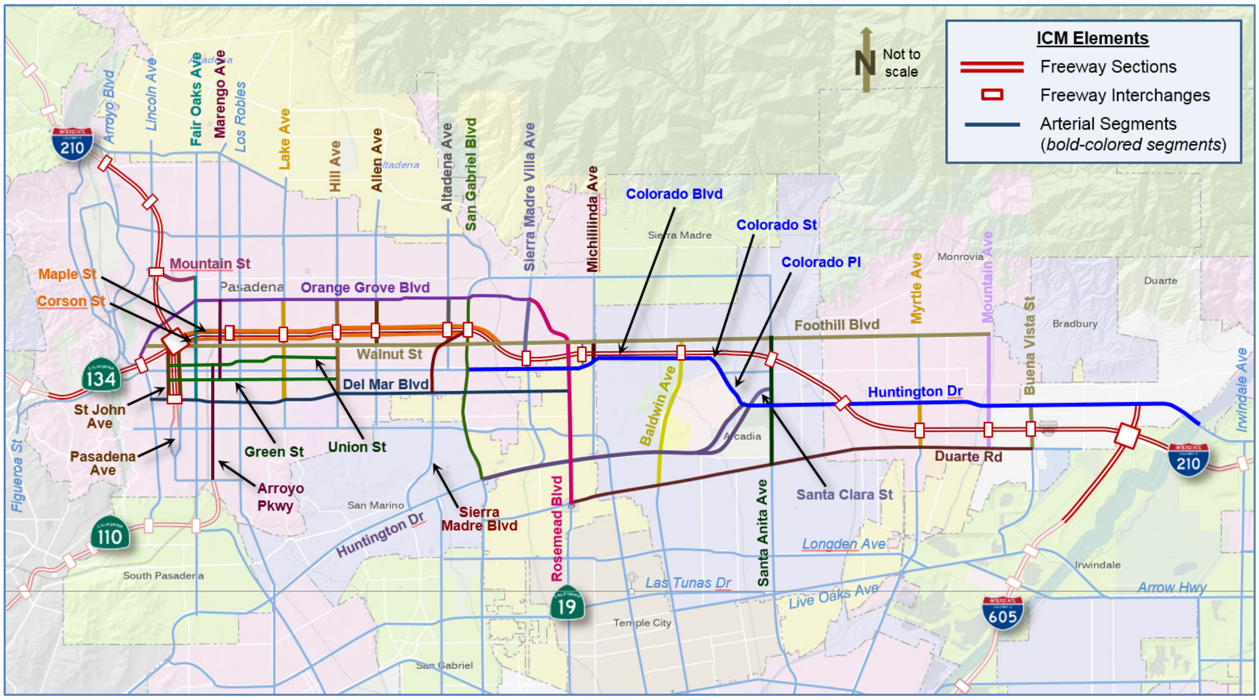

Figure 3-1 Connected Corridors I-210 Corridor Area

Figure 3-1 provides a view of the I-210 Corridor. The corridor encompasses four cities (Arcadia, Pasadena, Monrovia, Duarte), LA County, and Caltrans District 7 as the primary local jurisdictions with significant support and contributions from LA Metro, the Regional Integration of Intelligent Transportation Systems (RIITS), the Southern California Association of Governments (SCAG), the San Gabriel Valley Council of Governments, Foothill Transit, and UC Berkeley PATH.

The ICM system developed by PATH receives information and requests execution of traffic asset control commands from/to each of the primary local jurisdictions as listed in Table 3-1. In addition, supporting TMDD dialogs including the center active and organization dialogs are used to compliment the primary data exchanges. Full details are available in the System Interface Design Specification in the Documentation Library available on the main program website (https://connected-corridors.berkeley.edu/resources/document-library).

Table 3-1 I-210 Primary TMDD Interfaces

Jurisdiction	Asset Type	Information Details	Source System and Vendor
Arcadia	Intersection signals	Inventory Signal Plans Signal Plan Schedules Signal Status Intersection Signal Control	Transcore TransSuite®

Jurisdiction	Asset Type	Information Details	Source System and Vendor
	Intersection detectors	Inventory Status Data	Transcore TransSuite®
Pasadena	Intersection signals and sections	Inventory Signal Plans Signal Plan Schedules Signal Status Intersection Signal Control	McCain Transparity®
	Intersection detectors	Inventory Status Data	McCain Transparity®
	Dynamic message signs	Inventory Status Priority Queue Message Inventory Schedule Sign Control	Ledstar
LA County (includes Monrovia and Duarte)	Intersection signals	Inventory Signal Plans Signal Plan Schedules Intersection Signal Control	Kimley Horn KITS®
	Intersection detectors	Inventory Status Data	Kimley Horn KITS®
	Dynamic message signs	Inventory Status Priority Queue Message Inventory Schedule Sign Control	Ledstar
Caltrans District 7	Ramp Meters	Inventory Status Ramp Control	Parsons ATMS
	Freeway detectors	Inventory Status Data	Parsons ATMS
	Intersection signals	Inventory Signal Plans Signal Plan Schedules Signal Status	Transcore TransSuite®

Jurisdiction	Asset Type	Information Details	Source System and Vendor
		Intersection Signal Control	
	Freeway changeable message signs	Inventory Status Sign Control	Parsons ATMS
	Arterial dynamic message signs	Inventory Status Priority Queue Message Inventory Schedule Sign Control	Parsons ATMS
	Events	Event and event updates Response plans* Response plan approval and approval results* Response event logs*	Parsons ATMS
	Lane closure systems (2)	Lane closure schedule Lane closure inventory Lane closure status Lane closure control	CT Custom Application

*Not TMDD dialogs, but based on TMDD communication methods and data structures

The ICM system developed is composed of three primary subsystems: data hub, decision support system (DSS), and corridor management system (CMS). The TMDD interfaces to each of the external systems listed in Table 3-1 exist within the data hub for data ingestion and within the CMS for control commands. Primary data flow is described in Figure 3-2. Data flows from the source TMC systems that control and monitor field elements (such as traffic signals or ramps) to the ICM data hub. The data hub provides this information to the other two subsystems, the DSS and CMS. Commands required to implement response plans in response to traffic incidents and events are sent from the CMS to the same TMC systems as illustrated on the right side of Figure 3-2. In this diagram, all of the data flows represented by the green arrows between the TMC systems and the data hub and between the CMS and the TMC systems use a TMDD interface. Information flows exchanged between the data hub, CMS, and DSS do not use a strict TMDD interface, as they are neither SOAP, nor XML based, but do implement the same or similar data structures as the TMDD interfaces with the external TMCs.

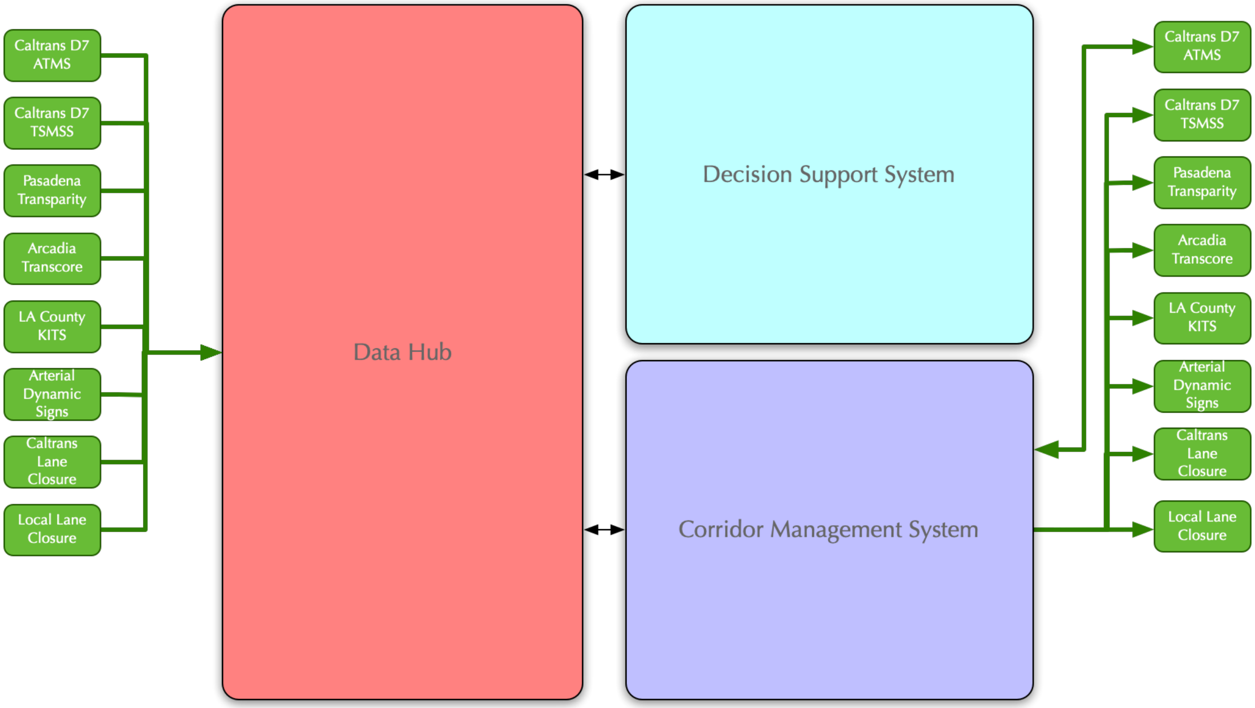

Figure 3-2 I-210 Connected Corridors ICM Data Flows

These TMDD interfaces in the data hub are implemented for all of the dialogs and for all of the vendors through a single service implementation. The data hub uses a micro-service design, and the TMDD interface is implemented within a “hub soap reader” service that provides a SOAP implementation using the I-210 Connected Corridors System Interface Design Specification and accompanying XSD and WSDL. Vendor schema differences required by the vendor implementation differences are maintained within the <any> schema extensions within TMDD. A separate XSD, shared between the vendors, maintains the implementation of the vendor differences. In this way, each vendor’s specifics do not conflict with the other vendor implementations, since all differences are maintained within optional schema elements. By doing this, the same “hub soap reader” implementation can be used across all vendors, and as a result, the “hub soap reader” services are clustered within the data hub and can be easily scaled as additional corridor field elements or additional jurisdictions are added to the project. The command interfaces on the right side of Figure 3-2 are managed by the CMS.

By using the TMDD data structures for internal ICM subsystem communications (data hub -> DSS and data hub -> CMS interfaces) and a publication-subscription messaging design for these interfaces, an important design objective, the ability to use alternative DSS and CMS systems is achieved. TMDD and a common data schema and semantics have played a critical role in making this possible.

3.3. SUCCESSES

The Connected Corridors program achieved several key successes with regards to TMDD and its implementation. Most importantly, it has resulted in a successful, multi-datatype, multi-vendor, multi-jurisdiction implementation of the TMDD standard of which there are few such implementations in existence. While not a strict implementation of the standard, it also has resulted in several recommendations for improvement of the standard that have been shared with the project's sponsor and with the ITE committee for the TMDD standard.

In addition, it has made the implementation of a multi-jurisdictional Integrated Corridor Management system possible. Without a common data standard and a shared implementation of that standard across vendors, the project would be more difficult and significantly more costly.

3.4. OVERVIEW OF IMPLEMENTATION ISSUES

Implementation of the standard has presented various issues. We will discuss and provide examples of some of the most difficult implementation issues, including:

- Missing data structures, lack of support for current and future transportation demands
- Limited implementation guidance and system behavior differences
- Limited communication management
- Limited extensibility (this has been partially addressed in v3.1 of the specification)
- Technology limitations

3.4.1. MISSING ELEMENTS

One of the key issues identified within the I-210 project was that TMDD did not address many of the requirements of the project. In many cases, that was a reasonable expectation. TMDD certainly should not address communication needs for every transportation topic. For example, TMDD does not address transit related information, nor should it, as there are existing standards for those needs that address their specific requirements.

However, there were project requirements that should reasonably be addressed by TMDD. Two key types of missing elements were identified in the project:

- Missing data objects as a result of TMDD Volume 1 requirements not addressing the current traffic management environment
- Other missing elements resulting from TMDD Volume 2 not addressing current traffic control asset capabilities

3.4.1.1.Missing Elements for the Current Traffic Management Environment

The current user needs and requirements listed in TMDD Volume 1 covered the majority of items required by the I-210 Connected Corridors program. However, there were some key user needs and requirements that were not addressed by TMDD. In particular, there were key elements not addressed that are required within current and near-term future traffic management needs expressed by Traffic Management Centers.

3.4.1.1.1. Response Plans

A critical missing user need is the need to support regional cooperative actions and, in particular for the I-210 project and integrated corridor management and TSMO operations, response plans. Response plans for events and incidents are a primary feature of integrated corridor management. This includes a base set of user needs that would include:

- An owner center shall send an incident response plan to an external center.
- An owner center shall send approval or disapproval of an incident response plan to an external center.
- An owner center shall send the results of cooperative jurisdiction approval or disapproval of an incident response plan to an external center.

3.4.1.1.2. Multi-Party Communications

Such user needs require actions that go beyond TMDD's owner center/external center two-party communication mechanisms. Figure 3-3 provides an illustration of this two-party communication. An external center (an external software system, often outside of the jurisdiction of the owner) communicates with an owner center, either requesting information or requesting execution of an action on an owner center's device. The owner center's communication consists generally of replying to requests for information or confirming receipt of the command execution request. TMDD's dialogs are based on these two-party exchanges. Parties may exchange roles as well within TMDD as owner center and external center. In fact, this role exchange is required for parties to have reciprocal information sharing agreements.

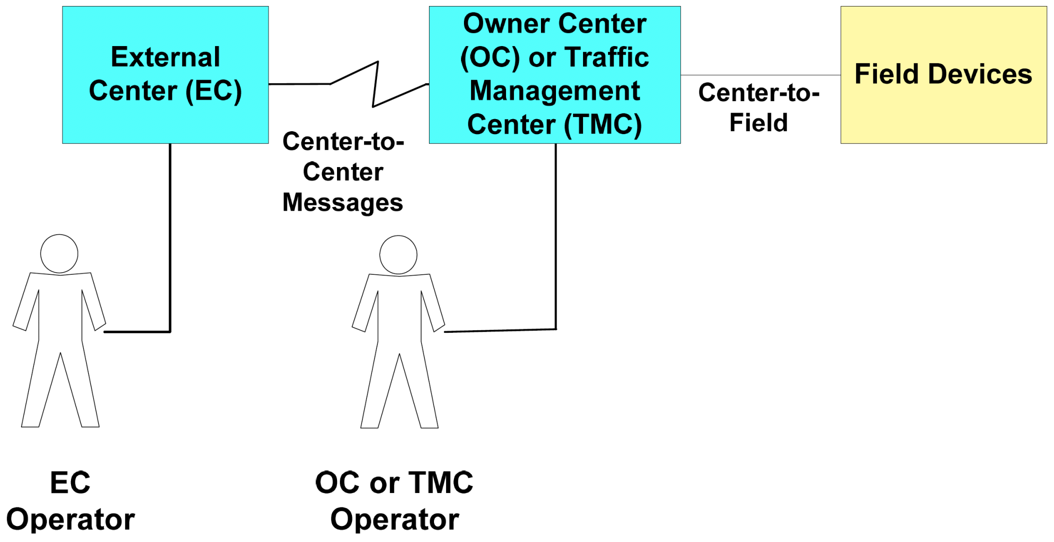

Figure 3-3 TMDDs Two Party Communication Mechanism

Much can be constructed with these two-party communications, but regional operations require additional coordinated messages between multiple parties. Consider the example in Figure 3-4.

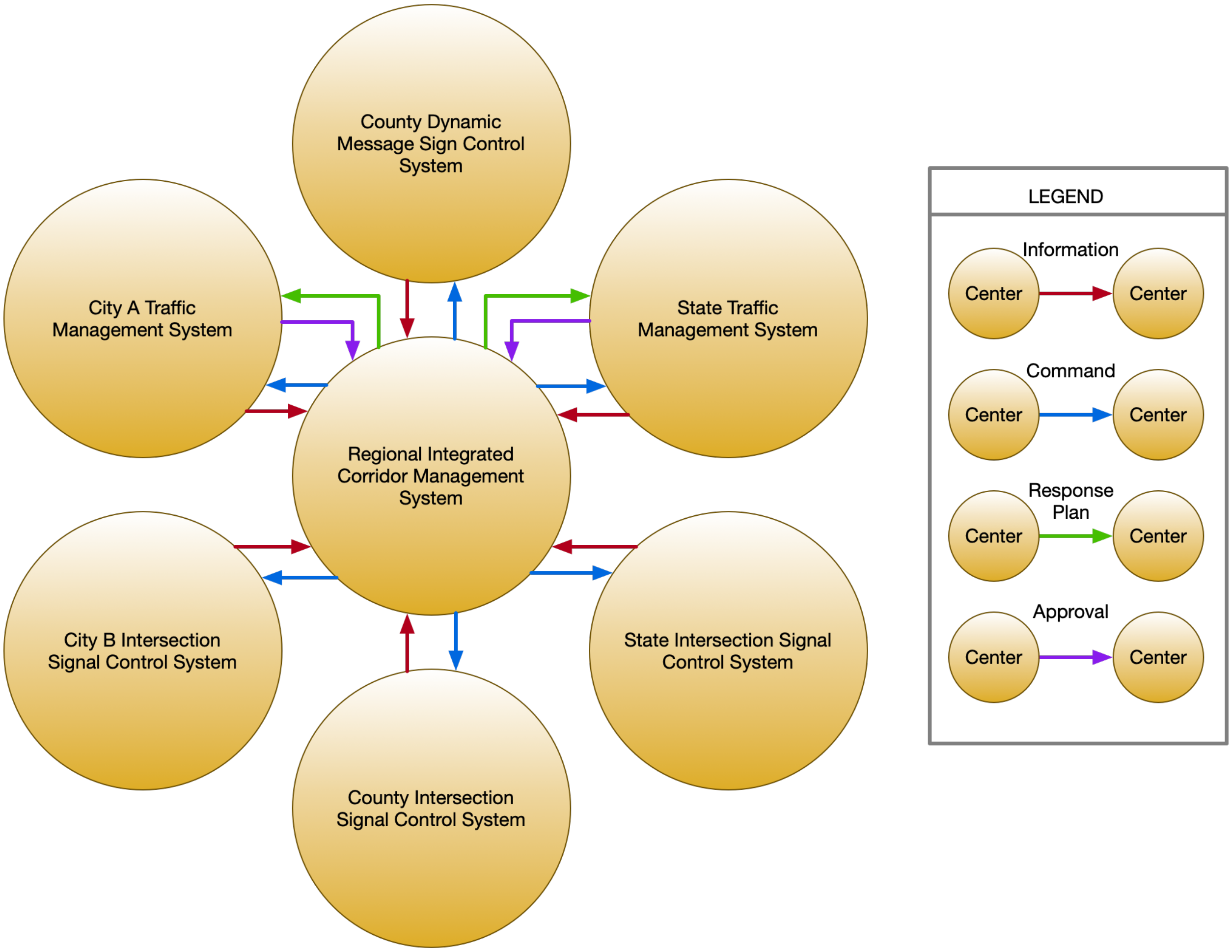

Figure 3-4 Example Multi-Party Communication

In this example, similar to the I-210 Connected Corridors situation, multiple organizations (local, county, and state) each with multiple systems/centers operate together via a regional Integrated Corridor Management (ICM) System. When a traffic incident occurs, the regions organizations and jurisdictions coordinate actions to limit the impact of that traffic incident via response plans generated by the ICM system. TMDD is used to communicate between the various centers, gathering information on transportation asset state such as inventories and state for traffic signals and dynamic message signs (illustrated by the red arrows). TMDD is also used to issue a request to execute commands that are required to implement a response plan. These command requests are issued by the regional ICM system to each of the owner centers (illustrated by the blue arrows). These communications are typical of the two-party communication system available within TMDD.

However, the development, review, approval/disapproval, and information regarding the response plan itself is a regionally coordinated action. TMDD does contain the idea of a response plan, but limited only to locally implemented or internal response plans (Section 2.3.3.2 TMDD V3.1, Volume 1, pg. 18) and limited to exchange of response plan identifiers only within its dialogs and messages. Regional multi-jurisdictional cooperation requires regional content within the response plan, sharing response plan elements that span all of the jurisdictions involved within a response plan. It also generally involves the approval of all jurisdictions involved prior to implementation. All of the parties need to know the traffic routing to be used, intersection signal plan changes, sign displays, the results of the approval process, and other details. This involves sharing information across multiple jurisdictions and coordination across multiple centers within those jurisdictions. While each individual information exchange continues a two-party exchange, the contents within the messages exchanged include information across all jurisdictions. For the response plan, the intersection signal plan changes for all jurisdictions are included. For the state freeway system, all ramp meter changes are shared with all jurisdictions, include the cities and county. Approvals are requested from all jurisdictions involved, and the results of those approval decisions, upon completion, are shared with all jurisdictions. Jurisdictions that have traffic management systems can use those systems to receive, visualize, and approve response plans. Those traffic management systems then also need to be informed of those response plans. The content of those messages contains information related to multiple organizations and centers rather than limited to the sender and receiver.

In a hub and spoke configuration, with all of the shared information at the hub as shown in Figure 3-4, the two party communication paradigm works, since no spoke requires sharing information with the other spokes. The district ATMS does not require knowledge of the county intersection signal system, since all actions requiring a common set of information are maintained at the hub. However, once information such as a response plan is shared with one of the spokes in the hub-spoke architecture, there is the possibility that the spoke receiving the response plan now needs inventory, state, or other information from the other spoke systems. This in fact is the case with the I-210 implementation, as the D7 ATMS receives response plans with all of the information required to evaluate and approve the plan contained within the plan, but it has no supporting inventory or status for any of the corridor assets owned by the other centers. As a result, it has no ability to give an operator full situational awareness of the corridor, such as arterial traffic capacity, intersection signal state, or other critical information. In effect, the district TMC operators are blind to anything happening in the corridor beyond what is managed within their own TMC. Costs to implement a full suite of not only TMDD interfaces to exchange information, but also to operationalize that information within the ATMS would be excessive. Individuals responsible for system architectures and requirements should keep this in mind when defining complex multi-jurisdictional, multi-vendor implementations.

3.4.1.1.3. Organization and Originating Systems Distinction and Implementation Guidance

TMDD specifies a method for exchanging organization information. This is accomplished via the Organization Class Dialogs including dlOrganizationInformationRequest, dlOrganizationInformationSubscription, and dlOrganizationInformationUpdate. This information can then be used to identify owning organizations for information provided regarding assets in other dialogs, such as data, inventory, status and other messages.

Organization information can contain information regarding the centers of the organization. Implementations of these structures on a scale larger than two exchanging parties must take into account what are often complex relationships between organizations and their structures and the different ways in which systems are used and allocated within the different organizations. TMDD is limited in its guidance regarding implementations and how this information is exchanged.

Within multi-party exchanges and when systems are shared across organizations, this limited implementation guidance, and the limitations within the structure, can result in unnecessary complexity within the sending and receiving systems. Consider the following use case:

Four cities, a county, and a Caltrans district share an ICM system dedicated to a specific set of limited transportation corridors. This ICM system is one of several maintained by the state across as many districts. These four cities, the district, and the county all are part of a larger effort to share not only the information related to the corridors, but a larger set of information, similar, but different from that set of information shared within as part of the ICM effort.

TMDD shares organization information, and provides a method via the organization and center identifiers to identify jurisdictional ownership of transportation field elements as well as the systems of record and control for those field elements. In order for this to be effective however, the sharing entities must agree on unique organization and center identifiers across those sharing information. This may sound simple for the first time this occurs, but as the circle of sharing entities expand, the opportunity for two organizations or centers to have identical ids becomes more likely. In practice, when systems that are shared or consolidate multiple organizations' information are first put in place, they are likely to encounter organizations or centers that share the same unique identifier.

The result is that systems have to be engineered to address the possibility for different organizations or centers to share the same unique identifiers. To avoid this, a centralized identifier registration system at the state level for organizations and centers would be beneficial.

3.4.1.1.4. Connection Management Requirements

TMDD has very limited connection management capabilities. In general, volume one (Section 3.3.1, pgs 37-39) limits connection management capabilities to exchange of the center active messages to verify the center is responding and the exchange of request response or subscription messages.

In practice however, additional connection management capabilities are required to manage the communications, particularly when subscriptions are used and communications are asynchronous in nature.

Consider the following use case: two systems are designed to exchange information via an on-change subscription. This on-change subscription results in infrequent updates, often on the order of a week or more between updates. This is likely the case with inventory on-change or periodic subscriptions. The two systems establish the subscription (identified with a subscription id of 3) with system 1 being the owner center and system 2 the external center. System 1 begins the subscription data exchange with the first update, system 2 receives the update, and communication is established. System 1 and 2 now have an established subscription with subscription id = 3 and data is flowing.

However, system 1, the owner center has a significant failure. It is also not designed to recover state of any existing subscriptions when it resumes operation. As a result, system 1 is no longer aware of, nor providing data for subscription id 3. However, system 2 is not aware of system 1's failure and still believes subscription id 3 is still valid and operating. It will never receive another data update from this subscription and is not aware that it should delete subscription id 3 from its list of existing subscriptions. Nor does it know it should reestablish the data exchange with a new subscription request. Since the data is infrequently updated, it is possible to miss this error condition for an extended period of time, perhaps weeks or longer.

To remedy this situation, TMDD needs additional connection management information exchange, such as the ability to report system restarts or failures, request and receive subscription status information for all existing subscriptions or specific subscriptions, or other types of connection management information exchanges for other possible system data exchange or system failure modes.

3.4.1.1.5. Other Missing Elements

Other missing elements that were discovered during the I-210 project include:

- Non NTCIP device information. TMDD dataframes and data elements are very NTCIP device oriented. Legacy non-NTCIP devices or other devices that deviate from NTCIP due to local standards or because of advanced technical capabilities cannot be adequately represented within its data structures. Often additional information is needed to describe the device state or inventory elements that are not available within TMDD. TMDD has extensibility to address much of this, and that extensibility has been greatly improved in TMDD v3.1. An example of this limitation is the description of intersection state within TMDD for AB3418E intersection signal controllers.
- TMDD has no way to associate the data received for a detector associated with an intersection signal controller with the inventory of the detectors associated with the same intersection signal controllers. In the I-210 project, TMDD's extension capabilities were used to add a detector index with the device.

The I-210 extensions used to address these and other issues are fully documented within the I-210 Systems Interface Design Document[7] and associated xsds.

3.4.2. IMPLEMENTATION GUIDANCE AND STANDARDIZATION

One of the most critical implementation issues for TMDD is limited implementation guidance. TMDD is very flexible and is built with that flexibility in mind in order to allow its use to be tailored to local project or program requirements. Volume 1's purpose is to provide some guidance in generating local requirements and guiding its implementation in a specific project.

However, with flexibility comes complexity. This complexity is not always evident within the specification. Here are some examples of some of the complex issues that must be solved and are not easily understood without implementation guidance or experience.

3.4.2.1. Define Dialog Behaviors

TMDD provides guidance regarding the dialogs available for implementation, the messages to be exchanged within those dialogs, the structure of information within those messages, data types, and some limited information regarding the definitions of the various elements. In general, it defines what the information is that can be exchanged and the format of the exchange, but nothing regarding the temporal nature of that exchange of information.

3.4.2.1.1. Guidance Regarding Type of Information Exchange

TMDD provides for synchronous request-response information exchange and a semi-asynchronous exchange utilizing multiple synchronous message exchanges (subscriptions). However, no guidance is given regarding which type of information exchange is best or which types should be supported within a given project or under specific conditions.

Let's consider the example of intersection signal status updates. Intersection signal status can be gathered using any of the available exchange types: request-response, one-time subscription, periodic subscription, or on-change subscription. All are valid within the specification. Which you should choose is very dependent upon the purpose of the information exchange.

If the purpose of intersection signal status information is to determine a snapshot of the signals current operational state (working or not working, perhaps current plan), then request-response or a one-time subscription are reasonable choices for communication. If the purpose is to maintain a list of the current signal plans in operation over the course of a day, then an on-change subscription is appropriate. A periodic subscription might be appropriate with a sufficient frequency of update as well. If the purpose is to maintain the list of every phase change during the course of a day, then an on-change subscription is the only appropriate choice. Even then, it is likely that some changes may be missed if the update frequency from the controller in the field is insufficient.

Time domain of the update frequency and time domain of the requirement are critical to information exchange method chosen. TMDD does not provide guidance regarding time domain issues.

3.4.2.1.2. Communication Startup Behavior

Communication startup behavior should be standardized in order to make different implementations of the standard capable of communicating across the industry. This issue can be illustrated easily within status or inventory dialogs.

Consider a detector status on-change subscription. In this case, the external center has requested that a message be sent every time the status of a detector has changed. In general, owner center updates in response to this subscription request are limited to state changes between operating and failed. With an on-change subscription, only when the state has changed between operating and failed will a message be sent. Without a known initial state of operating or failed for each sensor, the external center cannot know the operating states of any of the sensors from the updates alone, except for those that have changed since the subscription was started.

There are at least two possible methods of addressing this issue. First, the external center could issue a request via a request/response dialog for detector status to get a full status report followed by the subscription. Another method would be to ensure that the first message in the on-change subscription update contains the status of all detectors. There is little difference in these methods other than overhead of the communications involved, however, the first solution does present the unlikely possibility that the status change of a detector is missed between responses (if the status changes in between the request/response update and the subscription start). Starting the subscription prior to the request/response would eliminate this possibility, but would require the external center to ensure the order of messages processed was correct by validating the time of the updates.

For the purely subscription-based solution to work however, the behavior of ensuring the first message contains the status of all detectors must be specified. TMDD does not address this type of behavior and does not standardize content of messages beyond their structure. Without this standardization, different implementations of the standard are incompatible, resulting in critical standardization lapses and additional costs when moving solutions between projects. This results in implementation silos, even when compliant with TMDD.

Inventory messages are another common behavior issue. Consider an inventory message, particularly an on-change inventory message. In general, a request/response type of inventory exchange results in a description of a full inventory within a response to the external center. However, what is the expected behavior within an on-change subscription? If the expected behavior is that only changes be sent within the on-change subscription, there are two issues. First, the same issue as described above with the detector status on-change subscription exists. The first message would need to contain a full inventory. Second, if a new item is added, there would be a new inventory item in the record list. Not having a record of that within the external center, it could reasonably be assumed that it has been added. However, there is no way within the message to indicate removal of an inventory item. As a result, the inventory can grow, but assets can never be removed from the inventory. There are no fields within a record to indicate when a device is taken out of the inventory, and there is no action field to indicate within the inventory record whether the external center should add, update, or delete the item from the inventory. The only behavior that would work in an inventory on-change subscription is to always include the full inventory, replacing the previous inventory reported. Again, without guidance, it is up to the implementer to decide how to address this issue. While the I-210 project specified that all inventory updates would include a full inventory for each of the vendor solutions integrated within the project, another project could instead add a standard extension that would add an action (update, delete, add), or could create an extension that gave a date the asset was added to the inventory, and a date in which it was removed. An empty removal date would indicate the asset is still in service. Again, this results in a lack of standardization within the implementations across vendors and across projects, with added cost for each project, added maintenance of multiple implementations within vendor solutions, and a creation of information silos across the state and nation.

3.4.2.1.3. On-change Triggers

For on-change subscriptions, one key element left to the implementer is the identification of what constitutes a change that will trigger an update from the owner center to external center. While it could be decided that if any field within the message contents changes, an update would be triggered, this policy has practical and technical limitations.

From a practical perspective, not every field should be considered with the same importance by the receiving system. Increasing the sending and receiving system load for changes that have little importance is wasteful of computing resources, processing, and money. It is dependent upon the need of the receiver of the information and their system requirements to determine what should trigger an on-change event. For example, if using the intersection signal status update to determine if a signal is operable or not, then what should trigger an update message, at a minimum, is the change in signal operating status, generally via device-status and device-comm-status fields within the device header. If using the intersection signal status to understand actual cycle length and recreate green times, then the trigger required is much more complex, ensuring that a full time-series of the phase changes can be constructed by a series of update messages.

Technical limitations also exist within owner and external center systems, especially in smaller local installations, but often larger installations as well. The obvious technical limitation is that each system, sender and receiver must be capable of the performance required to generate, send, receive, and process the volume and size of messages generated based on the on-change trigger defined. The use of SOAP is considered much less performant than many newer technologies available and creates heavier computing burdens on installed systems, especially when an owner center may serve multiple external centers and when information exchange demands high update frequencies, such as with a large number of intersections and high levels of information detail (such as capturing each light change in a status message). Differences in center system architectures (owner or external) can limit system performance, as can supporting hardware and infrastructure. These limitations can provide further justification to limit what is considered a trigger for an on-change event. In general, systems use some sort of time-based check of what has changed in a supporting database, database triggers, or a combination of the two with a mechanism to build a queue of changes that must be reported, even for on-change subscriptions. Event based on-change subscriptions are rarely truly event based. These mechanisms, along with single instance relational databases, can limit performance, especially in cases of large stored data volumes and frequent data read and write volumes.

As a result, it is critical to define what triggers a change event for an on-change subscription. Project requirements and center limitations must be understood to correctly define such event triggers.

3.4.2.2. Enumeration Definition and Message Content Usage

Projects must define how the data content within messages will be used and there is little guidance in the standard beyond basic definitions and often no guidance into how to standardize implementations. As a result, TMDD does fairly well with point-to-point communications with customization of implementation details within C2C implementations, but these can rarely be extended to other locations or within multi-vendor C2C implementations. A couple of simple examples are provided to illustrate this issue below.

3.4.2.2.1. Event Location Choice

Event location within TMDD has several options (listed as <xs:choice> within Volume 2, 3.3.8.21 eventLocation (pg. 160)) to specify where an event has occurred. This allows the implementing systems to choose how the event location is to be described. These choices within TMDD allow for the specification of the location as one of the following:

- area-location, type = AreaLocation
- location-on-link, type = LinkLocation
- landmark, type = LandmarkLocation
- geo-location, type = lrms:GeoLocation
- other (used as a TMDD extension)

However, for two systems to exchange information, they must either understand and use all of the specified methods or must agree to which are used. There are two critical elements to distinguish in that statement: a) understand all of the specified methods, and, b) use all of the specified methods.

Understanding the specified methods requires that both systems allow each of the specific choices to be used. Generally, two systems designers would agree within a specification which choices would be used, and possibly under what conditions.

In the case of the I-210 project, after some effort, the location-on-link or landmark structures were specified, depending upon where the event occurred. An optional geolocation (latitude/longitude) was allowed. Location-on-link, specifying the linear post-mile as a linear reference, along with an optionally specified latitude and longitude provided freeway event locations. Arterial incident locations were specified using either the intersection where the event is located using the TMDD structure for a landmark dataframe (using a node-id for a defined network structure), or again using a location-on-link dataframe specifying the link and a pointonlink geolocation (optional).

In this case, the number of options available within TMDD work against a desire to facilitate a ubiquitous solution across systems in different jurisdictions and vendor implementations.

3.4.2.2.2. Lane Status Within an Event Example

Lane-status is an ITIS.Closures type. While this is not a TMDD type, its use is critical to describe lanes closed for an incident or event, and agreement between systems on how it is implemented is also critical to the ability of sender and receiver to understand the incident or event. Again, systems must specify how they will implement the enumerations provided for within the specification. In a multi-jurisdictional, multi-vendor environment, all systems must use the field and enumerations in a common way or be able to, at a minimum, accommodate the various selections used by all centers.

The enumerations available within lane-status are:

- closed-to-traffic
- closed
- closed-ahead
- closed-intermittently
- closed-for-repairs
- closed-for-the-season
- blocked
- blocked-ahead
- reduced-to-one-lane
- reduced-to-two-lanes
- reduced-to-three-lanes
- collapse
- out
- open-to-traffic
- open

- reopened-to-traffic
- clearing
- cleared-from-road

There is little guidance in how to use these enumerations, and multiple enumerations can easily be used to describe the same lane status. For example, closed, closed-ahead, closed-to-traffic, blocked, blocked-ahead, reduced-to-X-lane(s), out, and clearing could all be used to describe an incident that blocks a lane of traffic. For systems that act as sending and receiving centers, it is critical that any pair of systems agree on the enumerations used and under which circumstance in order for them to operate. This is especially critical if the information is not only displayed to a human, but also used within the logic of the receiving system. As systems become more complex in how they use information and include their usage in complex algorithms and automated decision making, as the Connected Corridors program does, standardization of not only the enumerations, but their usage is critical. Doing so requires additional guidance in how to use the enumerations, and in some cases, more careful selection of the enumerations available. Without such guidance and usage standardization, systems used in TMCs will remained siloed and unable to communicate without expensive modifications. Solutions that can be deployed in a more off-the-shelf manner in any location will not be possible, and extensive customization within each deployment will continue to be required. The ability to extend cooperation between jurisdictions and across extended geographic boundaries will be hindered at best.

3.4.2.2.3. Event-lanes example

Another example where projects must define specific implementation detail is in the message content and how the message information is structured. There are very frequently different ways in which a specific situation may be described within the message content and how that content uses the structures available in TMDD, as well as the enumerations usage for the same meaning. While this provides great flexibility, it also provides increased complexity when designing, developing, and implementing systems to use this information in a programmatic way.

```
<event-lanes>
                <event-lane>
                  <lanes-type>through lanes</lanes-type>
                  <link-direction>e</link-direction>
                  <lanes-total-original>5</lanes-total-original>
                  <lanes-total-affected>5</lanes-total-affected>
                  <event-lanes-affected>
                    <lanes>1</lanes>
                    <lanes>2</lanes>
                    <lanes>3</lanes>
                    <lanes>4</lanes>
                    <lanes>5</lanes>
                  </event-lanes-affected>
                  <lanes-status>closed</lanes-status>
                </event-lane>
                <event-lane>
                  <lanes-type>left shoulder</lanes-type>
                  <link-direction>e</link-direction>
                  <lanes-total-original>1</lanes-total-original>
                  <lanes-total-affected>0</lanes-total-affected>
                </event-lane>
                <event-lane>
                  <lanes-type>right shoulder</lanes-type>
                  <link-direction>e</link-direction>
                  <lanes-total-original>1</lanes-total-original>
                  <lanes-total-affected>0</lanes-total-affected>
                </event-lane>
                <event-lane>
                  <lanes-type>hOV lanes</lanes-type>
                  <link-direction>e</link-direction>
                  <lanes-total-original>1</lanes-total-original>
                  <lanes-total-affected>0</lanes-total-affected>
                </event-lane>
</event-lanes>
```

Figure 3-5 Event-lanes Example 1

We will use the event-lanes structure of the TMDD standard to illustrate the issue. Figure 3-5 provides an example of usage from the Connected Corridors program. In this example, the event-lanes message shown in the figure describes a freeway event lane blockage, where there are five through lanes, one HOV lane, a right shoulder and a left shoulder. The message indicates that all five through lanes are closed, while the HOV lane and both shoulders are currently open. This would indicate that the five primary lanes are not available for traffic. No other information is currently listed. Some systems could be designed to provide additional information, such as a distinction between closed and blocked. Without some standardization of usage of lane-status as described in section 3.4.2.2.2, it is likely that two systems, while able to exchange the message correctly, may not understand the meaning of the message within a specific algorithm without some modification to one or both systems.

```
<event-lanes>
        <event-lane>
                <lanes-type>through lanes</lanes-type>
                <link-direction>e</link-direction>
                <lanes-total-original>5</lanes-total-original>
                <lanes-total-affected>2</lanes-total-affected>
                <event-lanes-affected>
                        <lanes>1</lanes>
                        <lanes>2</lanes>
                </event-lanes-affected>
                <lanes-status>open</lanes-status>
        </event-lane>
        <event-lane>
                <lanes-type>through lanes</lanes-type>
                <link-direction>e</link-direction>
                <lanes-total-original>5</lanes-total-original>
                <lanes-total-affected>3</lanes-total-affected>
                <event-lanes-affected>
                        <lanes>3</lanes>
                        <lanes>4</lanes>
                        <lanes>5</lanes>
                </event-lanes-affected>
                <lanes-status>closed</lanes-status>
        </event-lane>
        <event-lane>
                <lanes-type>left shoulder</lanes-type>
                <link-direction>e</link-direction>
                <lanes-total-original>1</lanes-total-original>
                <lanes-total-affected>0</lanes-total-affected>
        </event-lane>
        <event-lane>
                <lanes-type>right shoulder</lanes-type>
                <link-direction>e</link-direction>
                <lanes-total-original>1</lanes-total-original>
                <lanes-total-affected>0</lanes-total-affected>
        </event-lane>
        <event-lane>
                <lanes-type>hOV lanes</lanes-type>
                <link-direction>e</link-direction>
                <lanes-total-original>1</lanes-total-original>
                <lanes-total-affected>0</lanes-total-affected>
        </event-lane>
</event-lanes>
```

Figure 3-6 Event-lanes Example 2

```
<event-lanes>
        <event-lane>
                <lanes-type>through lanes</lanes-type>
                <link-direction>e</link-direction>
                <lanes-total-original>5</lanes-total-original>
                <lanes-total-affected>3</lanes-total-affected>
                <event-lanes-affected>
                        <lanes>3</lanes>
                        <lanes>4</lanes>
                        <lanes>5</lanes>
                </event-lanes-affected>
                <lanes-status>closed</lanes-status>
        </event-lane>
        <event-lane>
                <lanes-type>left shoulder</lanes-type>
                <link-direction>e</link-direction>
                <lanes-total-original>1</lanes-total-original>
                <lanes-total-affected>0</lanes-total-affected>
        </event-lane>
        <event-lane>
                <lanes-type>right shoulder</lanes-type>
                <link-direction>e</link-direction>
                <lanes-total-original>1</lanes-total-original>
                <lanes-total-affected>0</lanes-total-affected>
        </event-lane>
        <event-lane>
                <lanes-type>hOV lanes</lanes-type>
                <link-direction>e</link-direction>
                <lanes-total-original>1</lanes-total-original>
                <lanes-total-affected>0</lanes-total-affected>
        </event-lane>
</event-lanes>
```

Figure 3-7 Event-lanes Example 3

Now let's look at two more examples. Figure 3-6 and Figure 3-7 illustrate a similar example with the same lane configuration, but in both examples, lanes 1 and 2 are not closed or blocked. The descriptions of the lane status for the freeway lane configuration are identical, but the message structures used to describe this status are different. Figure 3-6 is explicit that lanes 1 and 2 are open, where the lane status of lanes 1 and 2 being open is implicit in Figure 3-7.

Coupled with the possibility of having multiple systems with different usage of the lane-status field such as the use of "closed" vs. "blocked" vs. "blocked ahead" or "open" vs. "open-to-traffic" vs. "reopened-to-traffic" vs. "clearing" vs. "cleared-from-road", it is clear that systems that adhere to not only the data structure and content standard, but also a usage standard would be more "plug and play," would require a lesser expense to customize and implement, and are less likely to create “silos” of communication when more than two systems are part of the communication network.

3.4.2.2.4. Enumeration Definition and Content/Structure Utilization Guidance

TMDD is a complex standard trying to solve a complex problem: how to implement a method of communication that enables an intelligent transportation infrastructure by standardizing communications between traffic management centers. To do this, it provides a standard for information exchange, specifying the available exchanges between systems, their structure, and contents. It leaves the system designers that implement the standard great flexibility in the specific information exchanged and how meaning is defined and derived from the structure and content of a message. In doing so however, the end result is that each system uses a specific designer’s implementation which can only communicate with other systems with the same designer’s implementation. Extending these systems to communicate with other implementations requires extensive modification of one or both systems. Extending these implementations to a region or state with many systems becomes incredibly complex at a technical and program/project level, as well as very expensive. What is needed to minimize this complexity is a standard method of implementation and implementation guidance to assist designers in meeting the standard. This is not dissimilar to other industry standards attempts. Even the SOAP standard used by TMDD underwent a similar change with the implementation of the Web Services Interoperability agreements following the release of the SOAP standard.

3.4.2.3.Temporal Dissonance Between TMCS and Field Elements

Within a regional implementation of communication between multiple jurisdictions there can be temporal dissonance between traffic management centers and the field elements they individually control. Some examples that may exist include:

- Field intersection signal controller data polling frequency differences between jurisdictions
- Detector reporting frequency differences (30 sec vs 60 sec)
- TMDD implementation differences (behavior or subscription type changes such as on-change vs. periodic differences between jurisdictions)
- TMC system architectures and capabilities

As a result of these and other real-world issues, the implications for sharing data between systems and any programmatic use of that information that was shared can create issues. Examples of the issues that can result include:

- Limiting use of higher time resolution data from one source when other sources have lower resolutions, resulting in a least-common-denominator approach within algorithms utilizing the data
- Limiting geographic scope of data when high resolution data is not available

- Limiting functionality of systems as a result of the above-mentioned issues

These issues can have multiple potential causes. For field intersection signal controller polling frequency differences some causes of any differences may include:

- Field-to-center network communication limitations
- TMC center software systems or their infrastructure not capable of supporting near real-time operations at high levels of detail
- Field controller systems or firmware capability differences

Detector reporting frequency differences can have similar causes.

TMDD implementations may also cause some reporting differences. As an example, different systems may implement the on-change subscriptions differently. These differences may be a result of limitations of the center system architecture, especially with older system designs. Few, if any, systems have a truly event driven publishing mechanism to support an on-change, event-based publishing mechanism defined within the TMDD specification. TMDD doesn't even provide guidance on what events should drive the publishing of a message. Many systems use a time-based mechanism to poll a database to identify changes to be published to approximate an event-based mechanism. Others may create a queue of changes and publish those on a frequent, but still periodic basis. There is at least one system within the Connected Corridors system that, given a periodic subscription request, publishes on a periodic basis, but only publishes any changes in that period, while others publish a full report on a periodic basis. Others limit what periods can be used in reporting which may be different depending upon specific dialogs and data types. For that same system that publishes only changes in periodic subscriptions, any on-change subscription provides what is essentially the same behavior – changes published on a time-based period. In essence, there is no difference between a periodic or an on-change subscription.

The result of all of these differences in implementation, TMC system capabilities, and field to center differences is that, in a multi-jurisdictional environment, the ability to use all of the data being provided at different time scales can be complex and as a result limited. Standardization in implementations is critical to ensuring the usefulness of the information in future transportation solutions.

3.4.2.4. Implementation of Command and Control Messages

TMDD provides communication not only for the purpose of sharing information, but requesting control of an owner center's transportation assets. This is a critical element for regional cooperative efforts. There is no guidance on when control is requested and once external control of a field element is implemented, how to return control of that field element to local control. Take for example the implementation of a response plan being executed in a multi-jurisdictional environment due to a traffic accident on a state-controlled freeway. The response plan, in the case of the Connected Corridors program, is generated by an Integrated Corridor

Management System, and contains TMDD command messages to be sent to each of the state, regional, or local systems required to implement the response plan. These messages are requesting the owning jurisdictions to implement a control command to their field assets involved in the response plan. Those commands will override the normal time of day or traffic responsive control of the field elements involved and remain in effect until replaced by a subsequent control request. TMDD does not however dictate how control is released by the ICM system back to the local jurisdiction when all current and future response actions have been completed.

Experience in the Connected Corridors program found that vendors either had no solution to return field assets to their normal operation at the time of the completion of the response plan, or required a specific custom command that differed from that of other vendor solutions. As a result, there was no standard implementation of such a usage of the TMDD standard. For the Connected Corridors program, that meant either requesting vendors change their implementation, or customizing return of local control within the ICM system for each individual jurisdiction’s center software.

3.4.2.5. Non-Standard Implementations

Many TMDD implementations are not fully compliant with the standard, including the Connected Corridors implementation. While the Connected Corridors implementation is mostly compliant, the biggest variation from the standard was done to allow use of the standard java libraries for SOAP services rather than implement a significant amount of custom code. Without these changes, the TMDD standard would not compile with the standard libraries, as it is not WS-I compliant.

Other TMDD implementations provided within other vendor’s off-the-shelf software were not fully standard compliant. As a result, these implementations required modification in order to communicate with the ICM system or other systems that use TMDD based SOAP services, even when the Connected Corridors implementation of a specific dialog might be compliant with the TMDD standard.

The result of non-compliant implementations is to increase the complexity and cost of integration between systems with different implementations. Additionally, this increased complexity and cost serves to increase the potential for a certain amount of vendor lock-in for future systems deployment efforts and upgrades. Updating the standard could improve standardization across different vendor solutions.

3.4.2.6. Time Reporting within Messages

In general, the time attributes listed within the standard are self-evident. For example, within the detectorDataDetail dataframe, there are three time-related elements: detection-time-stamp, start-time, and end-time. The definition for these is self-evident. However, usage of these may be non-standard between implementations. In general, for the I-210 Connected

Corridors program, detection-time-stamp is the only field used. Start-time and end-time are not provided. This is reasonable, as both start and end time elements are not required. Usage can, however, become an issue in system integration.

3.4.2.6.1. *Event time*

Event time is described within TMDD in the eventTimes dataframe as shown in Figure 3-8.

```
<xs:complexType name="EventTimes">
    <xs:annotation>
        <xs:documentation>
            <objectClass>Event</objectClass>
                <definition>The information content describing a schedule of start and end times associated with an event.</definition>
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="update-time" type="DateTimeZone"/>
        <xs:element name="valid-period" type="ValidPeriod" minOccurs="0"/>
        <xs:element name="schedule-element-ids" minOccurs="0">
            <xs:complexType>
                <xs:sequence maxOccurs="20">
                        <xs:element name="event-schedule-element-identifier" type="Event-schedule-element-identifier"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="sequence-time" type="DateTimeZone" minOccurs="0"/>
        <xs:element name="start-time" type="DateTimeZone" minOccurs="0"/>
        <xs:element name="alternate-start-time" type="DateTimeZone" minOccurs="0"/>
        <xs:element name="alternate-end-time" type="DateTimeZone" minOccurs="0"/>
        <xs:element name="expected-start-time" type="DateTimeZone" minOccurs="0"/>
        <xs:element name="expected-end-time" type="DateTimeZone" minOccurs="0"/>
        <xs:element name="recurrent-times" minOccurs="0">
            <xs:complexType>
                <xs:sequence maxOccurs="64">
                        <xs:element name="recurrent-time" type="RecurrentTime"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="planned-event-continuous-flag" type="Binary-flag" minOccurs="0"/>
        <xs:element ref="tmddExt:eventTimesExt" minOccurs="1"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

Figure 3-8 eventTimes Dataframe

In this dataframe, the elements of DateTimeZone type are update-time, sequence-time, start-time, alternate-start-time, alternate-end-time, expected-start-time, and expected-end-time. Other times are provided for within some of the other elements via their specific types, such as within the eventTimesExt dataframe's roadway-cleared-time. All, with perhaps the exception of sequence time, are self-evident. However, there are clearly many ways through the times available and the other data elements and their complex types definitions to define the time span of any event. The result of this is a strong likelihood that two TMDD compliant systems will implement different descriptions of how an event's time elements are defined, making them incapable of communicating without some modification to one or both systems.

An additional complexity is introduced by how the times are captured within real TMC operations and the standard operating procedures of that traffic management center. As a real example, during a visit to a traffic management center, interviews with the TMC operators indicated the following regarding capture of traffic incident start times:

- The time of the incident was not always evident in reports from the field. There can be significant variation depending upon who in the field is reporting the incident.
- Operators report different time definitions in the ATMS interface, some entering the event time reported by the field, some entering the current time when they enter the value in the interface, and some reporting the time the report was received from the field.

Such differences in time reporting, coupled with how events and their time attributes are communicated between systems, can result in inconsistencies in incident and event response planning, particularly in a future of complex decision support systems, artificial intelligence/machine learning, and statistical and post incident response analysis.

3.4.3. COMMUNICATION MANAGEMENT

Communication management within TMDD is defined in the User Needs (Section 2.3.1) and Requirements (Section 3.3.1) of volume 1 of the standard and the Connection Management Class Dialogs defined within Section 3.1.3 of volume 2. This set of communication management requirements limits communication management to having:

- Working request/response dialogs
- Working subscriptions of each type
- A center active dialog to ensure communications are established and each center is responding to SOAP calls

This is extremely limiting and does not provide an additional layer of communication management required for real-life operational situations, including:

- Inquiries to verify active subscriptions and their status (dialog, type, duration, source endpoint parameters, target endpoint parameters, remaining time, filters applied, etc.)
- Re-establishing connections when one center goes offline due to system failure, planned or unplanned maintenance
- Notifications of planned or unplanned maintenance
- Recovery of lost or missing data

Experience in the I-210 program has shown that real-life systems need additional capabilities in order to ensure systems can be realistically managed in a production environment. As demands for data increase in the future, these capabilities will require significant enhancement and automation.

Examples of production issues identified during the I-210 program include:

- Incorrect requests for subscription that are not detected by the owner system, resulting in subscription updates sent to the wrong target endpoint
- Subscription cancel requests that are confirmed, but data continues to flow, indicating the owner system acknowledged the request, but did not cancel the subscription
- Differences in how owner systems recover from failure, with some returning with past subscriptions active and others with no subscriptions active
- Differences in how external systems recover from failure, with some re-establishing past subscriptions, and others not re-establishing past subscriptions
- Owner centers that queue up data when failure of an external center is detected and will continue to attempt to send that past data, and others that will not re-attempt to send past data.

Some of these issues are a result of system failures at either the owner or external center. Others are the result of a lack of consistency with how different systems recover from failure. Each system may recover from the failure, but without standardization of how that is achieved, a mismatch between the two systems is likely to affect the communications between those systems. Each of these examples, however, are real life situations that without some standardization, even their discovery can be difficult. Standardization needs to address how the issues are detected, how they are communicated between systems, and what communication is required to automate recovery. Real life demands a certain standardization of failure and non-standard behavior detection and recovery management within the communication standard along with implementation guidance. Standardization of these types of communication management issues would greatly improve the success and reduce the cost of projects and system implementations that use TMDD for center-to-center communications. Such guidance and standardization can have positive secondary impacts on security as well.

3.4.4. LIMITED EXTENSIBILITY

Versions of TMDD prior to version 3.1 provided for an <xs:any> field implementation in specific locations within the standard. This allows system designers and implementers to customize the information sent between owner and external centers. Unfortunately, there was no guidance in how to use these <xs:any> fields and they were located only in specific locations. In addition, given SOAPs defaults for implementation and the method by which they were specified, only one <xs:any> field was allowed at each location where they were specified. Designers of TMDD compliant software might not take into account the potential need for future growth and might not create complex types at each usage of an <xs:any> field. This would lock future growth as it would potentially impact any past implementations of their <xs:any> field definition.

TMDD provides some detailed guidance with regards to implementation of any fields, specifically about the limitations for extending the specification while maintaining compliance with the specification. This guidance, in summary, includes the following requirements for extensions:

- Must remain valid in accordance with the WSDL 1.1 XML Schema and NTCIP 2306 v01.
- May add optional elements in a separated XML schema(s) or WSDL, but may not change the TMDD xsd’s or wsdl’s directly.
- Dialogs may not be modified. Project dialogs may be specified consistent with the above requirements, compliant with NTCIP 2306, support request-response, optionally support subscription-publication, documented within XML, and defined by a WSDL.
- TMDD message content should not be modified, and if required, defined within a separate XML schema.
- Data frames shall use the <xs:any> construct.
- Data frame extensions in TMDD v3.1 have two types, TMDD extensions and project extensions. They are separately identified within the TMDD specification, and guidance is provided for use to allow future growth.
- Data elements in TMDD versions prior to v3.1 have no extension mechanism, but a method was introduced in v3.1. TMDD methods and project specific extension mechanisms are provided.

Version 3.1 of the TMDD specification is a significant improvement for providing guidance and methods for implementation of extensions. There remain some reasonable limitations that in some cases can complicate implementation. In general, all dataframes have this updated implementation, however there remain a few dataframes that have no <xs:any> fields available for use.

3.4.5. TECHNOLOGY LIMITATIONS

SOAP was designed in 1998 as work sponsored by Microsoft and released to the Internet Engineering Task Force in 1999. Version 1.0 was never released. Version 1.1 was released in May 2000, but did not garner a W3C recommendation. Version 1.2 was released in June 2003. The SOAP specification was maintained until July 2009. (https://en.wikipedia.org/wiki/SOAP) The latest W3C recommendation regarding SOAP v 1.2 was released in April 2007.

In terms of the pace of technology in the current world, SOAP is considered quite dated. As a communication mechanism it has been eclipsed by a number of technologies in use today. Some of which are also beginning to see their own use eclipsed by even newer technologies.

The continued use of SOAP as the sole communication mechanism for TMDD creates several issues for future use of the standard:

- Limits the speed at which information is communicated
- Limits the volume of information that can be transferred
- Requires significant complexity and additional infrastructure to provide sufficient performance for large-scale real-time operations with high detail data exchange such as real time on-change intersection signal phase status information
- Limits usage to local or small-scale regional operations
- Limits beneficial growth in information exchange and functional capabilities that would be possible when taking advantage of advances in software and systems technologies
- Increases cost of future implementations
- Limits growth potential of existing TMDD implementations
- Limits growth potential of the nation’s transportation infrastructure

These limitations are primarily a result of the following issues with SOAP as a technology:

- Message size with SOAP is generally larger than the message size other technologies; it’s use of XML is considered a very verbose message format
- Serialization computational costs can be very high, resulting in poorer performance and higher infrastructure requirements
- SOAP has components of its structure to manage security, structure standardization, optional elements, and error handling, resulting in significant overhead
- It adds significant complexity to the exchange of information

There are important benefits from the use of SOAP however, including:

- It is a standard for application programming interfaces and function access, rather than a simple format for encapsulation of data
- It provides standard security elements
- It provides the option of error handling instructions
- It can be transactional across multiple SOAP messages

3.5. PRACTICAL EXPERIENCE IN THE I-210 CONNECTED CORRIDORS PROGRAM

With TMDD as the national ITS standard for Center-to-Center information exchange, the I-210 Connected Corridors program utilized the TMDD standard and implemented its SOAP exchange mechanisms for the vast majority of its interfaces with its sources of data and its use of commands to TMCs for execution of response plans. Many of the issues with TMDD were present during the ICM systems implementation and had impacts on its schedule and cost, both negative and positive. A few examples are provided in the following sections.

3.5.1. MESSAGE SIZE LIMITATIONS

For the city of Arcadia, the size of an intersection signal timing plan inventory message containing the signal plan elements for a single intersection signal in a SOAP XML format was 2,297,870 bytes or 2.3MBs on disk. Given that an inventory message under TMDD must provide inventory records for all assets of interest, given its inability to provide an update, delete, or add action to records, a single message containing all of the intersection signal plan inventories for a large metropolitan area would be very large indeed. According to the LADOT (Fact sheet) in December 2016 there were 4689 signaled intersections connected to their centralized intersection signal control system. If each of LA signal's plan inventory was similar to the sample from Arcadia, the message size for LA's signal plan inventory would be approximately 11 GB in size. Such message sizes are not practical for transmission, and the processing power required for retrieving and serializing such a message are not reasonable in any real time system.

Experience with the I-210 Connected Corridors program indicated that data source systems experienced problems with inventories involving much smaller numbers of intersection signals, and server timeouts waiting for the data upon making a request were not unusual. Testing with some vendors where test inventories were larger failed, requiring the team to limit the number of assets being returned and increase the server timeout periods in order to successfully complete testing.

3.5.2. SCALABILITY LIMITATIONS

Scalability issues often occurred within the Connected Corridors program. Message size issues contribute to the scalability issue, since large message sizes by themselves limit scalability. Clearly a solution of providing a CRUD action, along with a record would limit the size of messages and provide a way to scale these inventory dialogs, allowing individual actions to be split into multiple messages. Methods to ensure a full receipt of a group of messages that represent the full inventory would be required.

In addition, the program also encountered scalability limitations of the source systems when large numbers of smaller messages were transmitted. An example of this was the intersection signal status messages. Some source systems could not retrieve real time intersection signal status from the field controllers due to network capacity limitations in the field. Others could maintain the required field to center bandwidth, but the software and hardware infrastructure available at the control center could not maintain the required stream of intersection signal status messages and as a result, messages were lost. There were potential fixes that could have been pursued for this issue, but due to cost constraints those fixes were not implemented. However, these limitations were for relatively small numbers of intersection signals, and scalability to a full, large scale urban area could be problematic.

3.5.3. COMPLEXITY

Complexity resulting from the use of TMDD and SOAP was experienced within multiple realms, many of them already discussed within this document. The primary complexity related issues were:

- Vendor specific or non-standard implementations of TMDD
- Differing data semantics and temporal behavior of field elements, field capabilities, and vendor software implementations
- Vendor differences within center capabilities and different field equipment standards and their impacts on extensions within TMDD
- Limitations of legacy system architectures or local system hardware or network infrastructure
- Lack of subscription management
- Differences in implementation between Microsoft .NET and Java implementations of SOAP service framework
- Maintaining a multi-vendor solution that allowed variation in vendor implementations, but still maintained a single, common implementation within the data hub

In addition, the SOAP framework reduced complexity in some key ways, including:

- Providing a method to maintain and distribute a web service contract (WSDL) that could be maintained and extended to add additional vendor implementations without impacting previous vendor implementations
- A common method of implementation that vendors understood and had the skills and staff to both understand and implement the specifications provided.

In general, the complexities within the I-210 program were a function of the complexity of the problem being solved rather than any result of the use of the SOAP framework. In many ways, SOAP's technical complexity served to reduce the complexity of the final implementation.

3.5.4. FUTURE EXPANSION

By design, TMDD specifies a center-to-center, point-to-point communication. It's use of SOAP can be performance limiting for high-volume, real-time data exchange. This can limit its usage to local or small regional implementations. An implementation of TMDD at the scale of a Connected Corridors installation is likely the limit of a reasonable TMDD implementation. The Connected Corridors program covers multiple jurisdictions including four cities, the county, state, and several regional agencies with a geographic area of approximately 17 by 3 miles. It integrates multiple systems within these boundaries from the various jurisdictions, including:

- Four intersection signal control systems (state, county, and two local) with the number of intersection controls accessed for data or control limited to about 460 intersections

- A state ATMS system managing ramps, freeway signs, arterial signs, freeway detection, and incidents for the district with approximately 17 miles of freeway along I-210 and 2 miles along I-605 of interest
- Two local arterial sign systems (Pasadena and LA County)
- Two lane closure systems (State and Local)

Figure 3-9Figure 3-9 provides a sense of the scope of the geographic area covered by the I-210 Connected Corridors program in relation to the boundaries of Caltrans District 7. The figure shows the boundaries for Los Angeles County and Ventura County, the two California counties managed by District 7. While the corridor is sizeable and complex, its data footprint in relation to the rest of the district is quite small. There are plans for additional corridors within the District and California, creating complex challenges to scale the Connected Corridors solution.

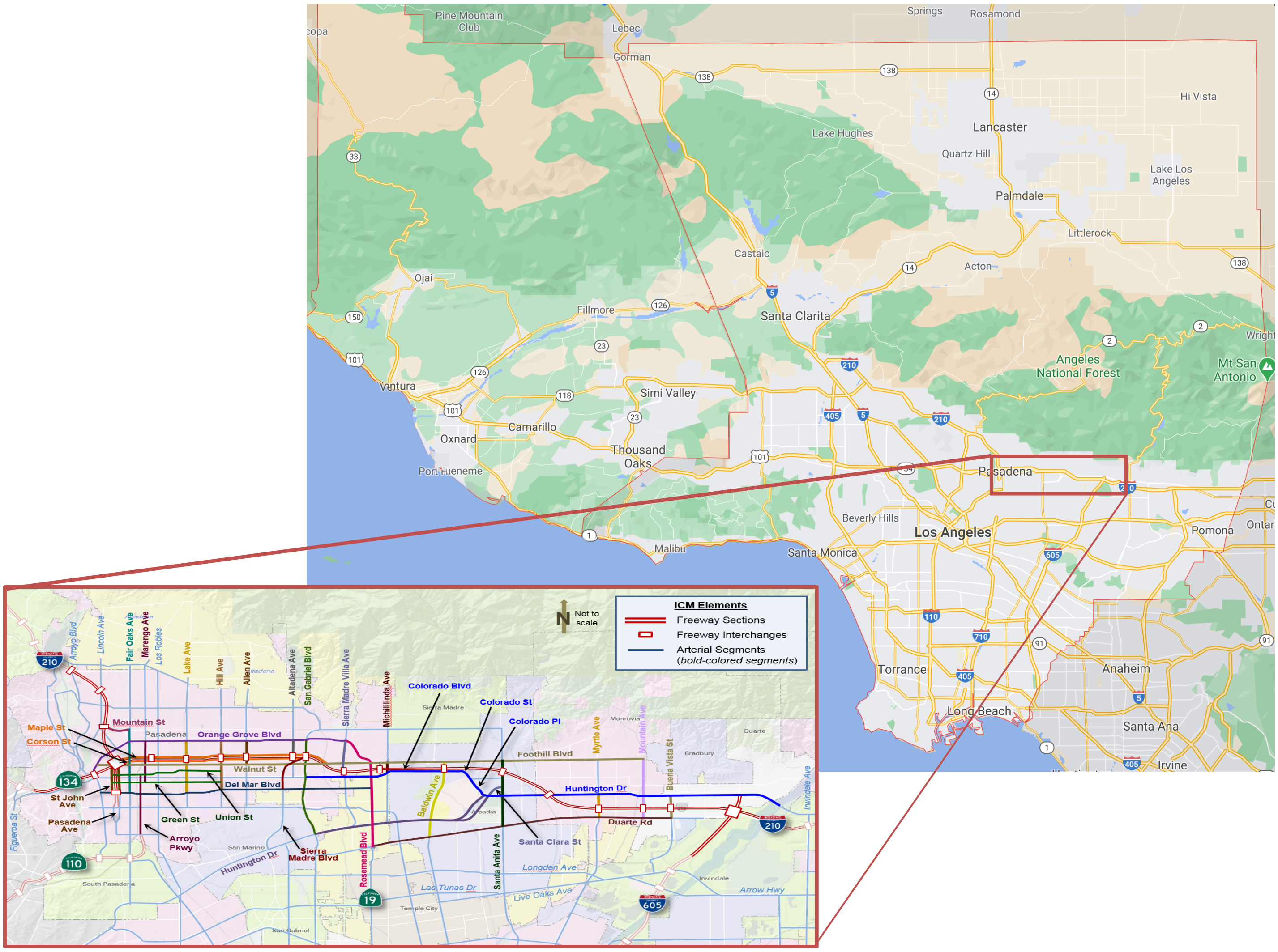

Figure 3-9 I-210 Corridor Geographic Area Within District 7

Another way to view this challenge is provided in Figure 3-10. The diagram illustrates the primary data flows of the ICM system described in Figure 3-9, specifically it's primary feeds. This includes the various ATMS systems that manage local, regional, and state jurisdictional systems and are responsible for providing data and receiving commands to execute approved response plans. Each external system is represented in green. The green command targets on the right represent the same ATMS systems on the left, just simplified for clarity within the diagram. The three primary subsystems of the ICM system: the data hub, corridor management system, and decision support system are represented in red. Each of the external (green) systems use SOAP TMDD protocols to provide data to the ICM system's data hub, and receive commands from the corridor management system via SOAP TMDD as well. Communications between the three ICM subsystems is accomplished using Apache Kafka or ActiveMQ messaging, preserving the TMDD data structures within JSON messages.

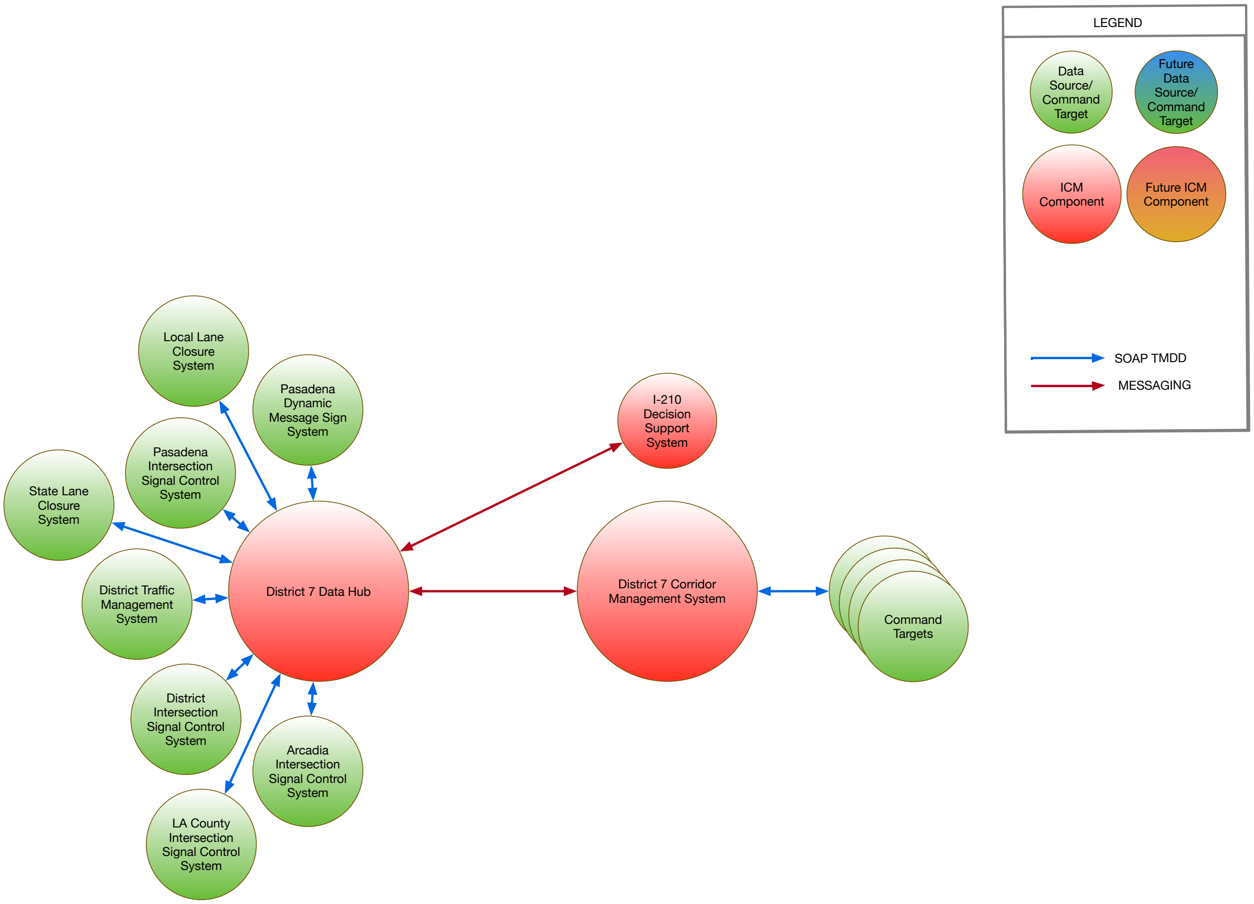

Figure 3-10 Primary I-210 Corridor Data Flow

This represents the connections required for a single, relatively small, urban transportation corridor. However, the design of the Connected Corridors ICM system is expected to support multiple corridors within a Caltrans district. To do this, a single data hub and a single corridor management system will be used to support all of the corridors within a district; and a single decision support system will be used for each corridor within the district.

A larger urban corridor is likely to include the use of additional data sources and individual data sources with larger field element inventories, further increasing the size of the data flows. For instance, Los Angeles itself has over 4000 intersection signals, compared to just over 450 within the I-210 corridor. Figure 3-11 shows the increase in data flows when additional corridors within a district are added, adding both data sources, and potentially larger data sources.

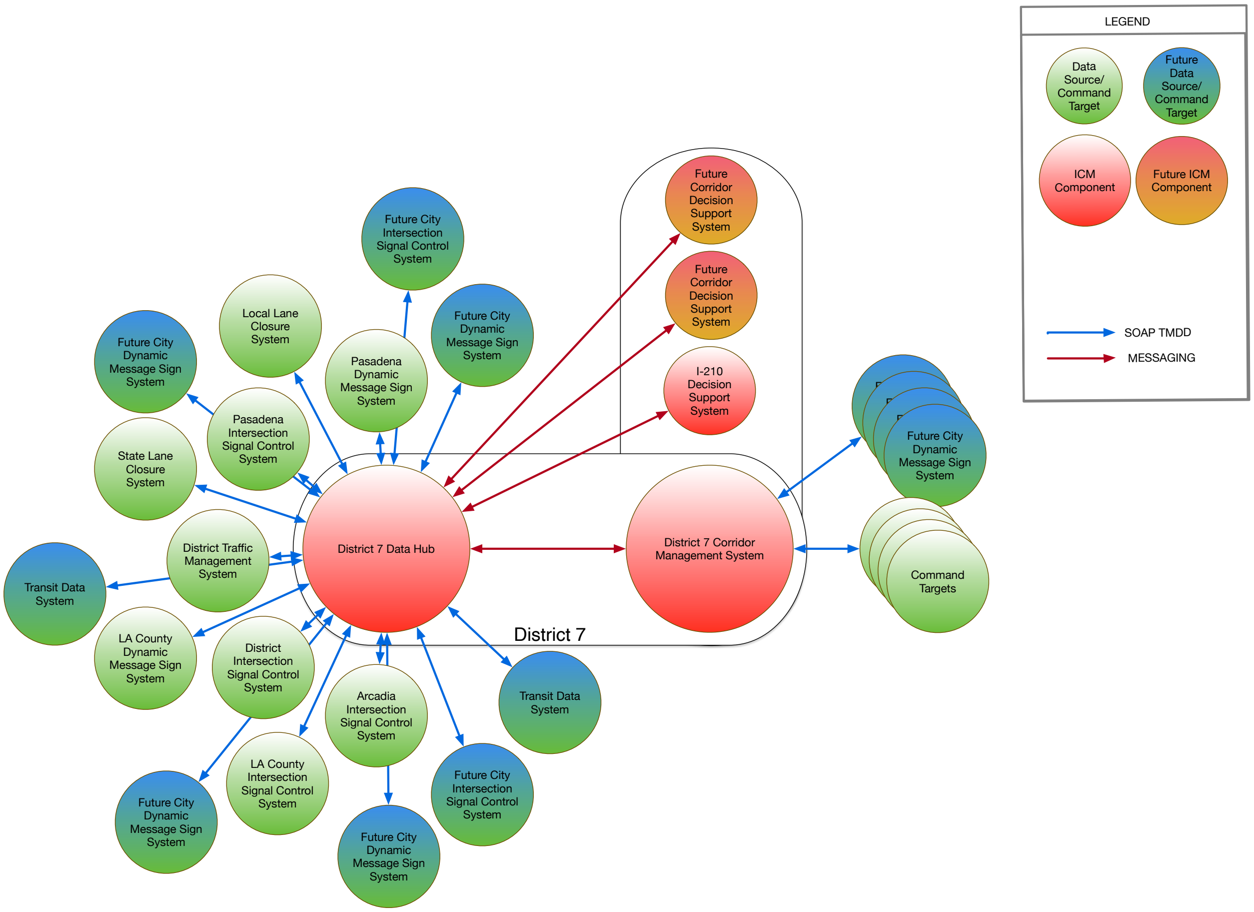

Figure 3-11 Future Expanded Corridor Data Flow

The next logical step in expansion is to deploy ICM or TSMO in other districts, along with a centralized state repository for data collection and analysis at the multi-district level. Such a step provides the benefits of a data centric approach to transportation management throughout the state, as well as an extensive set of data for analysis at the state level to support a wide variety of transportation programs. Figure 3-12 illustrates what the data flow might look like for such a distributed system approach.

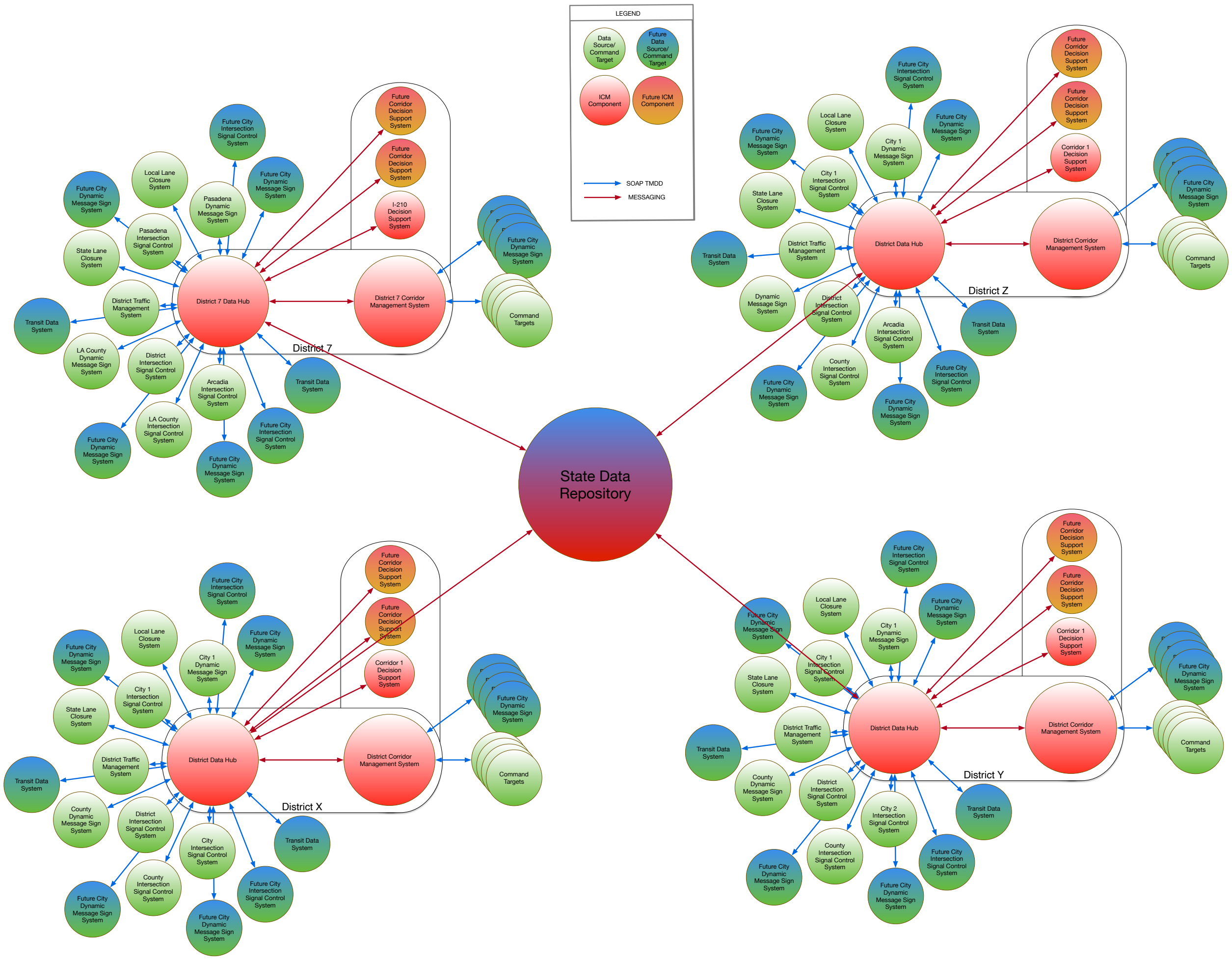

Figure 3-12 Multi-District, Multi-Corridor Data Flow

It's clear that as the need increases for traffic management from a small individual corridor to a statewide information distribution system across multiple corridors and districts, the increases in data scale linearly, assuming the type of data being exchanged is unchanged. Such expansion creates additional challenges in data management. Even basic concepts of unique identifiers for field assets across multiple jurisdictions and management of source systems to ensure consistency of such basic data management and data quality processes become complex. These issues were challenges for the I-210 system implementation, which made no assumptions on the consistency of data from source systems and no coordination of implementation of source systems data management or standards beyond the use of TMDD for transport. Issues not addressed by TMDD as a transport mechanism could be better addressed with additional methods of standardization of transportation related data either at the state level, or likely, the federal level.

As more data types from connected and autonomous vehicles, transit, and other data sources are incorporated into such a distributed transportation data system the data being transported can be expected to grow further.

In each of the diagrams, SOAP TMDD is limited to the individual source system data collection and command execution. SOAP TMDD is ill suited to the larger data volumes and real-time needs once multiple sources of information are aggregated at the district level and beyond. Different, more advanced, faster methods that operate at higher data volumes will be required as the data systems of the future advance beyond single point-to-point, center-to-center communication and the data volumes exceed the capabilities of a SOAP implementation of TMDD.

3.5.5. SECURITY

Security is one area where SOAP provides clear advantages over other protocols and technology selections. It's extension for WS-Security provides enterprise level security features not present within REST services. WS-Security provides for authentication, certificates, digital signature, and encryption. Security related data is added to the SOAP header for messages. In the I-210 program, this included username and password authentication for the external data sources SOAP services.

Security practice, however, in the I-210 project was limited given the capabilities of the source data service organizations. In general, security of the communications was highly reliant on network security. The system runs virtually within a private network in D7. The cloud deployment has no public network endpoints which limits access to the system in the D7 network. D7 and RIITS secured the networking between D7 and each source system. District source systems were routed via the D7 network, and sources external to District 7 are routed via RIITS, generally via a VPN between RIITS and the local source system. Network complexity to restrict access between the ICM and source systems was quite high.

No standard password or username policies exists between the various systems, and in general, the password and username complexity for credentials provided by source systems was non-existent.

Data in transit was generally not encrypted, given the limited capabilities of local organizations to install and maintain the certificates. Some stakeholders expressed that given the VPN encryption and private networking, HTTPS encryption was unnecessary.

SOAP security is only as useful and effective as its implementation. Additionally, within a region with many partners, the ability to implement effective security is often limited. It is an area ripe for significant improvement.

3.5.6. WS-I COMPLIANCE

As mentioned in Section 2.2.1, TMDD's SOAP implementation is not WS-I compliant. As a result, for the I-210 program, PATH modified its implementation of TMDD, violating TMDD's extension limitations, in order to be WS-I compliant. This modification simplified development of the interfaces and has worked well to maintain a common implementation across vendors.

However, there still remained a difference between vendors that used a .NET SOAP implementation and the Java-based implementation of the Connected Corridors program. The difference was in how namespaces were defined and used within the two different implementations. The most recent .NET implementation was more stringent in the use of namespaces within the SOAP implementation. However, this was overcome without changes in the Java implementation within the Connected Corridors interfaces, maintaining a single implementation across the vendor implementations.

3.5.7. MULTI-JURISDICTIONAL, MULTI-VENDOR IMPLEMENTATIONS

The most important issue that the I-210 program surfaced in implementing TMDD is that the usage of data that goes beyond sharing of the information into the realm of algorithmic use of that data, requires much more standardization not only in the format, but in the semantics of the data. Implementation of the TMDD interfaces themselves are complex undertakings between two entities, the owner and the external center, but usable implementations between multiple entities are extensive undertakings. Great care must be taken not only to verify that the communication occurs, but that the data across all vendors has a common meaning, expression, and temporal characteristics. This took significant interaction between each vendor and PATH, as well as great care at PATH to verify each successive implementation either maintains the same format, meaning, and expression, or can be processed in a way to achieve such common format, meaning, and expression. In practice, no two vendors, even with the same specification, achieved a common implementation. The other issues that impact differences between implementations, such as local field communication limitations and differences in field controllers, amplified these differences in implementation.

One result of these multiple implementations was to impose limitations on the systems capabilities. Lack of intersection signal status within one implementation meant that the arterial traffic estimation could not use that information for improvements in estimation accuracy and that loss of intersection signal capabilities and confirmation of control changes could not be verified during system operation.

The other result is the additional effort and cost that is required to achieve even a minimal commonality between multiple vendor implementations. Significant time to evaluate each implementation is required, with multiple rounds of review, testing, and verification of proper operations.

4. SUMMARY OF FINDINGS AND RECOMMENDATIONS

4.1. SUMMARY OF FINDINGS

In general, TMDD, and by extension NTCIP 2306, have provided a basic first step in interoperability between traffic management centers, focused on center-to-center communication. Without such a step, projects such as the I-210 Connected Corridors program would not be possible, or at least would likely be significantly more complex and expensive. The standard provides a framework upon which to build center-to-center communications.

However, it has several shortcomings. These include:

- Lack of coverage of modern traffic management requirements and missing elements
- Strict ties of the standard to the SOAP protocol and its limitations, including limited performance, high verbosity, and high complexity
- Poor implementation guidance
- Limited communication management and communication status capabilities
- Limited security implementation guidance and requirements
- Limitations of point-to-point communications in a multi-center, multi-jurisdiction environment
- Inability to take advantage of and keep pace with the state of systems technology advances

These shortcomings are not just theoretical, but rather based on real experience implementing TMDD in a complex urban environment with multiple jurisdictions and multiple vendors with complex analysis and computation required of the information provided via these TMDD interfaces.

The standard needs significant modernization to address these shortcomings in order to better address both current and future traffic and transportation management requirements, as well as to stay relevant in today’s rapidly advancing technology environment.

4.2. RECOMMENDATIONS

There are a number of improvements that could be made to the TMDD standard. However, most, if not all of these changes would require that NTCIP 2306 be updated, or that the standard abandon its NTCIP 2306 compliance. Based on experience, it is recommended that both standards be updated, with a focus on the following:

- Separation of the standard into two volumes, one for data structure and one for the communication protocol, allowing them to advance independently

- Addition of new dialogs, messages, and dataframes to address new requirements
- Ability for solutions designers to select the technology solution appropriate for the data being communicated, abandoning the one-size-fits-all approach of a SOAP-based standard
- Provide significantly more implementation guidance and standardization of the data semantics and temporal requirements for data exchange
- Updates of the standard to incorporate capabilities of advances in traffic management devices
- Improve the process to update the standard, making it more flexible and responsive to a changing transportation landscape
- Guidance for implementation of security elements, setting a minimum requirement specific to the different technology implementations allowed by the specification

5. REFERENCES

---

[1] AASHTO and ITE, *Traffic Management Data Dictionary (TMDD) Standard for the Center to Center Communications*, Version 3. American Association of State Highway and Transportation Officials (AASHTO), Institute of Transportation Engineers (ITE), 2020.

[2] AASHTO and ITE, *Guide to V3.1 Traffic Management Data Dictionary (TMDD) Standard for Center-to-Center Communications*. American Association of State Highway and Transportation Officials (AASHTO), Institute of Transportation Engineers (ITE), February 2020.

[3] Web Services Interoperability Organization, “Basic Profile Version 1.0.” [Online]. Available: http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html.

[4] Web Services Interoperability Organization, “Basic Profile Version 1.1.” [Online]. Available: http://www.ws-i.org/Profiles/BasicProfile-1.1.html

[5] Web Services Interoperability Organization, “Basic Profile Version 1.2.” [Online]. Available: http://ws-i.org/profiles/BasicProfile-1.2-2010-11-09.html

[6] Web Services Interoperability Organization, “Basic Profile Version 2.0.” [Online]. Available: http://ws-i.org/profiles/BasicProfile-2.0-2010-11-09.html

[7] Partners for Advanced Transportation Technology, Institute of Transportation Studies, University of California, Berkeley, “Connected Corridors: I-210 Pilot Integrated Corridor Management System, Systems Interface Design Specification.” [Online]. Available: https://connected-corridors.berkeley.edu/resources/document-library

This page left blank intentionally