Integrated Modeling for Road Conditions Prediction, Phase 5: System Design Description

April 4, 2025

Final

U.S. Department of Transportation
Federal Highway Administration

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1,000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2,000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERATURE (exact degrees)		
°F	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	°C
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2,000 lb)	T
		TEMPERATURE (exact degrees)		
°C	Celsius	1.8C+32	Fahrenheit	°F
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	2.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380. (Revised March 2003)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADCIRC	Advanced Circulation model for oceanic, coastal, and estuarine waters
AHPS	Advanced Hydrologic Prediction Service
ATMS	Advanced Transportation Management System
CAP	Common Alerting Protocol
CMS	Changeable Message Signs
ConOps	Concept of Operations
CSV	Comma Separated Value
CV	Connected Vehicle
DEM	Digital Elevation Map
ESS	Environmental Sensor Station
FHWA	Federal Highway Administration
GFS	Global Forecast System
GRIB2	NCEP and WMO's Gridded Binary File Format Version 2
IMRCP	Integrated Modeling for Road Condition Prediction
LaDOTD	Louisiana Department of Transportation and Development
MDSS	Maintenance Decision Support System
METRo	Model of the Environment and Temperature of Roads
MLP	Machine Learning-based Prediction
MRMS	Multi Radar Multi Sensor
NCEP	National Centers for Environmental Predications
NDFD	National Digital Forecast Database
NHC	National Hurricane Center
NOAA	National Oceanic and Atmospheric Administration
NWM	National Water Model
NWPS	National Water Prediction Service
NWS	National Weather Service
RAP	Rapid Refresh weather model
RTMA	Real-Time Mesoscale Analysis
SAD	System Architecture Description
SRS	System Requirements Specifications
TMC	Transportation Management Center
UML	Unified Modeling Language
URL	Uniform Resource Locator
WMO	World Meteorological Organization
WxDE	Weather Data Environment

EXECUTIVE SUMMARY

Transportation system management and operations (TSMO) is on the cusp of dramatic changes due to the increased availability of data and the increasing sophistication of models and systems supporting those operations. Intelligent transportation systems (ITS) are widely deployed and gather data about weather and traffic conditions across road networks. The imminent deployment of connected vehicles (CV) will bring an orders-of-magnitude increase in data availability. This array of data powers traffic and road condition predictions, and as data availability increases, the accuracy and reliability of the models improve. This convergence of opportunities presents enough potential for operational improvements in safety and mobility that the Federal Highway Administration's (FHWA) Road Weather Management Program (RWMP) is initiating research into the Integrated Modeling for Road Condition Prediction (IMRCP) system to investigate and capture that potential. This system design description (SDD) documents a common understanding among stakeholder groups of system features and components and serves as a basis for system development activities.

The SDD consists of an introduction describing the objectives of both the IMRCP system and the SDD itself, a general description of the IMRCP, and a design description of the system components.

CHAPTER 1. INTRODUCTION

BACKGROUND

FHWA has embarked on efforts to describe and create a tool that results from an ensemble of forecast and probabilistic models and incorporates real-time and/or archived data, fusing them to predict current and future overall road/travel conditions for travelers, transportation operators, and maintenance providers.

IMRCP Phase 1 created the foundational Concept of Operations and Requirements for the system. The RWMP envisioned the model would be a practical tool for State DOTs to use to support traveler advisories as well as maintenance and operational decisions at strategic and tactical levels. Phases 2 and 3 specified, implemented, tested, and evaluated the IMRCP concept in a demonstration deployment with the Missouri DOT and Kansas DOT in the Kansas City metropolitan area. Phase 4 worked with the Ohio DOT and the Louisiana Department of Transportation and Development (LaDOTD) in two parallel statewide deployments to demonstrate IMRCP applications and decision support in winter and tropical storm events.

PURPOSE

The objective of Phase 5 of IMRCP is to improve and demonstrate IMRCP capabilities as an agency-deployable system with multiple contiguous States managing evacuations and responses to adverse weather conditions. IMRCP improvements are based on the platform and gaps identified in Phase 4. Additional improvement priorities have been developed from stakeholder engagement and requirements to support an agency-deployable system. IMRCP capabilities are demonstrated in a regional deployment over multiple contiguous States—specifically Louisiana, Mississippi, and Alabama—to support analysis and management of evacuation, response, and recovery from adverse weather conditions as identified in the Congressionally-authorized Emergency Planning Transportation Data Initiative. The improved IMRCP capabilities, updated administrative guidance technical documentation, and training materials are to be packaged for future agency deployments and placed in an open-source repository.

This System Design Description updates the Phase 4 design to reflect system changes needed to support the Phase 5 scope and objectives.

SCOPE

IMRCP provides a framework for the integration of road condition monitoring and forecast data to support tactical and strategic decisions by travelers, transportation operators, and maintenance providers. The system will collect and integrate environmental observations and transportation operations data; collect forecast environmental data and operations data when available; initiate road weather and traffic forecasts based on the collected data; generate travel and operational advisories and warnings from the collected real-time and forecast data; and provide the road condition data, forecasts, advisories and warnings to other applications and systems. Road condition and operations data and forecasts to be integrated into the predictions may include atmospheric weather; road (surface) weather; small stream, river, and coastal water levels; road network capacity; road network demand; traffic conditions and forecasts; traffic control states;

work zones; maintenance activities and plans; and data related to emergency preparedness and emergency operations.

DOCUMENT OVERVIEW

The structure of this document is generally consistent with the outline of a System or Software Design Description defined in ISO/IEC/IEEE 15288:2023. Some chapters herein have been enhanced to accommodate more detailed content than described in the standard. Titles of some chapters have been edited to specifically capture that enhancement.

Chapter 2 provides a general description of the system perspective and stakeholder concerns. It is largely a summary of material described in more detail in the Concept of Operations (ConOps).

Chapter 3 documents the system design. The relevant architectural viewpoints are identified, and views and models are described for each viewpoint. Rationales for and correspondence among elements of the views are included in the view and model descriptions. Four viewpoints are described: composition, process, deployment, and related designs.

CHAPTER 2. GENERAL DESCRIPTION

SYSTEM PERSPECTIVE

Describing and predicting roadway conditions and events that may impact travel across road networks requires an understanding of and tools for interacting with the system and its operations across all of the road network's stakeholder groups. For example, travelers have an immediate need for information about conditions along their planned route, and they contribute to the aggregate travel conditions along their route by their choices and behaviors. Winter maintenance crews plan ahead for reducing the impact of storms on roadway conditions based on weather forecasts and perhaps on a sophisticated maintenance decision support system (MDSS), but also adapt to conditions on the roadway as they execute those plans. Operators in a transportation management center (TMC) monitor roadway conditions across a network with cameras and sensors accessed through an Advanced Transportation Management System (ATMS), and respond to conditions and events by generating alerts to be published on changeable message signs (CMS) on the roadside and pushed out to web pages and mobile apps through traveler information systems. In all these examples, stakeholders are making and executing plans, monitoring and adjusting to current conditions, and potentially changing their plans based on their analyses of potential future conditions.

A complete context for prediction of road conditions would have to consider a broad range of stakeholders, their activities, and interactions with the roadway, their decision processes, and the underlying models of the roadway and environmental conditions. Descriptions of the current state of stakeholders and their activities in the IMRCP ConOps therefore focused on identifying the processes and decisions that are affected by currently available roadway condition information and predictions. An analysis of current and imminent road and weather condition models was performed in a previous task and documented in the Integrated Modeling for Road Condition Prediction Model Analysis Upon delivery of IMRCP-5 to GitHub, the Model Analysis will also be available on GitHub. The aggregate of these analyses of modeling capabilities and stakeholder interests formed the basis for the architectural views of the functional and system packages for a potential IMRCP system in the ConOps.

Architectural views of a system provide sketches of what an implemented system might look like from various conceptual frameworks. For the IMRCP, the functional view of the system describes the system's purpose: to provide integrated predictions of road weather and traffic conditions. To fulfill this purpose, the system will have to have models for the roadways, weather, traffic, and hydrology; prediction capabilities and forecasts from other models; and current observations to set initial conditions for the predictions.

The system package view describes the system as a set of software packages to be implemented and deployed for operations. The IMRCP system package view divides the system packages into package types, which include data collection, models, forecasting, forecasts, decision support, and interface. The view also illustrates the relationships between these packages and external sources. The functional view and the system package view provide a context for the development and structuring of the system requirements. Specific requirements for the IMRCP can be found in the *Integrated Modeling for Road Condition Prediction System Requirements Specifications* (SRS).

Based on the requirements determined in the SRS, the system architecture is created using a set of architectural views in the *Integrated Modeling for Road Condition Prediction System Architecture Description* (SAD). The composition view describes the system in terms of sets of software components and their relationships. The process view describes the system in terms of data processing functions and flows. The deployment view describes the system in terms of the deployment of components to computing devices or nodes.

USER ROLES

The IMRCP SAD carried over a description of IMRCP end users from the SRS that was used to develop the overall system architecture. IMRCP design and implementation adds additional user roles for administrating the system and for administrating IMRCP road network models.

A system administrator is an operating system user that can set and elevate system privileges, modify IMRCP configuration settings, and manage IMRCP user privileges.

An IMRCP administrator is a user of the IMRCP system that can create and edit road networks, add and disable users, and access all of the interfaces available to an IMRCP user. IMRCP administrator privileges are assigned by system administrators.

An IMRCP user has access to the Map, Dashboard, Scenarios, Reports, and User Settings web interfaces. IMRCP user privileges are assigned by system administrators.

CHAPTER 3. DESIGN DESCRIPTION

The IMRCP system design is described here as a set of subsystem packages. These subsystem packages were previously identified and described in the System Architecture Description (SAD) composition view and are used in this design description as organizing entities for the details of the system design. The Unified Modeling Language (UML)[1] package and activity diagrams are used to illustrate the technical concepts and maintain continuity with the SAD. The integrated high-level composition view of the system is shown as a package diagram in Figure 1~~Figure 1~~, and the overall processing is illustrated with the activity diagram in Figure 2~~Figure 2~~. The deployment view from the SAD is filled out with more detail in allocating subsystem packages to particular computing devices on UML deployment diagrams. The requirements from which the design is derived are documented in the *Integrated Modeling for Road Condition Prediction System Requirements Specification*.[2]

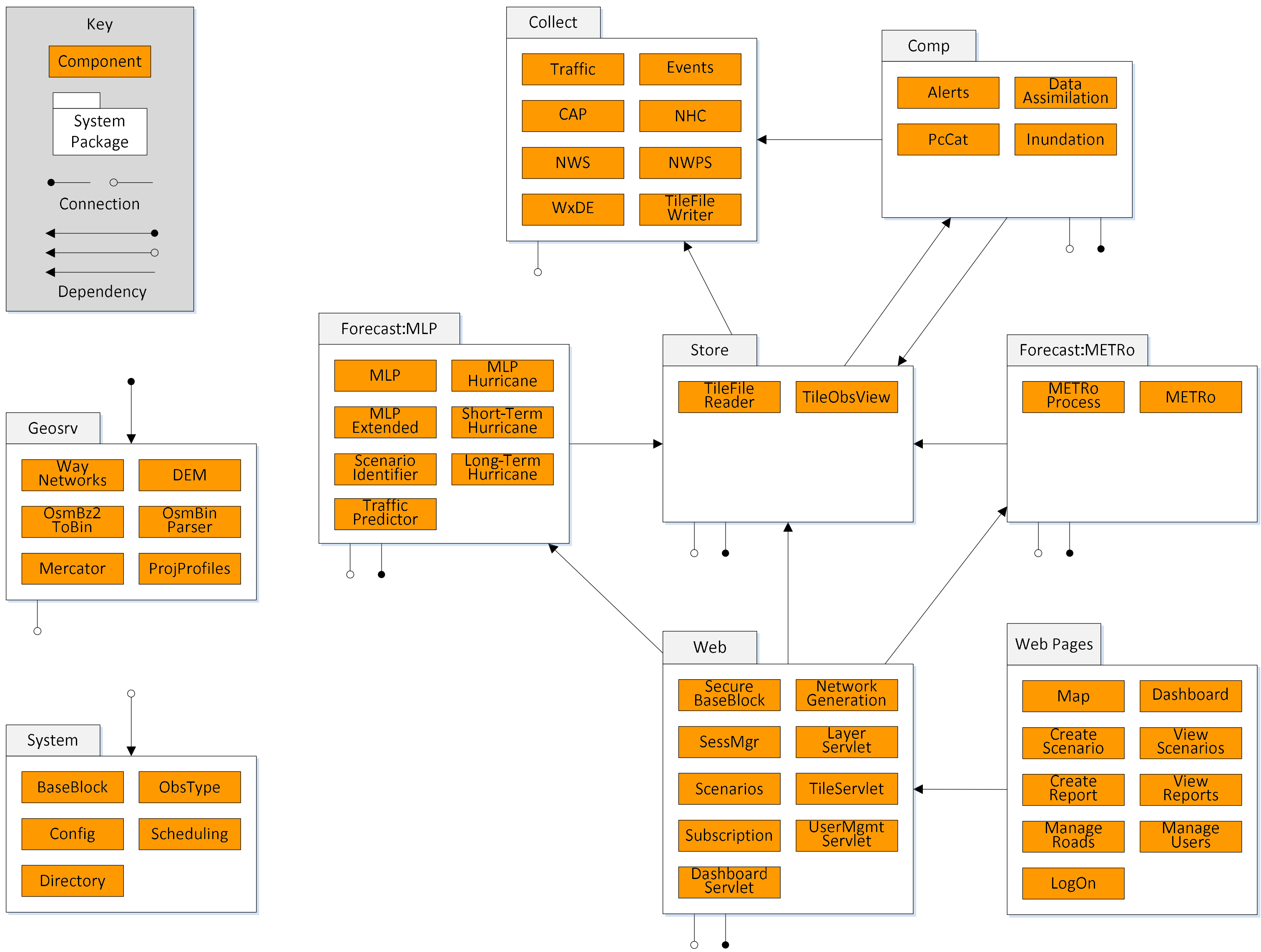
The description of each component will include its inputs, outputs, process, and dependencies.

- Inputs describe the things that the component needs to execute its process.
- Outputs describe the things that the component creates for other components to use.
- Process describes the procedure the component follows to achieve its designed purpose.
- Dependencies describe the components that are invoked by the component being described. Dependencies do not include configuration options or programming language features. Dependencies on an external interface are noted when necessary.

Further supporting design details can be found in the appendices and the IMRCP GitHub repository at https://github.com/OSADP/IMRCP.

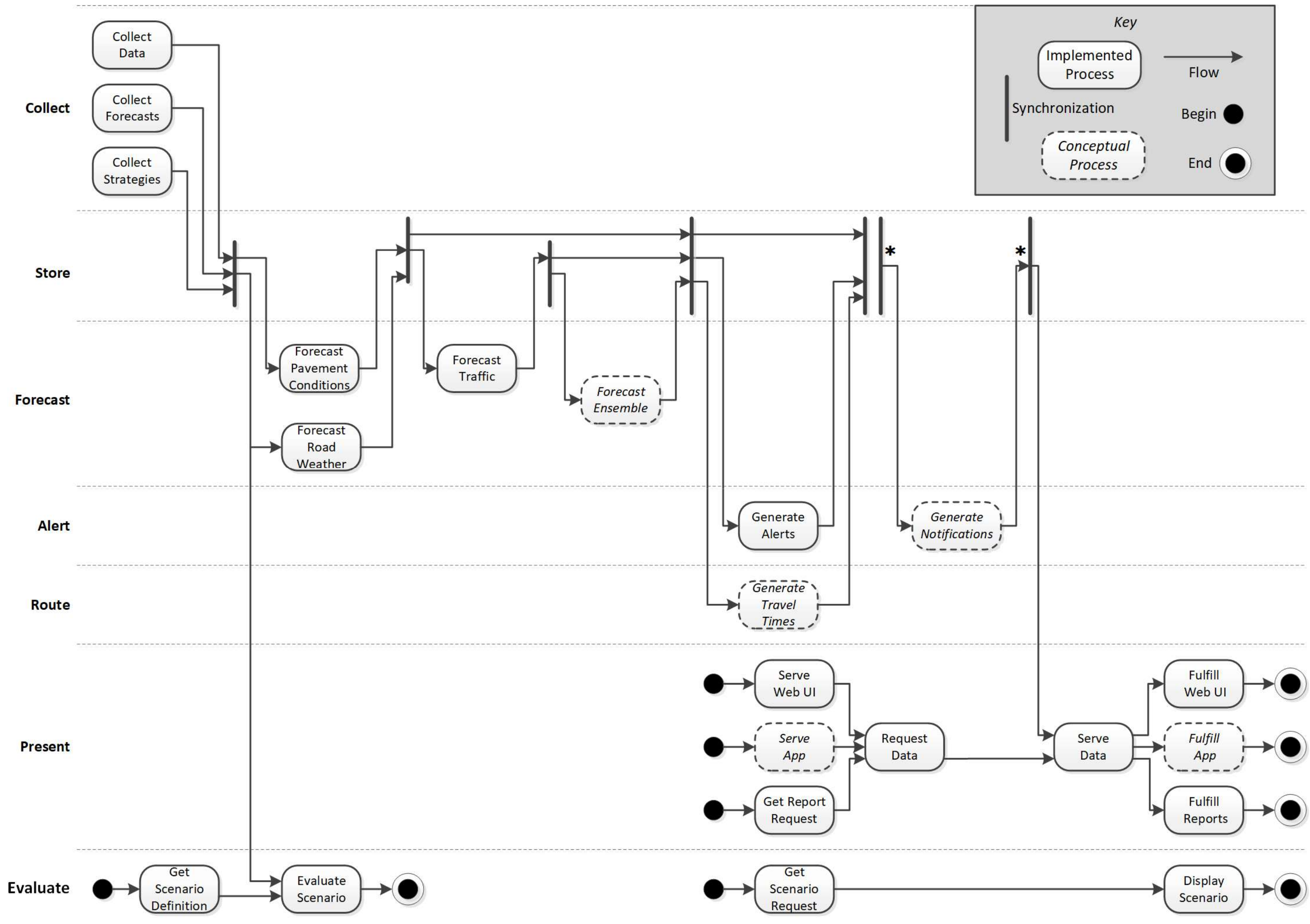
[1] Object Management Group. *OMG Unified Modeling Language*™, Version 2.5. OMG Document Number: formal/2015-03-01. Normative Reference: http://www.omg.org/spec/UML/2.5. Available at: https://www.omg.org/spec/UML/2.5/PDF, last referenced March 27, 2019.

[2] Leidos, Integrated Modeling for Road Condition Prediction System Requirements. 2016. Unpublished working paper developed under FHWA Contract DTFH61-12-D-00050, Task Order 5022, Integrated Modeling for Road Condition Prediction.



Source: FHWA.

Figure 1. Diagram. Integrated Modeling for Road Condition Prediction Composition.



Source: FHWA.

Figure 2. Diagram. Integrated Modeling for Road Condition Prediction Process.

COLLECT PACKAGE

The data collection (Collect) package collects data needed for IMRCP computations and forecast models. Collect components depend on external systems for making the data available for collection. Collection interfaces, formats, and intervals are determined by the external data sources. Appendix A~~Appendix A~~ lists the data availability for the systems accessed by the Collect components. The package is designed to be extensible and will accommodate new collection components as new sources are identified and made accessible to the IMRCP. All Collector components must have a corresponding Resource Record (described in detail in the System package) which encapsulates all of the parameters needed to create and store data files for system use. Any component used to collect remote data should inherit from the Collector class to make use of common parameters and functions. Components initially included within the Collect package include:

Collector Component

Background: IMRCP relies on many different types of data that are collected from various sources. The Collector class is the base class used for any remote data collection. Collectors are responsible for gathering data and processing that data into the IMRCP Tiled Data format to be used by the Store to serve data to the rest of the system.

Inputs: Inputs for Collectors reside in configuration files and include:

- base URL: The base Uniform Resource Locator (URL) of the source of the collected files
- source file: A formatted string to determine time dependent file names
- number of strings: The number of character strings per observation that need to be stored, examples of these include external system IDs, roadways affected, event description, etc.
- resource record: encapsulates the information needed to create and find archive and tiled data files

Outputs: Each Collector saves the collected files to the configured storage location and processes those files into IMRCP Tiled Data files.

Process: Collectors establish a connection to external sources and download available files for the system to use. Depending on the source, the method to obtain and process the data varies, but the outcome is the same.

Dependencies: Collectors are dependent on their external source and an Internet connection.

TileFileWriter Component

Background: The data Store expects data to be formatted into IMRCP Tiled Data files. TileFileWriter components are responsible for processing data from external sources or other components into IMRCP Tiled Data files. Therefore, all Collectors are TileFileWriters as well.

Inputs: Observed or predicted data from external sources or other components to process into IMRCP Tiled Data files along with the corresponding ResourceRecord used to determine where to store the files.

Outputs: IMRCP Tiled Data saved to the configured storage location.

Process: Once data is collected from an external source, or a forecast model is used to predict data, that data is processed into IMRCP Tiled Data files and saved to the configured storage location. The data is then available for use by the data Store.

Dependencies: TileFileWriters are dependent on data from external sources or other components and configured ResourceRecords.

Traffic Component

Background: Knowing the current traffic conditions are imperative for predicting future conditions. Transportation agencies provide current speed and sometimes volume and occupancy data feeds that IMRCP can use in forecast traffic models. Interface formats vary among agencies and systems and data feeds must be adapted to conform to the IMRCP Speed Data Adapter file for events. This process should happen outside of IMRCP.

Inputs: IMRCP Speed Data Adapter files containing speed, volume, and occupancy data.

Outputs: IMRCP Tiled Data files containing speed, volume, and occupancy data

Process: An external adapter should convert speed, volume, and occupancy data feeds into IMRCP Speed Data Adapter files. The Traffic collector then polls for new files on a fixed schedule to ensure up-to-date information being available to the system.

Dependencies: External speed, volume, and occupancy data feeds and an Internet connection

Events Component

Background: Events that occur upon roadways impact traffic flow and are necessary to correctly predict traffic speeds. Transportation agencies provide work zone and incident data feeds that IMRCP can use in forecast traffic models. Interface formats vary among agencies and systems and data feeds must be adapted to conform to the IMRCP Event Data Adapter file for events. This process should happen outside of IMRCP.

Inputs: IMRCP Event Data Adapter files containing incident and work zone data

Outputs: IMRCP Tiled Data files containing work zone and incident information

Process: An external adapter should convert work zone and incident data feeds into IMRCP Event Data Adapter files. The Events collector then polls for new files on a fixed schedule to ensure up-to-date information being available to the system.

Dependencies: External work zone and incident data feeds and an Internet connection

NWS Component

Background: The National Weather Service (NWS) collector provides a set of public atmospheric weather forecast products that vary by observation type, temporal extent and spatial extent. These products, for example Real-Time Mesoscale Analysis (RTMA), Rapid Refresh weather model (RAP), National Digital Forecast Database (NDFD), Global Forecast System (GFS), and Multi Radar Multi Sensor (MRMS), make up the bulk of weather data ingested by IMRCP. This generic collector can be configured to collect data from each of these products.

Inputs: NWS files in the National Centers for Environmental Predications (NCEP) and World Meteorological Organization‘s (WMO) Gridded Binary File Format Version 2 (GRIB2) containing observed and predicted weather data

Outputs: IMRCP Tiled Data files containing observed and predicted weather data

Process: A connection with NWS servers is established and a list of possible files to download is determined by the collector’s configuration. This list is compared to the files that have already been collected in the past. Any file that IMRCP does not have is then downloaded and, depending on configuration, is processed immediately or on demand into IMRCP Tiled Data files.

Dependencies: NWS GRIB2 files and an Internet connection

NHC Component

Background: The National Hurricane Center (NHC) provides hurricane probability cones and hurricane category forecasts. These forecasts are used as part of the Machine Learning-based Prediction (MLP) Hurricane Traffic Prediction model.

Inputs: NHC forecast files

Outputs: IMRCP Tiled Data files containing hurricane data

Process: A connection with NHC servers is established and a list of possible files to download is determined by the collector’s configuration. This list is compared to the files that have already been collected in the past. Any file that IMRCP does not have is then downloaded and is processed into IMRCP Tiled Data files.

Dependencies: NHC Hurricane forecast files and an Internet connection

NWPS Component

Background: The National Water Prediction Service (NWPS) (previously Advanced Hydrologic Prediction Service [AHPS]) provides flood stage observations and predictions for river and stream beds. Static inundation maps are also available for some of the stations.

Inputs: NWPS observation, forecast, and inundation files.

Outputs: IMRCP Tiled Data files containing flood stage observations and forecasts and inundation polygons for the current flood stage.

Process: NWPS servers are polled on a fixed schedule to determine if updated observations and forecasts are available. These files are not updated on a precise fixed schedule. If updated files are available, they are downloaded and then processed into IMRCP Tiled Data files.

Dependencies: NWPS observation, forecast, and inundation files and an Internet connection

NWM Component

Background: The National Water Model (NWM) is a hydrologic modeling framework that simulates observed and forecast streamflow, providing hydrologic guidance at a much larger number of locations than previously available using only traditional river forecasts. Multiple products are different temporal extents are available including inundation extent.

Inputs: Records from NWM FeatureServer products.

Outputs: IMRCP Tiled Data files containing polygons describing areas that are inundated.

Process: The most current data from the configured NWM FeatureServer product is downloaded on a fixed schedule and processed into IMRCP Tiled Data files.

Dependencies: NWM FeatureServer products and an Internet connection

WxDE Component

Background: The Weather Data Environment (WxDE) collector gets environmental sensor station (ESS) observation data from WxDE comma separated value (CSV) subscription files. Some of these observations, namely surface and subsurface temperatures, are used to initialize pavement conditions for the Model of the Environment and Temperature of Roads (METRo) road weather forecast model.

Inputs: WxDE CSV subscription files

Outputs: IMRCP Tiled Data files containing ESS observation data

Process: WxDE subscription files are updated on a fixed schedule. The collector downloads the files when they become available and process the observations into IMRCP Tiled Data files.

Dependencies: WxDE CSV subscription files and an Internet connection

STORE PACAKGE

TileFileReader Component

Background: All observed and forecasted data in the system are stored in IMRCP Tiled Data files. The static parseFile function is called to parse these files and create Observation objects that encapsulate observed and forecasted data.

Inputs: The file path, observation type, spatial and temporal extents, and Resource Record used to find the file are inputs for the parseFile function.

Outputs: A list of Observation objects that match the data request.

Process: TileObsView calls the parseFile function to fulfill data requests from other components.

Dependencies: TileFileReader is dependent on TileFileWriter components to create and store IMRCP Tiled Data files.

TileObsView Component

Background: The Tile Observation View component is the main endpoint for other components within the system to request specific observation types from the data Store. All observed and forecasted data in the system are stored in IMRCP Tiled Data files. These files are indexed in multiple ways to ensure data queries can be fulfilled efficiently. The file directory structure indexes data by observation type, source, and time; and the format of the files indexes data spatially. This component uses configured Resource Records to know how to find all of the data the system contains. The Tile Observation View aggregates all the requested data from the individual stores and returns the observation objects to the requesting components.

Inputs: Inputs to the Tile Observation View Store component include requests from other components within the system for specific observation types within specific geographic and temporal domains.

Outputs: The data requested by other components within the system are the outputs of the Tile Observation View component.

Process: Components within the IMRCP system request specific observation type data within a specific geographic and temporal domain from the Tile Observation View component using the getData method. Requests are fulfilled by searching through IMRCP Tiled Data files to find data that match the request. Searches are performed by looking up what ResourceRecords contain the requested observation type and using their information to determine where the Tiled Data files that could fulfill the request are located. The static parseFile method is then called to parse a Tiled Data file that matches the request. If no files are found to fulfil the request, TileObsView can request TileFileWriters to process the time extent of the request to create Tiled Data files if the corresponding archive files exist. Some TileFileWriters are not configured to reprocess time extents since the process to create Tiled Data files consumes too many system resources and time.

Dependencies: The Tile Observation View component is dependent on TileFileWriter components to create and store IMRCP Tiled Data files and ResourceRecords to describe how to find the data stored.

SYSTEM PACKAGE

The System package provides the base system operation components and utilities. These components are the first to be instantiated on system startup and provide services that enable other components to interact within the system.

Directory Component

Background: The Directory is the main system component that initializes the system by identifying and managing BaseBlocks and services available within the system. Use of a Directory component enhances system extensibility and maintainability relative to a closed system. Configuration (config) for other system components is handled by the Directory as well. JSON files contain configurations for all components and are accessed on demand. These config files reside in a single directory and can be for a Java class or a named BaseBlock instance. If the configuration file is for a Java class, the file name is the fully quantified java class name with the '.' character replaced with '_', for example the config file for the Directory class (imrcp.system.Directory) is imrcp_system_Directory.json. For a named BaseBlock, the file name is the name of the BaseBlock, for example, the config file for the RTMA collector is RTMA.json. Configuration items in named BaseBlock files will overwrite any item in a Java class name file. If config items cannot be found in either type of file, the default value will be used.

Inputs: Inputs for the Directory reside in the imrcp_system_Directory.json configuration file and include:

- classes: A string array that contains the BaseBlocks to instantiate and manage. The strings come in pairs, the first being the fully qualified Java name, and the second the name for the instance. An "@" can be placed at the front of the fully qualified Java name to represent a BaseBlock that will be instantiated by the web server's container manager instead of the Directory. An "#" can be placed at the front of the fully qualified Java name to have the Directory ignore that entry.
- threads: The number of threads system ThreadPool will use.
- resources: An object array that describes all of the Resource Records used by the system. A Resource Record is used to encapsulate all of the information needed to determine archive file names, read and write Tiled Data files, and search for data in the system. Resource Record parameters are listed below:
    - writer: Name of the TileFileWriter used to create Tiled Data files for this Resource Record.
    - contrib: Contributor ID, which is an up to 6-character string that gets converted into an integer using base-36.
    - sourceid: Source ID, which is an up to 6-character string that gets converted into an integer using base-36. Source Ids by default are the contributor id. They are used to differentiate data files when a contributor has multiple source files. For example, the NWPS Collector gets a file that contains observations and a different file that contains forecasts. A different Resource Record is used for the two files.

- archiveff: Archive File Format string. This string is appended to the TileFileWriter's Archive Path to create a string used to generate time dependent file names. This can be blank for Resource Record of derived observations like the PcCat.
- tileff: Tiled Data File Format string. This string is appended to the TileFilerWriter's Data Path to create a string used to generate time dependent file names.
- hrz: Name of horizontal axis in GRIB2 files.
- vrt: Name of vertical axis in GRIB2 files.
- time: Name of time axis in GRIB2 files.
- range: The time in milliseconds that files are valid for after the start time of the file.
- freq: The frequency in milliseconds that data files are produced for this Resource Record.
- archivefreq: The frequency in milliseconds that archive files are received for this Resource Record.
- zoom: The zoom level of the tiles used in the Tiled Data files.
- obsid: Array of observation types produced by the Resource Record.
- <obstypeids>: For each value in the obsid array, there is a key of the same value in the JSON config object. The value for this key is an array of strings that are labels for that observation type in GRIB2 files. If source files are not GRIB2 then these keys are not needed.
- srcunits: The values in this array correspond to the observation types in the obsid array. Each value is the units use in the source file for that observation type and is used to convert the observation into the correct system units defined in the ObsType class.
- round: The values in this array correspond to the observation types in the obsid array. Each value is the precision used to round observations to in the system units. For example, a round value of 1 for an air temperature would round temperature to the nearest degrees C.
- classes: The values in this array correspond to the observation types in the obsid array. Each value is the fully quantified java class name of the class to use for merging grids together in a GRIB2 file. This is usually blank meaning the default class for the TileFileWriter will be used.
- preference: The values in this array correspond to the observation types in the obsid array. A preference value is used by the data Store to determine which observation to use, if there are coincident observations from different sources. An observation from a Resource Record with a lower preference value will be used before an observation with a higher preference value.
- tilesize: A value used to determine the pixels per tile in a Tiled Data file. The equation to calculate pixels per tile is 2^tilesize – 1.
- maxfcst: The time in milliseconds that forecasts can be produced by this Resource Record for a single time interval. Defaults to the range if not specified.
- delay: The time in milliseconds added to the received time of a file to get the start time of the file.

- tilesearchoffset: The offset in milliseconds to keep searching for Tiled Data files. This value is divided by the frequency to determine how many Tiled Data files should be used before the store quits searching for more data.
- archivesearchoffset: The offset in milliseconds to keep searching for archive data files. This value is divided by the archive frequency to determine how many archive files should be used before the store quits searching for more data.
- reprocess: A Boolean to tell if Tiled Data files can be reprocessed on demand.
- varies: A Boolean to tell if observations are in file indexed by the VARIES observation type instead of individual observation types.
- boundingbox: An array of integers scaled to 7 decimal places describing the bounding box to process. The order is [min lon, min lat, max lon, max lat].
- valuetype: The values in this array correspond to the observation types in the obsid array and tell what kind of data type is used to write the observation values in Tiled Data files for this Resource Record. The enumeration of values is defined in Appendix F.

Outputs: The Directory provides notifications from one BaseBlock to another and Resource Records and Configuration objects to system components.

Process: Directory is the first servlet instantiated by the web server's container manager and initializes the rest of the system. First the Scheduling component is instantiated. Next the BaseBlock instances in the classes array are instantiated, initialized, registered to the Directory, and queued to start. Since some BaseBlocks depend on data from other BaseBlocks to start, the Directory dynamically determines the instantiation order based off of the BaseBlock's configuration, ensuring that BaseBlocks are only started after their dependencies have finished their start method.

Dependencies: The Directory is dependent on the web server's container manager.

ObsType Component

Background: The different observation types in the system have metadata associated with them including name, integer value, English and metric units, and enumerated values that are managed by the ObsType class. It also contains the RangeRule objects that are used to bucket observation values to aid with presentation.

Inputs: Inputs for ObsType reside in the configuration file and include:

- lrain: light rain threshold in kg/(m^2*s)
- mrain: medium rain threshold in kg/(m^2*s)
- lsnow: light snow threshold in kg/(m^2*s)
- msnow: medium snow threshold in kg/(m^2*s)
- raintemp: temperature for rain threshold in K
- snowtemp: temperature for snow threshold in K
- lightrainmm: light rain threshold in mm/hr
- medrainmm: medium rain threshold in mm/hr
- lightsnowmm: light snow threshold in mm/hr

- medsnowmm: medium snow threshold in mm/hr
- rules: a string array of observation types that have configured RangeRules

Outputs: The ObsType class returns requested metadata associated with the observation types.

Process: Upon receiving the first request from another component, configuration values are retrieved from Config. All of the observation types and enumerated values are defined in the static block of the class. System components can query ObsType for observation type's name, English or metric units, enumerated values, and RangeRules.

Dependencies: ObsType is dependent on Directory.

Scheduling Component

Background: The Scheduling component provides scheduling services to other system components, such as collecting forecast files on a regular fixed interval. The Scheduling and Configuration components work together to orchestrate the process-related operations of the system.

Inputs: The inputs to Scheduling interfaces include the Runnable component (in most cases a BaseBlock) to be scheduled, period of execution and midnight offset, or delay time.

Outputs: Scheduling returns a schedule id to ensure that only the component that requested the scheduled tasks to be created can request them to be canceled.

Process: Scheduling uses a built in Java thread pool to execute tasks. During the start method of a BaseBlock that has a task to be executed on a regular schedule, it requests a schedule to be created and receives a schedule id from Scheduling. The schedule id is used as a parameter to request a BaseBlock's task to be canceled.

Dependencies: The Scheduling component does not have any dependencies.

COMPUTATION [COMP] PACAKGE

The Computation (Comp) package contains components that use algorithms to compute new observations based on other system observations and conditions. For example, alert generation is based on a set of rules in the form of logical statements about transportation system conditions. A "slick pavement" alert could be based on a measurement of ice on a roadway segment, or on an assessment of pavement surface temperature less than a configured threshold temperature with precipitation present along a roadway segment. The level of alert (i.e., advisory, watch, or warning) depends on the confidence and likelihood of the conditions. An observation or measurement of a condition would merit a warning, whereas an assessment based on future regional conditions might only warrant an advisory.

Alerts Component

Background: Alerts components can generate alerts for configured situations related to exceptional weather, hydrological, or traffic observations/predictions.

Inputs: Alerts components use data from the data Store and rule sets found in the configuration file.

Outputs: Alert components save generated alerts to an IMRCP Tiled Data file.

Process: Since alerts are primarily used to identify locations that require a closer look, the Alerts component only creates alerts on demand when a request is made from another system component like the Map or Dashboard webpage. Data is gathered by the data Store that matches the request and compared with a set of rules in the form of logical statements defined in the configuration file about transportation system conditions. If the data meets the requirements of a rule, an alert is generated and saved to an IMRCP Tiled Data file. Rules for Alerts were defined for the sources available at the time of implementation. Different logical statements can be defined to meet the needs of future sources.

Dependencies: Alerts components are dependent on the data Store.

DataAssimilation Component

Background: IMRCP uses algorithms based on observed and forecasted conditions to predict weather and roadway conditions for large areas. Some external data sources and types only provide limited spatial coverage of conditions. To be able to make predictions for the desired spatial extents in IMRCP, Data Assimilation techniques are used to combine a numerical model, namely a class of Kriging algorithms, with observations to have a more complete coverage of conditions. Currently these techniques are used to approximate surface and sub-surface temperatures across road networks using observed values from ESS.

Inputs: DataAssimilation components use data from the data Store.

Outputs: DataAssimilation components save computed approximate values to an IMRCP Tiled Data file.

Process: DataAssimilation components run on a fixed schedule (defined by the configured values for the period and midnight offset). Each execution cycle, the most recent data is gathered from the data Store to process. The data assimilation algorithm (inverse weighted distance) is run on the gathered data and produces a grid of approximate values of the desired observation type.

Dependencies: Data Assimilation components are dependent on the data Store.

Precipitation Category [PcCat] Component

Background: Many weather forecast products provide precipitation forecasts as a category, instead of a numeric rate. IMRCP collects raw precipitation rate and precipitation type data from different sources and converts those rates and types into categories to have easily understandable precipitation forecasts.

Inputs: PcCat uses precipitation rate, precipitation type, and air temperature data as input to produce precipitation category observations.

Outputs: IMRCP Tiled Data files with precipitation category data.

Process: PcCat components convert precipitation rate and type observations and forecasts into easily understandable precipitation categories. If the precipitation type is not provided by the source of the precipitation rate, air temperature is used to infer the type. Since precipitation categories are not used for any forecast components, Tiled Data files are only made on demand when another system component, like the Map or Dashboard webpage, requests data.

Dependencies: PcCat components are dependent on Collector blocks that collect precipitation rate, precipitation type, and air temperature data.

Inundation Component

Background: Floods are dangerous and can damage transportation agency assets. The Inundation component uses inundation maps and flood stage data to determine if any roadway segments modeled by the system are or will be flooded.

Inputs: Inundation maps and flood stage data.

Outputs: IMRCP Tiled Data files with flooded roadway data.

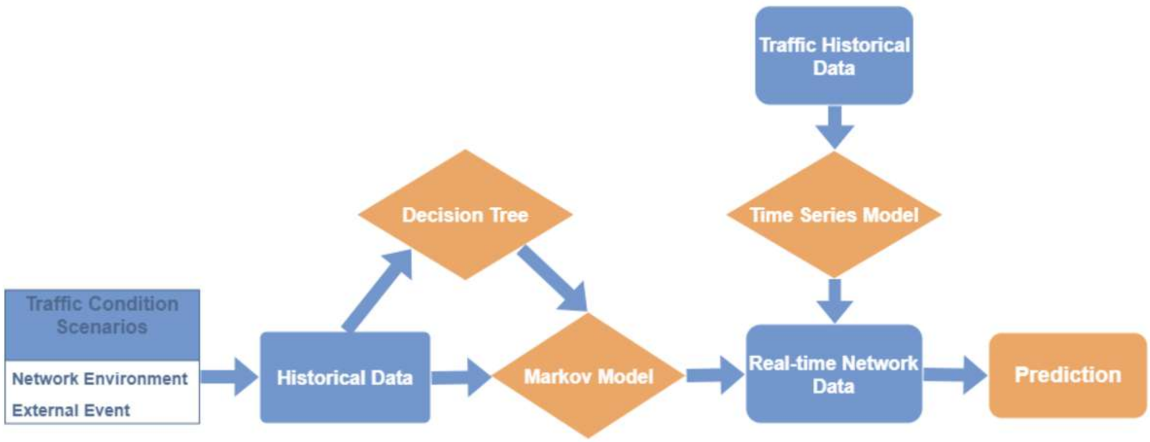
Process: The Inundation requests flood stage from the data Store on a fixed schedule. Flood stages are used to determine which inundation maps should be used for the current time domain. Then for each inundation polygon, the WayNetworks component is queried to determine if any road segments intersect the polygons and are therefore flooded. IMRCP Tiled Data files are created to save the flooded roadway data.

Dependencies: The Inundation component is dependent on inundation maps and Collectors that provide flood stage data and the WayNetworks component for road segment locations.

FORECAST:MLP PACKAGE

The Machine Learning-based Prediction (MLP) package predicts traffic network conditions given a set of system variables that include weather, work zones, incidents, and special events. MLP methods and components create a comprehensive, data-driven prediction module that uses a Markov process to explicitly characterize the probabilistic transition between traffic states under different external conditions (e.g., weather, incidents).

A Markov stochastic process is used to model the randomly evolving system with the assumption that future states depend only on the current state. A Markov transition matrix consists of a set of probabilities that are used to represent the transition probabilities between different traffic states. It is built based on archived data and can be applied online using real-time feeds to generate precise prediction models and results. With a calibrated Markov model, the probability of transition between traffic states under different external conditions can be computed. The algorithm considers that the environment variables (e.g., weather) and the external event variables (e.g., incidents) affect the transition probability matrices between different traffic states. A decision tree model constructed based on the historical data is used to determine whether the external events will affect the traffic states and whether the Markov model will be applied. The time series model takes online data as input to reflect the most current traffic conditions observed in the field. This makes the prediction model robust, particularly during special conditions that have traffic patterns that are different from regular scenarios.



Source: FHWA.

Figure 3. Machine Learning-based Prediction Network Model Algorithm.

Variables used in the IMRCP MLP model are presented in **Error! Reference source not found.**~~Table 1. Variables and State Definitions for Machine Learning-based Prediction.~~ A total of 13 variables are categorized into four groups, including network environment, external event, and traffic condition.

Table 1. Variables and State Definitions for Machine Learning-based Prediction.

Node Group	Variable	States	State Definitions
Group 1: Network Environment	Direction	-Eastbound -Southbound -Westbound -Northbound	
	Weather	-Clear -Light rain, clear visibility -Light rain, reduced visibility -Light rain, low visibility -Moderate rain, clear visibility -Moderate rain, reduced visibility -Moderate rain, low visibility -Heavy rain, reduced visibility -Heavy rain, low visibility -Light snow, clear visibility -Moderate snow, reduced visibility -Heavy snow, reduced visibility -Heavy snow, low visibility	00mm/h; visibility >3300ft < 2.5mm/h; > 3300ft < 2.5mm/h; 330 - 3300ft < 2.5mm/h; < 330ft 2.5 - 7.6mm/h; > 3300ft 2.5 - 7.6mm/h; 330 - 3300ft 2.5 - 7.6mm/h; < 330ft ≥ 7.6mm/h; 330 - 3300ft ≥ 7.6mm/h; < 330ft ; >3300ft* ; 1650 - 3300ft* ; 330 - 1650ft* ; < 330 ft*
	DayOfWeek	-Weekend -Weekday	Saturday, Sunday Monday - Friday

Node Group	Variable	States	State Definitions
	TimeOfDay	-Morning -AM peak -Off-peak -PM peak -Night	1AM - 6AM (5 hrs) 6AM - 10 AM (4hrs) 10AM - 4PM (6hrs) 4PM - 8PM (4hrs) 8PM - 1AM (5 hrs)
Group 2: External Event	IncidentDownstream	-No incident -Incident	
	IncidentOnLink	-No incident -Incident	
	IncidentUpstream	-No incident -Incident	
	Work zone	-No work zone -Work zone	
	RampMetering	-No ramp metering -Ramp metering	
	SpecialEvents	-Special event Type 1 -Special event Type 2 -Special event Type 3 -No special event	Local special events as defined by the agency, such as football games
Group 3: Traffic Management Strategies	VSL	-0 -1	- vsl off - vsl on
	Contraflow	- 0 - 1 - 2 - 3 - 4	- no contraflow - add one lane - add two lanes - close one lane - close two lanes
Group 4: Traffic Condition	Speed (mph)		

Source: FHWA.

* Snowfall's intensity is determined by visibility.

MLP Component

Background: The MLP component is responsible for producing input files, processing output files, and calling the different MLP methods that are a part of the Forecast:MLP package. Prediction methods included in the Forecast:MLP package are real-time, short-term traffic speed forecast, long time series traffic speed forecast, and Oneshot scenario-based traffic speed forecast.

Inputs: The MLP component requires current traffic speed, work zone, incident, and weather data, a long time series of previous traffic speeds, roadway segment metadata, and Scenario metadata for the Oneshot model.

Outputs: The MLP component creates short-term and Oneshot scenario-based traffic speed forecasts.

Process: On a fixed schedule, MLP requests the most recent traffic speed, work zone, incident, and weather data to create input files for the Machine Learning-based Prediction Scenario Identifier and Predictor. Multiple sets of files are created, indexed spatially, to allow for multithreaded processing. As sets of files are finished, different threads call the functions in the MLP package. After the predictions are finished, MLP processes the data into IMRCP Tiled Data Files.

When a user submits a Scenario to process, MLP requests the necessary traffic speed, work zone, incident, and weather data for the Oneshot scenario traffic speed forecast, an algorithm included in the MLP package. Input files for the Oneshot forecast are created which include the changes to metadata defined by the Scenario. The Oneshot forecast produces output files whose data can be viewed on the View Scenarios webpage.

Dependencies: The MLP component is dependent on the data Store, a calibrated Markov model, Machine Learning-based Prediction Scenario Identifier, Machine Learning-based Prediction Traffic Predictor, and Scenarios

MLP Scenario Identifier Component

Background: The Scenario Identifier identifies what specific scenario the current traffic condition belongs to and then selects a corresponding model for traffic-state prediction. This component will use an approach based on decision tree classification and pattern recognition for scenario identification. It also distinguishes different clusters which allow the Traffic Predictor to make predictions under the selected clusters or scenarios to enhance prediction performance.

This component also contains a Markov Model Generator, which computes Markov transition matrices from the archived data. The Markov model is developed based on historical traffic state transitions and is used to explain the statistically random evolutions of traffic states for each link.

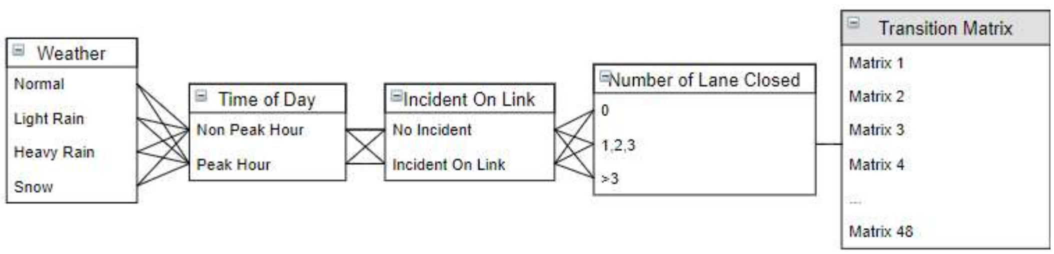
Inputs: Inputs to the MLP Scenario Identifier component include weather, roadwork, incidents, traffic control strategies, special events, and traffic state data. These datasets are obtained from the data Store using the getData method from TileObsView. They are categorized into the four groups shown in Table 4: Network Environment, External Events, Traffic Management Strategies and Traffic Condition.

Table 2. Machine Learning-based Prediction Scenario Identifier Input Categories.

Group 1: Network Environment	Group 2: External Events	Group 3: Traffic Management Strategies	Group 4: Traffic Condition
• Direction • Weather • Time of day • Day of week	• Incident downstream • Incident on this link • Incident upstream • Work zone • Ramp metering • Special events	• VSL • Contraflow	• Speed

Source: FHWA.

Outputs: The MLP Scenario Identifier component generates clustered groups of links and different traffic states based on historical traffic data. The Markov Model computes a transition probability matrix between different states for each scenario of system variables. **Error! Reference source not found.**~~Figure 4. Machine Learning-based Prediction Estimator Output Example.~~ provides an example of the output from the MLP Scenario Identifier.



Source: FHWA.

Figure 4. Machine Learning-based Prediction Estimator Output Example.

Process: This component uses an approach based on the K-means clustering algorithm and pattern recognition for scenario identification. The K-means clustering method is used to define and categorize the traffic states into different levels of congestion, from free flow to heavily congested. Each of the traffic states is assigned with a distribution of traffic speeds and volumes based on the archived data. Then roadway links are clustered into different groups based on factors such as the link volume pattern and functional type. For each detector and traffic condition scenario (based on the weather, roadwork, traffic and other data inputs), the component computes and estimates the transition matrix that includes the transition probabilities between the traffic states.

Dependencies: The MLP Scenario Identifier component depends on the weather, roadwork, incidents, traffic control strategies and traffic state data from data Store.

MLP Traffic Predictor Component

Background: The MLP Traffic Predictor is a component within the MLP package that predicts likely future network traffic states for a specified prediction horizon (such as 15 min, 30 min, 1 hour, or 2 hours) under the specific network environment and external event conditions. The

Predictor contains two parts: a One-Shot Predictor and an Online Predictor. Both Predictors use the information on current network link traffic states and other system variables, such as work zones and incidents, as model inputs to predict future network states. The One-Shot Predictor uses past-7 days' traffic state data and the forecasted next 7 days weather data to predict the future traffic state without feeding with real-time traffic data. In the One-Shot Predictor, only the future weather data are used as the scenario identifier. The purpose of the One-Shot Predictor is to provide an approximate prediction of the traffic speed for the next 7 days under forecasted weather conditions. On the other hand, the Online Predictor considers different transition probabilities between traffic states under different external conditions (e.g., weather, incident) and uses the time series model to account for the latest trends and observations from the field. Therefore, it is able to accurately predict traffic state evolution under different external conditions and adjust the prediction based on real-time field observations.

Inputs: The MLP Traffic Predictor inputs contain four groups of data. The first group is Network Environment (e.g., weather, time, date). The second is External Events, such as incidents, work zones, and special events. The third group is traffic management strategies, such as variable speed limit (VSL) and contraflow. The fourth group is real-time and archived data of traffic data as clustered into different traffic states. These inputs are similar to the inputs in **Error! Reference source not found.**~~Table 2~~. All these data inputs are current information on the current network link provided by the data Store. The outputs from the MLP Scenario Identifier also serve as input to this component.

Outputs: The MLP Traffic Predictor provides future network traffic conditions (i.e., speed) for the next specified prediction horizon (15 mins, 30 mins, 1 hour or 2 hours). For Oneshot Predictor, the prediction horizon ranges from 1 day to 7 days at a minimum time interval of 15 minutes.

Process: A time series model is applied to predict traffic speeds under normal traffic conditions. When the network environment changes or an external event occurs, the decision tree model is applied to determine whether the external events will affect the traffic speed. If yes, then the appropriate Markov transition matrix is used to predict the traffic state for the next 5-min interval. Then the speed estimate is calculated using the means of speed distributions for different traffic states (extracted from the archived data), the time series prediction, and transition matrix probabilities. For each 5-minute time interval prediction, data from the previous 7 days are taken as inputs for the time series model.

Dependencies: The MLP Predictor depends on current weather, roadwork, incidents and traffic control strategies from the data Store and the outputs from the MLP Scenario Identifier. Particularly, both archived data and real-time feeds are used.

MLPExtended Component

Background: A longer horizon for traffic speed forecast has been a point of interest for the duration of the project. To enable that, the MLPExtended component takes the Oneshot scenario traffic speed forecast and applies it to entire roadway networks without any changes to scenario variables.

Inputs: The MLPExtended component requires past and current traffic speeds, work zone, incident, and weather data and roadway segment metadata.

Outputs: The MLPExtended component creates long term traffic speed forecasts.

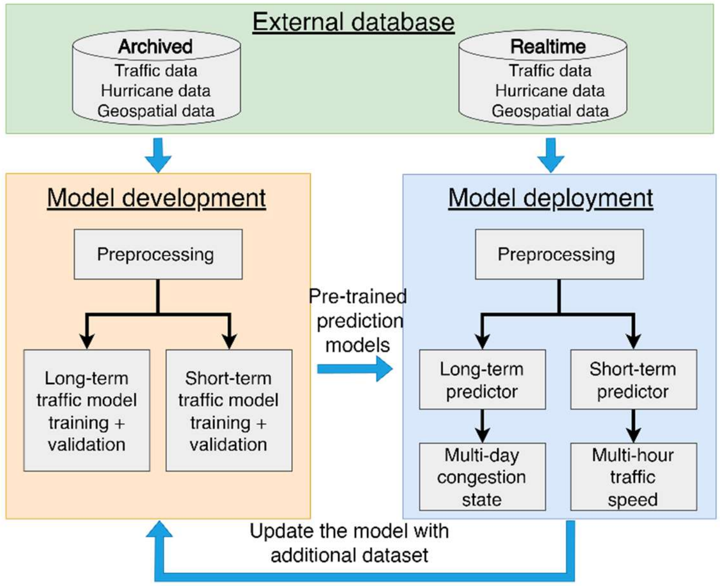
Process: On a fixed schedule, MLPExtended requests the most recent traffic speed, work zone, incident, and weather data to create input files for the Oneshot forecast algorithm. Once the algorithm has finished, MLPExtended processes the data into IMRCP Tiled Data Files.

Dependencies: The MLPExtended component is dependent on the data Store, a calibrated Markov model, Machine Learning-based Prediction Scenario Identifier, and Machine Learning-based Prediction Traffic Predictor.

MLPHurricane Component

Background: The MLPHurricane component is responsible for producing input files, processing output files, and calling MLP hurricane traffic prediction methods. MLP for hurricane scenarios predicts traffic network conditions given a set of system variables that include geospatial, hurricane forecast, and most recent traffic data. There are two components that tackle different aspects of the evacuation planning requirements: 1) a long-term prediction component, which provides a prediction with a long span into the future with low time granularity, this helps initiate the evacuation plan days before the hurricane makes landfall, and 2) a short-term model, which focuses on the near future but with high time granularity, this helps evacuation agencies to perform timely responsive operations to the most updated road traffic conditions based on the real-time traffic data feed.

Figure 5 presents the framework of the hurricane evacuation traffic prediction model developed in this phase. This model employs three primary categories of input data: traffic data, hurricane data, and network geospatial data. The input data encompasses both historical data from past hurricane evacuation events for training purposes and real-time data during active hurricane events for deployment. The model utilizes IMRCP's integrated data sources for both development and deployment phases. During the model development phase, multi-source historical data from previous hurricane events are extracted from the IMRCP system and leveraged to train both long-term and short-term prediction models. In the deployment phase, the model utilizes the pre-trained prediction model as a predictor and leverages real-time data from IMRCP to provide predictions for both the long-term congestion state and short-term traffic speed. After each new hurricane event, the data is archived and subsequently transferred to the model development component for the purpose of updating the prediction models.



Source: FHWA.

Figure 5. Graphical Representation of the Integrated Modeling for Road Condition Prediction Machine Learning-based Prediction Process for Hurricane Traffic.

Variables used in the IMRCP MLP model are presented in **Error! Reference source not found.**~~Table 3~~. A total of 12 variables are categorized into three groups, including network environment, external event, and traffic condition.

Table 3. Variables and State Definitions for Machine Learning-based Prediction for Hurricane Traffic.

Node Group	Variable	States	State Definitions
Group 1: Geospatial	DayOfWeek	-Weekend -Weekday	Saturday, Sunday Monday - Friday
	Direction	-Eastbound -Southbound -Westbound -Northbound	
	Latitude		Latitude GPS coordinate of the link centroid
	Longitude		Longitude GPS coordinate of the link centroid
	Length (m)		Link length
	Lanes		Number of lanes
Group 2: Hurricane	Hurricane Category	-1 -2 -3	The intensity of hurricane based on Saffir-Simpson Hurricane Wind Scale.

Node Group	Variable	States	State Definitions
		-4	
	Hurricane Landfall Zone	-East -West	Partition the coastline into two zones based on the central longitude line. Hurricanes are classified as 'east' or 'west' depending on the longitude of their landfall location.
	Distance to landfall		The distance from target link to the hurricane landfall location
	Time to landfall		The time from current timestamp to the hurricane landfall time
Group 3: Traffic	Speed (mph)		
	Weekly average speed (mph)		Weekly average speed on the target link from past 7 days.

Source: FHWA.

Inputs: The MLPHurricane component requires current and historic traffic speed data and hurricane forecasts.

Outputs: The MLPHurricane component creates IMRCP Tiled Data files with traffic speed predictions during hurricane events.

Process: When a new hurricane forecast is available, the forecast path of the hurricane is used to determine if any roadway segments configured for the MLPHurricane model will be affected by the hurricane. If the hurricane's path intersects a roadway network, the model is executed. The model includes short-term and long-term algorithms. For each algorithm, input files are created using current traffic speeds and the hurricane forecast. After the algorithms finish, MLPHurricane processes the data into IMRCP Tiled Data files.

Dependencies: The MLPHurricane component is dependent on the data Store, MLP Long-Term and Short Term Hurricane components, and hurricane forecasts from the National Hurricane Center.

MLP Long-Term Hurricane Component

Background: The long-term model's primary objective is to offer multi-day predictions concerning the location and timing of congestion during a hurricane. We adopt a 7-day time span as the prediction duration for long-term congestion prediction during hurricane evacuation and recovery. To encompass traffic patterns both before and after the hurricane, the 7-day prediction horizon is defined as follows: 3 days before landfall, the day of landfall, and 3 days after landfall.

The long-term model aims to predict the timing of congestion periods on each road segment across the hurricane-impacted 7-day range in the whole network. Therefore, we break down the 7-day time span into 28 6-hour time periods and categorize the congestion state into clusters with different congestion labels, then transform the congestion state prediction problem into a multi-class classification problem by predicting the congestion label for each of the 6-hour periods in the 7-day range. The congestion states are defined using the speed performance index (SPI) shown in (1), where v_{6hour} is the average speed of a 6-hour period, and v_{weekl} refers to the weekly average speed. Congestion states are categorized as heavy congestion (SPI < 50%), light congestion (50% < SPI < 75%), and no congestion (SPI > 75%). The predicted label can further be used to infer the aggregated 6-hour speed values based on SPI and the weekly average speed of the target link segment.

$$SPI = \frac{v_{6ho}}{v_{weekly}} \times 100\% \tag{1}$$

Inputs: Inputs to the MLP Hurricane Traffic Impact Identifier component include link coordinates, data record time, hurricane information, and traffic state data. These datasets are obtained from the data Store. They are categorized into the three groups shown in **Error! Reference source not found.**~~Table 3~~: Network Environment, Hurricane Condition and Traffic Condition.

Outputs: The direct model output is a set of congestion state labels for each 6-hour period across 7 days. This output can be further transferred as an approximation of traffic speed using SPI and the link segment's weekly average speed.

Process: The model training and prediction process is presented in Figure 6. Firstly, the input layer serves as the initial processing stage, receiving and transmitting input data to subsequent layers. Each node (neuron) in this layer corresponds to a specific feature or attribute present in the input data. The input features utilized in our model can be categorized into four distinct groups. These groups encompass link features, such as the number of lanes, directions, and non-evacuation average speed, providing essential transportation link characteristics. Spatial features, including latitude, longitude, and distance to the landfall location, contribute valuable spatial context. Temporal features, such as the time of day and time to landfall, offer crucial temporal information. Lastly, the hurricane features, encompassing forecast hurricane category and potential landfall location, provide essential hurricane-related data. Secondly, the hidden layer plays a pivotal role as intermediary layers responsible for information processing. Each node in a hidden layer receives inputs from all nodes in the preceding layer and forwards its output to all nodes in the subsequent layer. This configuration allows the neural network to learn intricate patterns and relationships present in the data across different categories of features in the input layer. Lastly, the output layer generates the final predictions or outputs of the model. In our long-term prediction model, the output layer offers predictions among three congestion labels: no congestion, light congestion, and heavy congestion. This final prediction from the output layer reflects the model's evaluation of the congestion state during the hurricane evacuation process.

Dependencies: The MLP Hurricane Traffic Impact Identifier component depends on the network, hurricane, and traffic state data from data Store.

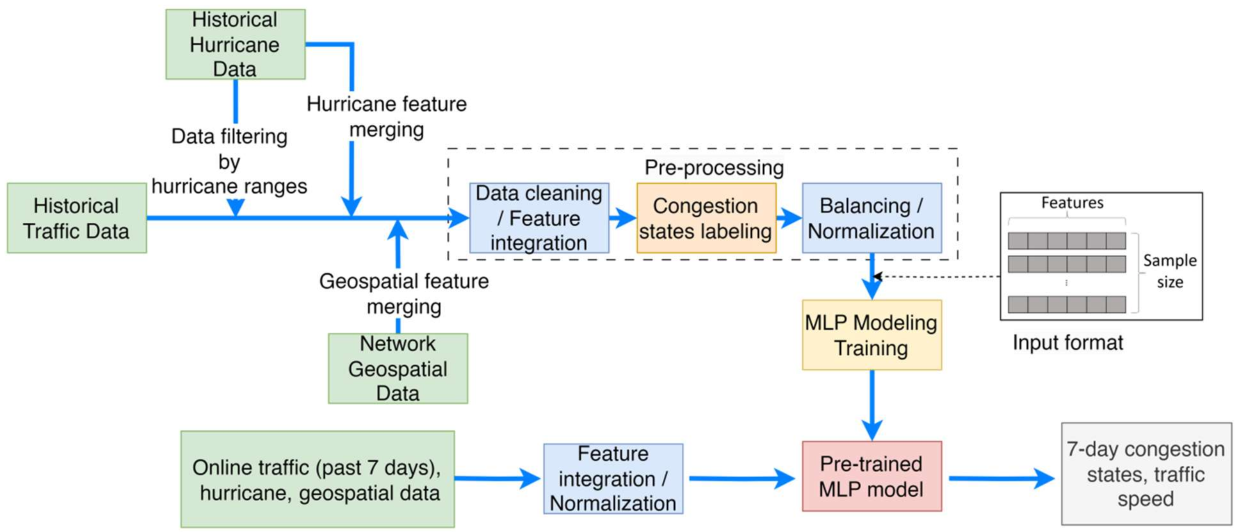


Figure 6. Model framework for long-term hurricane traffic prediction model

MLP Short-Term Hurricane Component

Background: The short-term model focuses on accurately predicting speed values for each link during the 7-day hurricane impact horizon. It aims to forecast speeds for a short-term horizon ranging from one to several hours into the future, given a particular start time. The short-term speed prediction is framed as a many-to-one time series regression problem, where the input includes time sequence data (e.g., speed data for the last 24 hours) along with link attributes, spatial-temporal attributes, and the latest hurricane attributes. The output is the predicted speed value after the specified prediction horizon from the current time step.

Inputs: The input consists of various features. These features include time-varying features such as traffic speed, time to landfall, time of day, and static features such as hurricane category, hurricane landfall zone, link coordinates, distance to landfall, and number of lanes. All these data inputs are current information on the current network link provided by the data Store.

Outputs: The objective is to predict a single future speed value at the next time step or next few time steps, ranging from 1 to 6 hours ahead.

Process: Figure 7 illustrates the model framework for the short-term traffic speed prediction component. The short-term component utilizes data resources similar to those of the long-term component as input. Following the integration of features, a crucial step in the short-term component involves partitioning the raw time-series data into small sequential samples of equal sequence length. In this stage, we utilize pre-generated labeled data from the long-term component to ensure a balanced representation of time-series samples across different congestion states. After balancing and normalizing the pre-processed data, we then proceed to feed the data into the model training procedure. The short-term prediction adopts a Long-Short-Term Memory Neural Network (LSTM), an advanced form of Recurrent Neural Network (RNN). LSTM is specifically designed to handle sequential data, making it highly effective in capturing long-term dependencies. These characteristics render LSTM a suitable choice for real-time traffic prediction during hurricane and evacuation scenarios. The processed input data are passed to the trained LSTM model and generates the speed prediction for the next 1 to 6 hours depending on the given prediction horizon.

Dependencies: The MLP Predictor depends on the current network environment, hurricane, and traffic state data from the data Store and the outputs from the Machine Learning-Based Long-Term Hurricane Traffic Predictor. Particularly, both archived data and real-time feeds are used.

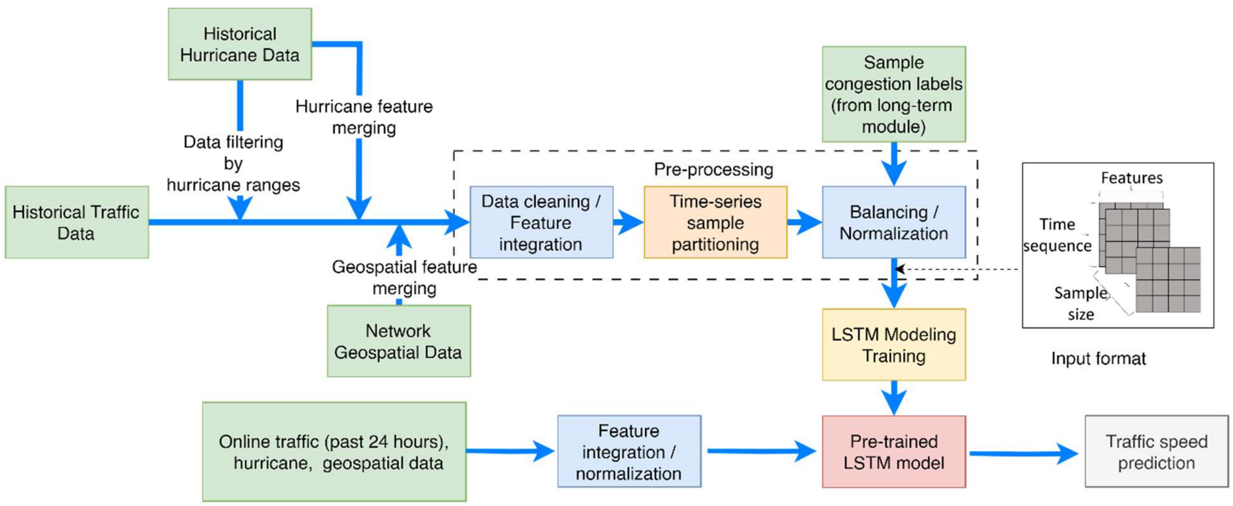


Figure 7. Model framework for short-term hurricane traffic prediction model

FORECAST:METRO PACKAGE

METRo Component

Background: The METRo model was developed by The Canadian Meteorological Center of Environment Canada as a standard pavement thermal modeling tool that is part of its road weather forecasting suite. It is widely used and adapted in many winter maintenance decision support systems. IMRCP uses METRo as its Road Temperature and Snow Module. Performance and model improvements have been made to METRo inside of IMRCP to allow for wide spread predictions that keep track of the water and snow/ice reservoirs from one prediction cycle to the next.

Inputs: The METRo component takes many atmospheric weather parameters (air temperature, dew point, wind speed, precipitation rate, precipitation type, pressure, and cloud cover) both observed and forecasted, to run. Pavement state, pavement temperature, subsurface temperature, water reservoir, and snow/ice reservoir results from previous METRo predictions are used to initialize the state for subsequent predictions, if there is not another source providing those observations. Metadata for road segments including geo coordinates and if the segment is a bridge is needed as input to the model as well.

Outputs: The METRo component saves pavement state, pavement temperature, subsurface temperature, water depth, and snow/ice depth predictions to a file and sends a notification to the Directory that new data is available.

Process: The METRo component runs on a schedule, executing every 10 minutes in the current deployment. Atmospheric and road weather observations and forecasts, including the predictions and water and snow/ice reservoir levels from the previous METRo run, are gathered from the data Store using the getData and getReading interfaces for each segment in each network. To avoid extra processing, segments are categorized by location and whether or not they are a bridge. The model is then run for each categorized location and the results are applied to each segment in that category.

Dependencies: The METRo component is dependent on the data Store for providing all of the needed atmospheric and road weather observations and forecasts. It is also dependent on the WayNetworks component for metadata describing the segments that are used to determine the locations to run the METRo model.

METRo Scenario Assessment [MetroProcess] Component

Background: The Scenarios component uses the METRo model to make 24-hour pavement state and temperature predictions for the roadway segments included in created scenarios.

Inputs: MetroProcess require many forecasted atmospheric and road weather parameters (air temperature, dew point, wind speed, precipitation rate, precipitation type, pressure, cloud cover, road temperature, subsurface temperature, and pavement state) and whether roadway segments are to be treated or plowed.

Outputs: MetroProcess creates long-term pavement state and temperature predictions.

Process: After a Scenario is created and queued to process, the Scenarios component instantiates a MetroProcess object, passing it the configured actions of the Scenario, namely which roadway segments are to be plowed or treated and at what time. For each roadway segment in the Scenario, the METRo model is run 24 times, once for each hour of the Scenario. The outputs of each hour's run are used as inputs for the next hour's run.

Dependencies: MetroProcess is dependent on the data Store, roadway segment definitions, and the Scenarios component.

GEO SERVICES [GEOSRV] PACKAGE

The Geo Services (GeoSrv) package maintains the fundamental geographical description of the roadway system and its components.

WayNetworks Component

Background: The WayNetworks component defines and describes the different deployment networks and all of the ways (roadway segments) contained in each network. A way is a set of two or more points called nodes. Metadata for each way, including number of lanes, speed limit, and elevation data, is also stored by the component.

Inputs: The WayNetworks component creates and uses the networks configuration file. Requests from the NetworkGeneration webpage are made to create, finalize, reprocess, and delete networks.

Outputs: The WayNetwork component saves any updates to a network in the network configuration file and serves network and way definitions to other components.

Process: At system start up, the WayNetworks component reads the networks configuration file and loads the existing network and way definitions into memory to be available for other system components. Queries for way objects can be made by id or geo location. System administrators can use the Network webpage to manage networks; operations include creating, finalizing, reprocessing, and deleting a network. Creating a network has this component save the network's bounding polygon, label, and filter options to the network configuration file. Finalizing a network runs algorithms (merge, split, separate) designed to convert the existing roadway network into a more robust transportation model. Reprocessing a network flags that it is no longer finalized and queues it to be reprocessed which allows the network to get any updates from the OSM database. Deleting a network removes the network from the networks configuration file.

Dependencies: The WayNetworks component is dependent on networks being created by an IMRCP Administrator using the Network webpage, which uses Open Street Map way and node definitions as a starting point for creating the ways and nodes used by IMRCP.

DEM Component

Background: Mapbox provides a raster tileset that contains global elevation data. The Mapbox Terrain-DEM contains raw height values stored in PNG tiles that can be decoded to raw heights in meters. To have access to the API, a Mapbox access token is required. IMRCP downloads and caches the elevation tiles on demand. Tiles are only cached for a configured time, currently one month, so infrequent elevation changes are handled as needed. The Digital Elevation Map (DEM) is used to lookup the elevation for a particular geo coordinate when the elevation is needed and not otherwise provided directly by a data source.

Inputs: DEM takes a geo-coordinate (lon/lat pair) as input.

Outputs: DEM returns the elevation at the given geo-coordinate in meters.

Process: When an elevation request is made, first the geo-coordinate is converted into map tile coordinates at a specific zoom level and generates the file name of the desired tile. Next the component checks if the file is cached or if it needs to be updated because it is out of date. If needed, the tile is downloaded from the Mapbox API and cached on the server. Finally, the corresponding pixel in the PNG of the geo-coordinate is determined and its value is decoded and returned as an elevation in meters.

Dependencies: The DEM component is dependent on the Mapbox API.

Mercator Component

Background: Most open source and commercial Maps API providers use the Spherical Mercator projection coordinate system since it is easy to work with and preserves shapes and angles. Therefore, IMRCP contains the Mercator component which provides convenience methods for converting lon/lat, Mercator, and map tile coordinates from one coordinate system to another.

Inputs: The Mercator class accepts lon/lat, Mercator, or map tile coordinates.

Outputs: The Mercator class returns lon/lat, Mercator, or map tile coordinates.

Process: The Mercator class converts lon/lat, Mercator, and map tile coordinates from one coordinate system to another.

Dependencies: The Mercator class does not have any dependencies.

OsmBz2ToBin Component

Background: The definitions of roadway segments in IMRCP are derived from the Open Street Map database. OSM files for entire states can be downloaded in an XML format compressed by the bz2 algorithm. These files end up being very large and take a considerable time to open and process. Therefore, IMRCP converts these files into a binary representation that is much more compact and faster to process.

Inputs: OsmBz2ToBin requires an OSM .xml.bz2 file.

Outputs: OsmBz2ToBin saves a binary representation of an OSM .xml.bz2 file.

Process: OsmBz2ToBin decompresses and loads an OSM .xml.bz2 file into memory which includes the definitions of ways and nodes and their associated set of key-value tags. While the ways and nodes are being processed a string pool of all the tags is created as well as a hashed spatial index for the ways. The string pool is written to the file first. The nodes and ways are written next, referencing any tag by index, instead of writing the same string multiple times. Ways reference nodes by file position, instead of by id. A separate binary index file is written which stores the hashed spatial index which allows for quick lookup of ways given a geo-coordinate.

Dependencies: OsmBz2ToBin objects are dependent on the OSM database

OsmBinParser Component

Background: OsmBinParser is used to load the files created by the OsmBz2ToBin class.

Inputs: OsmBinParser uses OSM .bin and hashed spatial index files.

Outputs: OsmBinParser creates OsmWay and OsmNode objects as outputs.

Process: OsmBinParser can load an entire OSM .bin file or a specific part of it based off of a given geo-coordinate using the hashed spatial index file, creating the OsmWay and OsmNode objects defined by the files.

Dependencies: OsmBinParser objects are dependent on OSM .bin and .bin.ndx files

ProjProfiles Component

Background: Geodetic information for observations and forecasts is stored in different projection coordinate systems throughout the data Store. To save memory and processing time, projection profiles common among the sources are cached in memory. For example, all data files from RTMA can use the same projection. This component manages the creation and caching of the system's projection profiles.

Inputs: ProjProfiles uses projection coordinate system grids.

Outputs: ProjProfiles returns projection profile objects.

Process: When a data file is loaded into memory that requires a projection profile to determine its spatial information, the projection coordinate system grid is passed to ProjProfiles. If a projection profile for that grid does not exist, ProjProfiles creates and caches the projection profile. Now ProjProfiles can serve that projection profile to data files in the system.

Dependencies: The ProjProfiles component does not have any dependencies.

WEB PACKAGE

SecureBaseBlock Component

Background: To ensure that only known users have access to the system and to prevent potential distributed denial-of-service (DDoS) attacks, SecureBaseBlock, a child class of BaseBlock, was developed with security algorithms to be used for every request made from the webpages. User privileges and access are assigned by a system administrator.

Inputs: SecureBaseBlocks accept GET or POST request URLs.

Outputs: SecureBaseBlock component outputs are specific to the implemented child classes.

Process: GET or POST requests are passed to a SecureBaseBlock from the web server's container manager based on the URL pattern. First the SecureBaseBlock ensures that the request comes from a known, logged-in IMRCP user and that they have the correct permission for the component. Next the SecureBaseBlock determines if the users have sufficient privileges for the operation. If both are true, then the request is processed; if not, the request is ignored.

Dependencies: SecureBaseBlocks are dependent on the web server's container manager to pass them requests.

NetworkGeneration Component

Background: A robust and well-defined roadway network is essential to a successful IMRCP deployment. To assist in the arduous task of creating a roadway network, a set of tools and algorithms were developed to help automate the task as much as possible.

Inputs: NetworkGeneration accepts a polygon that defines the border of the desired network and set of options to start the creation of a roadway network.

Outputs: A finalized roadway network is the product of NetworkGeneration after a created network has any necessary actions and edits performed on it.

Process: Using the Network webpage an IMRCP administrator can create a new roadway network and perform actions on the segments in the network. Actions include add, remove, merge, and split segment. After all desired actions are performed the network can be submitted to be finalized which runs multiple algorithms (merge, split, separate) on the segments to convert the existing roadway network into a more robust transportation model.

Dependencies: NetworkGeneration is dependent on the Open Street Map database to extract roadway segment definitions and metadata.

SessMgr Component

Background: Only authorized users with a valid session are allowed to access the system's user interfaces. SessMgr manages registered users, passwords, permissions, and browser sessions.

Inputs: The users file, username, password, and security token are inputs to SessMgr.

Outputs: Session objects with security tokens are generated by SessMgr.

Process: The users file is read at system startup to get the list of registered users' usernames, encrypted passwords, and permissions. Registered users log into the system by providing their username and password. If valid credentials are provided, a Session is created in memory and a security token is passed back to the browser and used for the remainder of the user's Session to authenticate access to other pages and interfaces. Sessions are removed from memory when a user logs out or is inactive for a specified amount of time.

Dependencies: The SessMgr component is dependent on a System Administrator using the main method outside of the system to create and add records to the users file.

Scenarios Component

Background: One of IMRCP's purposes is to assist in planning how to use resources during storms, scheduled maintenance, and rush hour traffic. To achieve this purpose, the Scenarios tool was developed, which allows users to create Scenarios and generate a 24-hour prediction for road conditions and traffic based on associating actions (plowing, chemical treatment, Variable Speed Limit change, and number of available lanes change) with segment groups.

Inputs: The Scenarios component receives segment groups with associated time series of actions and a reference time.

Outputs: The Scenarios component generates 24-hour predictions of road conditions and traffic speeds.

Process: The Scenarios component receives a request to save a scenario template from the Create Scenarios webpage which includes the definitions of the segment groups and associated time series of actions for each group of the Scenario. Scenario templates can then be edited via the Create Scenarios webpage or given a reference time to process. Requests are processed in serial and generate a 24-hour prediction of road conditions and traffic speeds that can be viewed via the View Scenarios webpage.

Dependencies: The Scenarios component is dependent on the data Store and the METRo and MLP models to provide predictions.

Subscriptions Component

Background: Subscriptions enables IMRCP to provide data reports and subscriptions to external systems and end users. One-time reports can be helpful in analyzing past events while recurring subscriptions can be used to receive the most recent forecasts at a regular interval.

Inputs: The Subscriptions component receives information necessary to create a report or subscription. This includes the requested observation types, geographic context (a set of roadway segments), reference time, offset time, and duration.

Outputs: The Subscriptions component generates data reports to fulfill the requested reports or subscriptions. Available data reports are displayed and can be downloaded via the View Reports webpage.

Process: The Subscriptions component runs on a regular schedule. Each period of execution, it checks to see if there are any reports or subscriptions that need to be fulfilled and processes them. For each report or subscription being processed, a request for the desired data is made to the data Store and results are formatted into a data report.

Dependencies: The Subscriptions component is dependent on the data Store and the Create Report webpage.

LayerServlet Component

Background: Detailed data requests from the Map webpage are processed by different LayerServlets depending on the geographic extents of the request, which can be a single point, polyline (roadway segment), or polygon.

Inputs: The LayerServlet components receive a reference time, query time, and geographic extent.

Outputs: The LayerServlet components return a list of Observation objects.

Process: Users can left-click on point, road, or area layers on the Map webpage to request detailed data at that location. LayerServlet components query the data Store for the desired time and location and return a list of all observations that match the query.

Dependencies: The LayerServlet components are dependent on the data Store and the web server's container manager to pass them requests.

TileServlet Component

Background: Creating data views for the Map webpage can be done on the fly for each request. TileServlet components create data views for observations associated with a single point, roadway segments, or areas described by a polygon.

Inputs: The TileServlet components receive a reference time, query time, tile coordinates (x, y, zoom), observation type, and geometry type.

Outputs: The TileServlet components return vector tiles that represent data requested for the given tile coordinate.

Process: The Map webpage sends reference time, query time, tile coordinate, observation type, and geometry type requests to TileServlet components. To process a request, the data Store is queried for the desired time and matching data is processed into a vector tile view using configured RangeRules to bucket similar values together. The vector tiles are returned to fulfill the current request.

Dependencies: The TileServlet component is dependent on configured RangeRules and the data Store.

DashboardServlet Component

Background: The Dashboard webpage allows users to monitor specific locations and observation types within the system. This servlet handles all of the requests from the Dashboard.

Inputs: The DashboardServlet receives requests from the Dashboard webpage.

Outputs: The DashboardServlet sends responses to the Dashboard webpage

Process: The Dashboard webpage has multiple operations that the DashboardServlet handles. Each operation has a corresponding request including queuing/fulfilling data requests, geojson defining the locations, adding/editing/deleting an alert, and getting the status of each alert.

Dependencies: The DashboardSerlvet is dependent on configured alerts made by the user and the data Store.

UserManagementServlet Component

Background: The Manage Users webpage allows IMRCP Administrators to create, edit, and disable system users. This servlet handles all of the requests from the Manage Users webpage.

Inputs: The UserManagementServlet receives requests from the Manage Users webpage.

Outputs: The UserManagementServlet sends responses to the Manage Users webpage and updates both the in-memory and on-disk list of users and ensures they are consistent.

Process: IMRCP Administrators send requests to create, edit, and disable users from the Manage Users webpage. The in-memory and on-disk list of users is maintained through these requests.

Dependencies: The UserManagementServlet component is dependent the web server's container manager to pass them requests.

WEB PAGES PACKAGE

The WebPages package provides the primary graphical user interfaces to the IMRCP system outputs. Only registered users have access and permission to use them.

LogOn Webpage

Background: The LogOn page is the first page users view when accessing IMRCP and allow them to submit their username and password to log into the system. It can also be used to display system messages, like scheduled maintenance times, to users.

Inputs: The LogOn webpage receives a username and password.

Outputs: The LogOn webpage returns a security token used to authenticate other API calls for the current session.

Process: Users enter their username and password to submit them for authentication. If valid credentials are provided, a security token is returned and stored by the browser to authenticate other pages and API calls for the user's current session.

Dependencies: The LogOn webpage is dependent on the SessMgr component to authenticate credentials and generate a security token.

Map Webpage

Background: The Map webpage provides a selectable, layered presentation of archived, current, and forecast traffic and weather conditions across the roadway networks. Map layers are described in Appendix B. Alert layers are further described in Appendix D.

The map provides several sets of controls for setting the view context. Typical map interactions (e. g., pan, zoom) set the spatial context for the view. Time controls set the temporal context for the view which consist of a reference time and query time. Using the calendar control, the reference time can be set, meaning that only data that was available at or before that time will be displayed. Date and time sliders are available at the bottom of the map to select the query time. Going back in time from the reference time enables viewing observations and model results as they were current at that time. Going forward in time from the reference time enables forecast data to be viewed as it was available at that reference time. Layer controls set the map overlays, icons, and display options. Selection categories are based on map graphical constructs. Layers available for each category correspond to particular data elements available from the data Store. “Point” constructs are used to locate and identify observations from sensors (mobile and stationary) and alerts to localized events (e.g., incidents on roadways). “Road” constructs are used to locate and describe attributes of roadway segments including traffic and pavement conditions. “Area” constructs are used to represent areal attributes such as atmospheric conditions and weather alert areas. Users can left-click on any construct to receive detailed data at that location for the current reference and query time in a table view.

Two predefined views are available to users to automatically select related observations types. The “Tropical Storms” view turns on the Tropical Storm Cone, Tropical Storm Category, and Wind Speed layers. The “Winter Storms” view turns on the Precipitation Rate and Type and Pavement State layers.

Map “Settings” enable to the user to select whether the map refreshes itself to maintain the reference time at “now”, to adjust the opacity for area layers, to display the longitude/latitude coordinate of the cursor, and to select the default view when they log in: either Dashboard or Map. Users can also save their settings causing the map to remember which layers are selected, the current map position, and zoom level for subsequent uses of the map.

Inputs: The Map webpage takes input from users using mouse and keyboard to interact with the different controls to set the spatial and temporal contexts and view data.

Outputs: The Map webpage displays archived, current, and forecast traffic and weather data on a map and table.

Process: After a user logs on to the system, the Map webpage is the default page. Users interact with the controls to view data for different spatial and temporal extents.

Dependencies: The Map webpage is dependent on LayerServlet and TileServlet components to provide the different views of system data.

Dashboard Webpage

Background: Dashboards are helpful for seeing an overview of a system. Since IMRCP covers different locations depending on the deployment, a default view for a dashboard is difficult to define. Because of this, users are able to configure different locations for the Dashboard to monitor. Alerts can be made for any location and any observation type and are saved so that a user will always see their configured Alerts on the Dashboard. For each location and observation type, thresholds can be defined to make it easier to determine there are areas of concern within the network. The Dashboard keeps a current view of the data needed to determine if any configured thresholds are met.

Inputs: Users define Alerts and thresholds for areas of interest in their network. Inputs include locations (region or set of roads), observation types, comparison operators, and value thresholds.

Outputs: A table and map view of configured Alerts which includes a color indicator for the severity, how long until the Alert threshold is predicted to trigger, the number of road segments affected, and the total number of linear miles affected for each Alert.

Process: Once a user has defined at least one Alert, requests are made to the server for the current data for the configured observation types. Each data point is evaluated based on the comparison operator and value threshold the user defined. A summary of the data is then displayed in the table. Clicking on an entry in the table will show the location of that Alert on the map. As long as the user is viewing the Dashboard webpage, the data with automatically be updated as new data is available within the IMRCP system.

Dependencies: The Dashboard webpage is dependent on the data Store to provide the data sets for the user configured alerts.

Create Scenario Webpage

Background: The Create Scenario webpage provides a map interface to create and edit scenario templates which can then be given a name and reference time and submitted to process, generating a 24-hour prediction for road and traffic conditions.

Inputs: The Create Scenario webpage takes input from users using mouse and keyboard to interact with the controls.

Outputs: The Create Scenario webpage lists current scenario templates and their details.

Process: By following the instructions found in the dialog box, users create scenario templates by providing a name and creating roadway segment groups. Once a group is created, users associate a time series of actions to be applied to each group. Scenario templates are saved so that can be edited later or processed for different reference times. When a scenario template is complete, users enter a name and select a reference time to submit the template as a scenario to be processed. The 24-hour predictions that are generated by a processed scenario can be viewed via the View Scenarios webpage.

Dependencies: The Create Scenario webpage is dependent on the WayNetworks component to display the available networks and roadway segments and the Scenarios component to process all of the requests made by users.

View Scenarios Webpage

Background: The View Scenarios webpage provides a map and table interface to view the 24-hour predictions for road and traffic conditions generated by a processed scenario.

Inputs: The View Scenarios webpage takes input from users using mouse and keyboard to interact with the controls.

Outputs: The View Scenarios webpage lists scenarios that have been processed and displays the predicted road and traffic conditions on the map and in a table.

Process: Users select which Scenario they would like to view. They can select which observation type to be displayed on the roadway segments on the map. The time to view can be selected by the time slider at the bottom of the map. By clicking on a roadway segment, a table will be displayed containing data and metadata for that segment.

Dependencies: The View Scenarios webpage is dependent on the WayNetworks component for roadway segment geometry definitions and the Scenarios component to serve the predictions created for each Scenario.

Create Report Webpage

Background: The Create Report webpage provides a map interface to create data reports and subscriptions.

Inputs: The Create Report webpage takes input from users using mouse and keyboard to interact with the controls.

Outputs: The Create Report webpage provides the URL to download data reports.

Process: By following the instructions found in the dialog box, users create a report or subscription by first selecting the roadway segments they would like included in their report or subscription. Next users provide a name and select up to five observation types. In the same dialog, users choose whether they are creating a report or a subscription and select the reference time, offset, and duration. Once all the parameters are set, the report or subscription can be submitted to be processed.

Dependencies: The Create Report webpage is dependent on the WayNetworks component to display the available networks and roadway segments and the Subscription component to process all of the requests made by users.

View Reports Webpage

Background: The View Report webpage provides a list interface to view and download available data files for reports and subscriptions.

Inputs: The View Report webpages take input from users using mouse and keyboard to make selections on the interface.

Outputs: The View Reports webpage provides data files to download.

Process: A list of pending and available reports and subscriptions is displayed when the page is accessed. Left-clicking on a report downloads the data file for that report. Selecting a subscription will list all of the available data files associated with that subscription. Each data file can be downloaded by left-clicking on its name.

Dependencies: The View Reports webpage is dependent on the Subscriptions component to serve the data files created to fulfill each report and subscription.

Manage Roads Webpage

Background: The Manage Roads webpage provides a map interface to create, edit, and publish roadway networks. Hurricane Traffic model training for a roadway network is initiated on this page. Only IMRCP Administrators have permission to access this page.

Inputs: The Manage Roads webpage takes input from users using mouse and keyboard to interact with the controls.

Outputs: The Manage Roads webpage displays all of the networks and metadata about them.

Process: When the page is first accessed, the bounds of all existing networks are displayed on the map. Selecting the bounds of a network loads the detailed geometry of the network and allows the user to enter different modes to edit the network or initiate Hurricane Traffic model training. Roadway segments can be added or removed. Once all desired edits are complete, the network can be published, making it accessible to other users. The roadway segments in a published network will also be used in the MLP and Metro forecasts if the network is configured that way. Users can export the network as an OSM .xml file at any time.

Dependencies: The Manage Roads webpage is dependent on the Open Street Maps database and the WayNetworks component to process all of the requests made by users.

Manage Users Webpage

Background: The Manage Users webpage provides a table interface to create, edit, and disable IMRCP users. Only IMRCP Administrators have permission to access this page.

Inputs: The Manage Users webpage takes an email address as a username and a string for a password as input.

Outputs: The Manage Users webpage displays a table containing all users within the system, their user group, and if they are disabled.

Process: IMRCP Administrators can click the Add User button or double click a user in the table and follow the subsequent dialogs to make edits to the user table.

Dependencies: The Manage Users webpage dependents on responses from the UserManagementSerlvet Component.

SYSTEM COMPUTING INFRASTRUCTURE

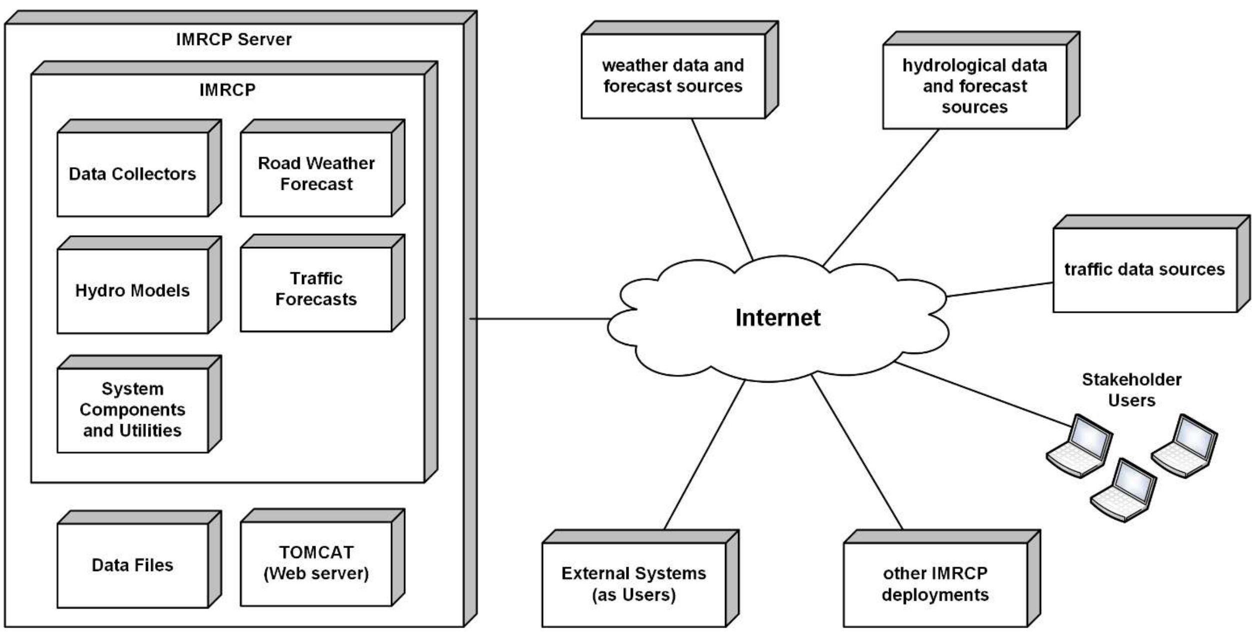
The deployment model for IMRCP has been demonstrated in prior phases for statewide networks and applications. As such, this phase is demonstrating parallel deployments and view. The deployment model is being retained in order to minimize system management overhead, which is particularly helpful for agency initiated deployments that focus on adaptation and configuration for their data environments rather than on new development. The deployment model provides distributed user access for the development team, the review team, and agency partners. The following factors contribute to determining this configuration:

- IMRCP data services and computational services are closely linked to and benefit from co-location to reduce latencies and remote network calls.

- Other data sources, (for example, atmospheric weather and hydrology) are provided by external web services that do not drive any particular deployment solution.

- Potential future phase operational deployments might be linked to transportation management centers (TMCs) and integrated management solutions. The bulk of the real-time operational and traffic data comes from transportation management systems, and the majority of the end users are either agency personnel or travelers for whom data is already sourced from TMCs and their associated systems. It makes sense within that context to anticipate and demonstrate a deployment as a forecast "appliance" rather than a distributed system with the potential limitations of traffic predictions as external services.

The demonstration system deployment is shown in Figure 8~~Figure 8~~. The IMRCP system software and the Apache Tomcat Web server will be deployed on a common server. The server will use a high-bandwidth connection to the Internet to access data contributors and to provide access to IMRCP forecast products for stakeholders and systems. This configuration will be subject to review and re-evaluation during the development process to assure the project and system needs are being met.



Source: FHWA, 2018.

Figure 8. Integrated Modeling for Road Condition Prediction Deployment.

CHAPTER 4. REFERENCES

ISO/IEC/IEEE. 2023. *Systems and Software Engineering – System Life Cycle Processes.* ISO/IEC/IEEE 15288:2023

Leidos. 2025. *Integrated Modeling for Road Condition Prediction Installation and Administration Guide*. Unpublished working paper developed under FHWA BPA 693JJ322A000005, Task Number HOTO220039PR, Integrated Modeling for Road Condition Prediction (IMRCP) – Phase 5.

Leidos. 2015. *Integrated Modeling for Road Condition Prediction Model Analysis*. 2015. Unpublished working paper developed under FHWA Contract DTFH61-12-D-00050, Task Order 5022, Integrated Modeling for Road Condition Prediction.

Leidos. 2015. *Integrated Modeling for Road Condition Prediction Concept of Operations*. Unpublished working paper developed under FHWA Contract DTFH61-12-D-00050, Task Order 5022, Integrated Modeling for Road Condition Prediction.

Leidos. 2016. *Integrated Modeling for Road Condition Prediction System Requirements.* Unpublished working paper developed under FHWA Contract DTFH61-12-D-00050, Task Order 5022, Integrated Modeling for Road Condition Prediction.

Lukasik, et al. 2011. *Assessment of Emerging Opportunities for Real-time, Multimodal Decision Support Systems in Transportation Operations*. Report Number FHWA-JPO-10-058. Washington, DC: FHWA.

APPENDIX A. INPUT DATA AVAILABILITY

All collectors poll for new files or updated data feeds on a regular schedule. Some sources, primarily National Weather Service forecasts, store multiple days' worth of files. Those collectors try to download any file that IMRCP has not collected by comparing the list of file available and files already stored in IMRCP's persistent storage. The polling frequency does not always match the expected availability because files are not always available at the exact same offset between collection cycles due to processing and uploading speeds.

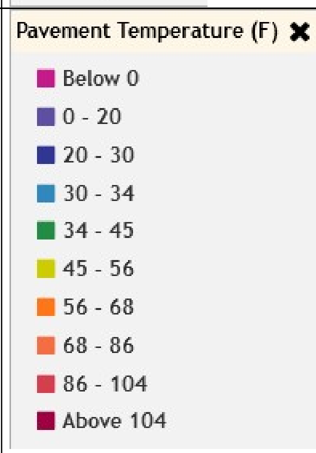
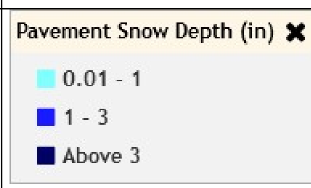
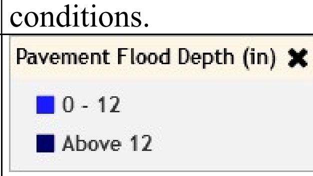
Table 4. Input Data Available Frequency.

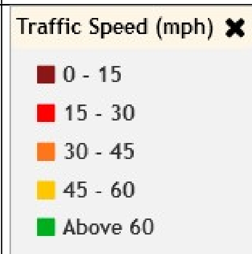
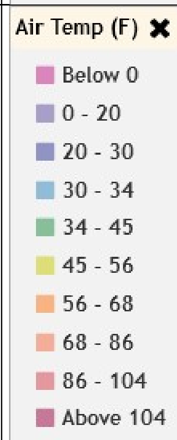
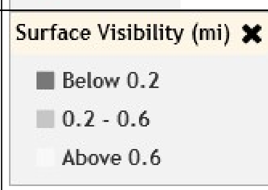
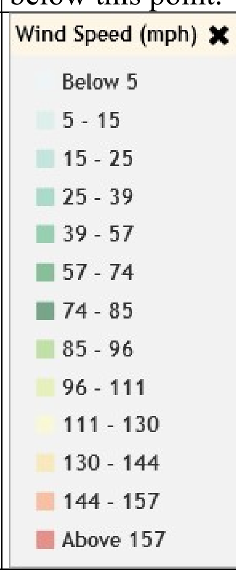
Source	Polling Frequency	Expected Availability	Time range
ADCIRC	30 minutes	New files 4 times a days	up to 180-hour forecasts (IMRCP uses hours 6-120)
CAP	5 minutes	Updated possibly every minute	Alerts and watches up to 3 days
GFS	6 hours, retries after 20 minutes if an error occurs	New files 4 times a day	up to 384-hour forecasts (IMRCP hours every 3rd hour in-between hours 54 and 168)
MRMS	1 minute	New files every 2 minutes	2-minute observations
NDFD	10 minutes	New files every 1 hour	up to 66-hour forecasts starting 1 hour after collection
NHC	30 minutes	New files approximately every 6 hours when there is an active tropical storm	120-hour forecasts
NWM	5 minutes	New files every hour	120 hour forecasts
NWPS	5 minutes	Updated 4 non-regular times an hour	1 hour for observations 24 hours for forecasts
RAP	10 minutes	New files every 1 hour	21-hour forecasts starting at collection time
RTMA	10 minutes	New file every 1 hour	1 hour forecast starting at collection time
WxDE	10 minutes	Updated every 10 minutes	up to 1-hour observations

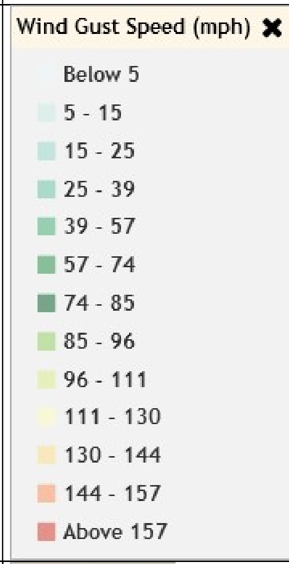
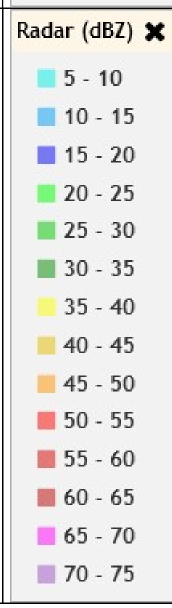
Source: FHWA

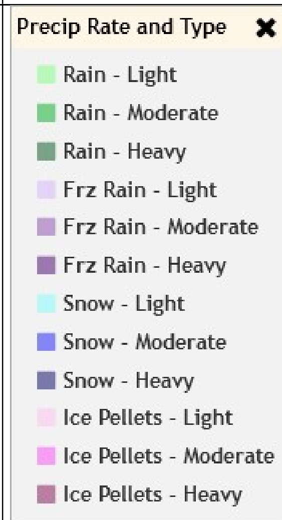
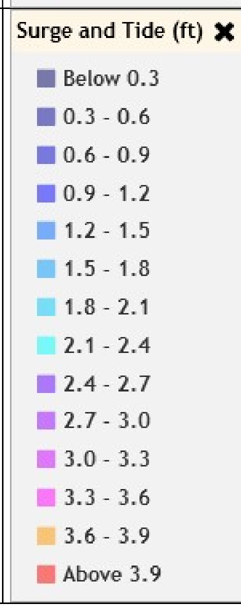
APPENDIX B. LAYER DEFINITIONS

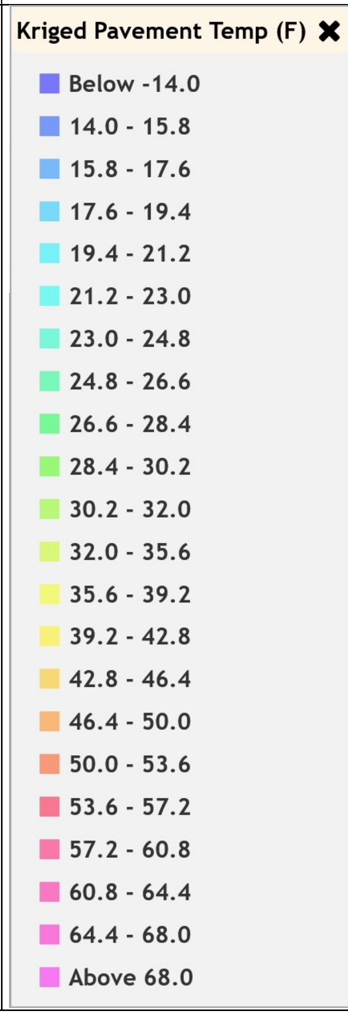
Table 5. Layer Definitions.

Layer	Observation Type	Legend
Pavement State	STPVT	 The Pavement State layer categories are consistent with the METRo model pavement state categories. A flooded state is projected from local inundation calculations, where available.
Pavement Temperature	TPVT	 The Pavement Temperature map layer is divided into levels based on pavement behaviors at temperature intervals. Pavement temperatures between 30°F and 34°F indicate a transition to freezing conditions. Salt treatment may be effective on pavement at temperatures between 20°F and 29°F. Salt loses its effectiveness as an anti-icing agent below 20°F.
Pavement Snow Depth	DPHSN	 The Pavement Snow Depth layer is represented in bands for noticeable, actionable, and significant impacts on travel conditions.
Pavement Flood Depth	DPHLNK	 The flood depth calculation is available only at specific locations, and an absence of a flood depth indication is not necessarily evidence of an absence of flooding.

Layer	Observation Type	Legend
Traffic Speed	SPDLNK	 The Traffic Speed layer is divided into five equal bands ranging from 0 to 75 mph.
Air Temperature	TAIR	 The Air Temperature layer is divided into layers based on typical temperature behaviors. The narrow band at 32 F highlights the precipitation freezing point.
Surface Visibility	VIS	 The Surface Visibility layer remains white until the visibility is below 0.6 mi (1 km). Travelers can be significantly affected by visibility below this point.
Wind Speed	SPDWND	 The Wind Speed bands mark increasing intensity up to and through the tropical storm and hurricane levels.

Layer	Observation Type	Legend
Wind Gust Speed	GSTWND	 The Wind Gust Speed bands use the same levels as Wind Speed.
Radar	RDR0	 The Radar layer is divided into levels based on those used by the NWS.

Layer	Observation Type	Legend
Precipitation Rate & Type	PCCAT	 The Precipitation Rate & Type layer is divided into categories based on the observation returned by PCCAT. PCCAT categories are described in Appendix C.
Surge and Tide	DPHLIQ	 The surge and tide layer presents the depth of water from storm surge and tide over the coastal landform, provided by the NWS from its Advanced Circulation model for oceanic, coastal, and estuarine waters (ADCIRC).

Layer	Observation Type	Legend	
Kriged Pavement Temp	TPVT		Spatial estimates of pavement temperature are computed by IMRCP from measurements at ESS using Kriging statistical methods.

Layer	Observation Type	Legend
Kriged Subsurface Temp	TSSRF	Kriged Subsurface Temp (F) Below -14.0 14.0 - 15.8 15.8 - 17.6 17.6 - 19.4 19.4 - 21.2 21.2 - 23.0 23.0 - 24.8 24.8 - 26.6 26.6 - 28.4 28.4 - 30.2 30.2 - 32.0 32.0 - 35.6 35.6 - 39.2 39.2 - 42.8 42.8 - 46.4 46.4 - 50.0 50.0 - 53.6 53.6 - 57.2 57.2 - 60.8 60.8 - 64.4 64.4 - 68.0 Above 68.0 Spatial estimates of subsurface temperature are computed by IMRCP from measurements at ESS using Kriging statistical methods.
NWS Alerts	EVT	NWS Alerts Fire Heat Storm/Tornado Wind/Fog/Smoke Air Quality Earthquake/Volcano Winter Storm Freeze Cold Flood Lake/Marine/Coastal Tropical Storm Special Weather Other Other The NWS Alerts Layer is categorized based on the type of alert issued by the NWS.

Layer	Observation Type	Legend	
Tropical Storm Cone	TRSCNE		Tropical storm cones are represented from National Hurricane Center (NHC) data to show the potential path and current category of tropical storms.
Inundation	STG		The NWPS inundation polygons show locations that would be flooded at the current or predicted flood stage.

Source: FHWA, 2019

APPENDIX C. OBSERVATION TYPE DEFINITIONS

Table 6. Observation type descriptions.

Name	Description
COVCLD	total cloud cover
DIRWND	wind direction
DPHLIQ	liquid inundation depth
DPHLNK	link depth
DPHSN	snow inundation depth
EVT	event
GSTWND	wind speed gust
MPLOW	MAC main plow
PCCAT	precipitation category
PRSUR	surface pressure
RH	relative humidity
RTEPC	precipitation rate
RTLIQM	liquid material rate
RTPREM	prewet material rate
RTSLDM	solid material rate
SPDLNK	average speed of vehicles on each link
SPDWND	wind speed
SSCST	extra tropical storm surge combined surge and tide
STG	flood stage
STPVT	pavement state
TAIR	air temperature
TDEW	dew point
TPLIQM	liquid material type
TPLOW	MAC tow plow
TPPREM	prewet material type
TPSLDM	solid material type
TPVT	pavement temperature
TRFLNK	traffic
TRSCAT	tropical storm category
TRSCNE	tropical storm cone
TRSTRK	tropical storm track
TSSRF	subsurface temperature
TYPPC	precipitation type
VIS	surface visibility
WPLOW	MAC wing plow

Table 7. Observation types enumeration.

Name	Enumeration	Description
EVT	101	light-winter-precip
	102	moderate-winter-precip
	103	heavy-winter-precip
	104	light-precip
	105	moderate-precip
	106	heavy-precip
	107	low-visibility
	108	flood-stage-action
	109	flood-stage-flood
	201	dew-on-roadway
	202	frost-on-roadway
	203	blowing-snow
	204	icy-roadway
	301	incident
	302	workzone
	303	slow-traffic
	304	very-slow-traffic
	305	flooded-road
	306	lengthy-queue
	307	unusual-congestion
	399	test
	512	accident
	513	serious-accident
	514	injury-accident
	515	minor-accident
	516	multi-vehicle-accident
	517	numerous-accidents
	518	accident-involving-a-bicycle
	519	accident-involving-a-bus
	520	accident-involving-a-motorcycle
	521	accident-involving-a-pedestrian
	522	accident-involving-a-train
	523	accident-involving-a-truck
	524	accident-involving-a-semi-trailer
	525	accident-involving-a-hazardous-materials
	526	earlier-accident
	527	medical-emergency
	528	secondary-accident
	529	rescue-and-recovery-work-removed
	530	accident-investigation-work
	531	incident
	532	stalled-vehicle
	533	abandoned-vehicle
	534	disabled-vehicle

Name	Enumeration	Description
	535	disabled-truck
	536	disabled-semi-trailer
	537	disabled-bus
	538	disabled-train
	539	vehicle-spun-out
	540	vehicle-on-fire
	541	vehicle-in-water
	542	vehicles-slowing-to-look-at-accident
	543	jackknifed-semi-trailer
	544	jackknifed-trailer-home
	545	jackknifed-trailer
	546	spillage-occurring-from-moving-vehicle
	547	acid-spill
	548	chemical-spill
	549	fuel-spill
	550	hazardous-materials-spill
	551	oil-spill
	552	spilled-load
	553	toxic-spill
	554	overturned-vehicle
	555	overturned-truck
	556	overturned-semi-trailer
	557	overturned-bus
	558	derailed-train
	559	stuck-vehicle
	560	truck-stuck-under-bridge
	561	bus-stuck-under-bridge
	562	accident-cleared
	563	incident-cleared
	1000	Extreme Fire Danger
	1001	Fire Warning
	1002	Fire Weather Watch
	1003	Red Flag Warning
	1004	Heat Advisory
	1005	Excessive Heat Warning
	1006	Excessive Heat Watch
	1007	Severe Thunderstorm Warning
	1008	Severe Thunderstorm Watch
	1009	Storm Warning
	1010	Storm Watch
	1011	Tornado Warning
	1012	Tornado Watch
	1013	Severe Weather Statement
	1014	High Wind Warning
	1015	High Wind Watch

Name	Enumeration	Description
	1016	Wind Advisory
	1017	Extreme Wind Warning
	1018	Brisk Wind Advisory
	1019	Blowing Dust Advisory
	1020	Dust Storm Warning
	1021	Dense Fog Advisory
	1022	Dense Smoke Advisory
	1023	Air Quality Alert
	1024	Air Stagnation Advisory
	1025	Ashfall Advisory
	1026	Ashfall Warning
	1027	Earthquake Warning
	1028	Volcano Warning
	1029	Winter Storm Warning
	1030	Winter Storm Watch
	1031	Winter Weather Advisory
	1032	Ice Storm Warning
	1033	Blizzard Warning
	1034	Blizzard Watch
	1035	Avalanche Warning
	1036	Avalanche Watch
	1037	Blowing Snow Advisory
	1038	Snow and Blowing Snow Advisory
	1039	Heavy Snow Warning
	1040	Sleet Advisory
	1041	Sleet Warning
	1042	Snow Advisory
	1043	Freeze Warning
	1044	Freeze Watch
	1045	Freezing Drizzle Advisory
	1046	Freezing Fog Advisory
	1047	Freezing Rain Advisory
	1048	Freezing Spray Advisory
	1049	Frost Advisory
	1050	Hard Freeze Warning
	1051	Hard Freeze Watch
	1052	Wind Chill Advisory
	1053	Wind Chill Warning
	1054	Wind Chill Watch
	1055	Extreme Cold Warning
	1056	Extreme Cold Watch
	1057	Flash Flood Statement
	1058	Flash Flood Warning
	1059	Flash Flood Watch
	1060	Flood Advisory

Name	Enumeration	Description
	1061	Flood Statement
	1062	Flood Warning
	1063	Flood Watch
	1064	Hydrologic Advisory
	1065	Hydrologic Outlook
	1066	Beach Hazards Statement
	1067	Coastal Flood Advisory
	1068	Coastal Flood Statement
	1069	Coastal Flood Warning
	1070	Coastal Flood Watch
	1071	Gale Warning
	1072	Gale Watch
	1073	Hazardous Seas Warning
	1074	Hazardous Seas Watch
	1075	Heavy Freezing Spray Warning
	1076	Heavy Freezing Spray Watch
	1077	High Surf Advisory
	1078	High Surf Warning
	1079	Lake Effect Snow Advisory
	1080	Lake Effect Snow and Blowing Snow Advisory
	1081	Lake Effect Snow Warning
	1082	Lake Effect Snow Watch
	1083	Lakeshore Flood Advisory
	1084	Lakeshore Flood Statement
	1085	Lakeshore Flood Warning
	1086	Lakeshore Flood Watch
	1087	Lake Wind Advisory
	1088	Low Water Advisory
	1089	Marine Weather Statement
	1090	Rip Current Statement
	1091	Small Craft Advisory
	1092	Special Marine Warning
	1093	Tsunami Advisory
	1094	Tsunami Warning
	1095	Tsunami Watch
	1096	Hurricane Force Wind Warning
	1097	Hurricane Force Wind Watch
	1098	Hurricane Statement
	1099	Hurricane Warning
	1100	Hurricane Watch
	1101	Hurricane Wind Warning
	1102	Hurricane Wind Watch
	1103	Tropical Storm Warning
	1104	Tropical Storm Watch
	1105	Tropical Storm Wind Warning

Name	Enumeration	Description
	1106	Tropical Storm Wind Watch
	1107	Typhoon Statement
	1108	Typhoon Warning
	1109	Typhoon Watch
	1110	Hazardous Weather Outlook
	1111	Special Weather Statement
	1112	911 Telephone Outage
	1113	Administrative Message
	1114	Child Abduction Emergency
	1115	Civil Danger Warning
	1116	Civil Emergency Message
	1117	Evacuation Immediate
	1118	Hazardous Materials Warning
	1119	Law Enforcement Warning
	1120	Local Area Emergency
	1121	Nuclear Power Plant Warning
	1122	Radiological Hazard Warning
	1123	Shelter In Place Warning
	1124	Test
	5888	impassable
	5889	almost-impassable
	5890	passable-with-care
	5891	passable
	5892	surface-water-hazard
	5893	danger-of-hydroplaning
	5894	wet-pavement
	5895	treated-pavement
	5896	slippery
	5897	low-ground-clearance
	5898	at-grade-level-crossing
	5899	mud-on-roadway
	5900	leaves-on-roadway
	5901	loose-sand-on-roadway
	5902	loose-gravel
	5903	fuel-on-roadway
	5904	oil-on-roadway
	5905	road-surface-in-poor-condition
	5906	melting-tar
	5907	uneven-lanes
	5908	rough-road
	5909	rough-crossing
	5910	ice
	5911	icy-patches
	5912	black-ice
	5913	ice-pellets-on-roadway

Name	Enumeration	Description
	5914	ice-build-up
	5915	freezing-rain
	5916	wet-and-icy-roads
	5917	melting-snow
	5918	slush
	5919	frozen-slush
	5920	snow-on-roadway
	5921	packed-snow
	5922	packed-snow-patches
	5923	plowed-snow
	5924	wet-snow
	5925	fresh-snow
	5926	powder-snow
	5927	granular-snow
	5928	froazen-snow
	5929	crusted-snow
	5930	deep-snow
	5931	snow-drifts
	5932	drifting-snow
	5933	expected-snow-accumulation
	5934	current-snow-accumulation
	5935	sand
	5936	gravel
	5937	paved
	5938	dry-pavement
	5939	snow-cleared
	5940	pavement-conditions-improved
	5941	skid-hazard-reduced
	5942	pavement-conditions-cleared
MPLOW	0	Plow up
	1	Plow down

Name	Enumeration	Description
PCCAT	0	no-precipitation
	1	light-rain
	2	moderate-rain
	3	heavy-rain
	4	light-freezing-rain
	5	moderate-freezing-rain
	6	heavy-freezing-rain
	7	light-snow
	8	moderate-snow
	9	heavy-snow
	10	light-ice
	11	moderate-ice
	12	heavy-ice
	101	other
	102	unknown
	104	light-unidentified
	105	moderate-unidentified
	106	heavy-unidentified
STG	0	not-defined
	1	no-action
	2	action
	3	flood
	4	moderate
	5	major
STPVT	1	other
	2	error
	3	dry
	4	trace-moisture
	5	wet
	6	chemically-wet
	7	ice-warning
	8	ice-watch
	9	snow-warning
	10	snow-watch
	11	absorption
	12	dew
	13	frost
	14	absorption-at-dewpoint
	20	ice/snow
	21	slush
	22	melting-snow
	23	icing-rain
	30	flooded
TPLOW	0	Plow up
	1	Plow down

Name	Enumeration	Description
TRSCAT	479 642 809 1057 1072 37345 37360	Tropical Depression Hurricane Major Hurricane Tropical Depression Tropical Storm Subtropical Depression Subtropical Storm
TRSCNE	479 642 809 1057 1072 37345 37360	Tropical Depression Hurricane Major Hurricane Tropical Depression Tropical Storm Subtropical Depression Subtropical Storm
TRSTRK	479 642 809 1057 1072 37345 37360	Tropical Depression Hurricane Major Hurricane Tropical Depression Tropical Storm Subtropical Depression Subtropical Storm
TYPPC	0 1 2 3 4 5 6	none rain snow ice-pellets freezing-rain other unknown
WPLOW	0 1	Plow up Plow down

Table 8. Observation type source descriptions.

Source	Forecast/ Observation	Spatial Extent	Temporal Extent	Observation Types
ADCIRC	forecasts	2.5 km x 2.5 km grid for CONUS	1 hour forecasts for 120 hours starting 6 hours after collection	SSCST
CAP	observations and forecasts	County and custom polygons	Varies	EVT
GFS	forecasts	25 km x 25 km grid for entire world	3 hour forecasts for 168 hours starting 54 hours after collection	DPHSN, GSTWND, PCCAT, PRSUR, RH, RTEPC, SPDWND, TAIR, TDEW, TYPPC, VIS
IMRCP	observations	Area surrounding individual stations	Observations valid for 1 hour	TPVT, SSRF
IMRCP	forecasts	2.5 km x 2.5 km grid for CONUS	1 hour forecasts for 72 hours starting 1 hour after collection	PCCAT
IMRCP	observations	1 km x 1 km grid for CONUS	Observations valid for 4 minutes	PCCAT
METRo	forecasts	Individual segments	2 minute forecasts for 1 hour, then 20 minute forecasts for 11 hours	DPHLIQ, DPHSN, STPVT, TPVT, TSSRF
MLP	forecasts	Individual segments	15 minute forecasts for 2 hours, or 1 hour forecasts for 24 hours	SPDLNK
MRMS	observations	1 km x 1 km grid for CONUS	Observations valid for 4 minutes	RDR0, RTEPC
NDFD	forecasts	2.5 km x 2.5 km grid for CONUS	1to 3 hour forecasts for 72 hours starting 1 hour after collection	COVCLD, RTEPC, SPDWND, TAIR, TDEW
NHC	forecasts	Tropical storm cones of probability	6 hour forecasts for 120 hours	TRSCAT, TRSCNE, TRSTRK
NWM	forecasts	Inundation polygons	120 hour forecasts every hour	STG
NWPS	observations and forecasts	Individual stations	Most recent observed values	EVT, STG, STPVT

Source	Forecast/ Observation	Spatial Extent	Temporal Extent	Observation Types
			and 24 hour forecast	
RTMA	forecasts	2.5 km x 2.5 km grid for CONUS	1 hour forecast	COVCLD, DIRWND, GSTWND, PRSUR, SPDWND, TAIR, TDEW, VIS
WxDE	observations	Individual stations	Observations valid for 1 hour	DIRWND, DPHLNK, GSTWND, PCCAT, PRSUR, RH, RTEPC, SPDWND, STPVT, TAIR, TDEW, TPVT, TSSRF, TYPPC, VIS

Table 9. Observation type synthesis algorithms.

Name	Description	Source – Observations	Source – Predictions
dphliq	liquid inundation depth	Model of the Environment and Temperature of Roads (METRo) is run for each link in the road network model to determine liquid inundation depth estimations.	METRo is run for each link in the road network model to determine liquid inundation depth predictions.
dphlnk	link depth	NWPS stage observations at select locations in the road network model are collected when new values are available. These values are used to determine the flood depth on links based on inundation mapping provided by National Oceanic and Atmospheric Administration (NOAA)/NWS.	NWPS stage predictions at three locations in the road network model are collected when new values are available. These values are used to determine the flood depth on links based on inundation mapping provided by NOAA.
dphsn	snow inundation depth	METRo is run for each link in the road network model to determine pavement snow depth estimations. The snow inventory is tracked from previous runs.	METRo is run for each link in the road network model to determine pavement snow depth predictions. The snow inventory is tracked from each run to the next, accounting for new accumulation and melting.
evt	event	Workzone and Incident event details are collected from contributing transportation management centers. NWS Common Alerting Protocol (CAP) alert events are collected from NWS. CAP alerts affecting counties use previously stored county defintions to display on the map. CAP alerts affecting areas other than counties use the area definition provided in the CAP alert to display on the map.	Workzone and Incident event details are collected from contributing transportation management centers. NWS CAP alert events are collected from NWS. CAP alerts affecting counties use previously stored county defintions to display on the map. CAP alerts affecting areas other than counties use the area definition provided in the CAP alert to display on the map.

Name	Description	Source – Observations	Source – Predictions
pccat	precipitation category	The precipitation category is determined based on observation TYPPC and RTEPC. • Light Freezing Rain: RTEPC <= 7.056x10-5 kg/m2-s and TYPPC = [freezing rain] • Medium Freezing Rain: 7.056x10-5 < RTEPC <= 7.056x10-4 kg/m2-s and TYPPC = [freezing rain] • Heavy Freezing Rain: 7.056x10-4 < RTEPC kg/m2-s and TYPPC = [freezing rain] • Light Snow: RTEPC <= 7.056x10-5 kg/m2-s and TYPPC = [snow] • Medium Snow: 7.056x10-5 < RTEPC <= 7.056x10-4 kg/m2-s and TYPPC = [snow] • Heavy Snow: 7.056x10-4 < RTEPC kg/m2-s and TYPPC = [snow] • Light Ice Pellets: RTEPC <= 7.056x10-5 kg/m2-s and TYPPC = [ice pellets,] • Medium Ice Pellets: 7.056x10-5 < RTEPC <= 7.056x10-4 kg/m2-s and TYPPC = [ice pellets,] • Heavy Ice Pellets: 7.056x10-4 < RTEPC kg/m2-s and TYPPC = [ice pellets] • Light Rain: RTEPC <= 7.056x10-4 kg/m2-s and TYPPC = [rain]	The precipitation category is determined based on predicted TYPPC and RTEPC. • Light Freezing Rain: RTEPC <= 7.056x10-5 kg/m2-s and TYPPC = [freezing rain] • Medium Freezing Rain: 7.056x10-5 < RTEPC <= 7.056x10-4 kg/m2-s and TYPPC = [freezing rain] • Heavy Freezing Rain: 7.056x10-4 < RTEPC kg/m2-s and TYPPC = [freezing rain] • Light Snow: RTEPC <= 7.056x10-5 kg/m2-s and TYPPC = [snow] • Medium Snow: 7.056x10-5 < RTEPC <= 7.056x10-4 kg/m2-s and TYPPC = [snow] • Heavy Snow: 7.056x10-4 < RTEPC kg/m2-s and TYPPC = [snow] • Light Ice Pellets: RTEPC <= 7.056x10-5 kg/m2-s and TYPPC = [ice pellets,] • Medium Ice Pellets: 7.056x10-5 < RTEPC <= 7.056x10-4 kg/m2-s and TYPPC = [ice pellets,] • Heavy Ice Pellets: 7.056x10-4 < RTEPC kg/m2-s and TYPPC = [ice pellets] • Light Rain: RTEPC <= 7.056x10-4 kg/m2-s and TYPPC = [rain]

Name	Description	Source – Observations	Source – Predictions
		• Medium Rain: 7.056×10^{-4} < RTEPC <= 2.117×10^{-3} kg/m^2-s and TYPPC = [rain] • Heavy Rain: 2.117×10^{-3} < RTEPC kg/m^2 and TYPPC = [rain]	• Medium Rain: 7.056×10^{-4} < RTEPC <= 2.117×10^{-3} kg/m^2-s and TYPPC = [rain] • Heavy Rain: 2.117×10^{-3} < RTEPC kg/m^2 and TYPPC = [rain]
stpvt	pavement state	METRo is run for each link in the road network model to determine pavement state estimations. • Dry Road: The water reservoir contains less than 0.01 mm and the ice/snow reservoir contains less than .2 mm of water equivalent. • Wet road: The water reservoir contains more than 0.01 mm of water. • Ice/Snow: The ice/snow reservoir contains more than 0.2 mm of water equivalent. • Water/Snow: Both of the reservoirs (water and ice/snow) contain more than 0.2 mm of water equivalent. • Dew: Condensation on the road when the temperature of the surface of the road is above the freezing point. • Frost: Condensation on the road when the temperature of the surface of the road is below the freezing point or water already present on the road is turning into ice.	METRo is run for each link in the road network model to determine pavement state predictions. • Dry Road: Each reservoir (water and ice/snow) contains less than 0.01 mm of liquid water equivalent. • Wet road: The water reservoir contains more than 0.01 mm of water. • Ice/Snow: The ice/snow reservoir contains more than 0.2 mm of water equivalent. • Water/Snow: Both of the reservoirs (water and ice/snow) contain more than 0.2 mm of water equivalent. • Dew: Condensation on the road when the temperature of the surface of the road is above the freezing point. • Frost: Condensation on the road when the temperature of the surface of the road is below the freezing point or water already present on the road is turning into ice.
trflnk	traffic	The estimated speed value for each link is divided by the speed limit for that link.	The predicted speed value for each link is divided by the speed limit for that link.

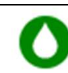
APPENDIX D. ALERT DEFINITIONS

Table 10.Traffic Alert Definitions

Type	Algorithm	Extent	Reference	Notify	Icon
Incident	EVT=Incident	point	TMC	n/a	
Work Zone	EVT=Work zone	link	TMC	n/a	

Source: FHWA, 2019

Table 11. Weather Alert Definitions.

Type	Algorithm	geoExtent	Reference	Notify	Icon
Medium Winter Precip	PCCAT= [Medium Freezing Rain, Medium Snow, Medium Ice Pellets]	area	RAP	Y	
Heavy Winter Precip	PCCAT= [Heavy Freezing Rain, Heavy Snow, Heavy Ice Pellets]	area	RAP	Y	
Medium Precip	PCCAT= [Medium Rain]	area	RAP	N	
Heavy Precip	PCCAT= [Heavy Rain]	area	RAP	Y	
Flood Stage Action	n/a	point	NWPS	n/a	
Flood Stage Flood	n/a	point	NWPS	n/a	
Low Visibility	VIS < 0.2 mi	area	RAP	Y	

Source: FHWA, 2019

Table 12. Road Condition Alert Definitions.

Type	Algorithm	geoExtent	Reference	Notify	Icon
Low Visibility	VIS < 0.2 mi	area	RAP	Y	
Ice on Bridge	STPVT= [ice]	segment	METRo	Y	
Flooded Road	DPHLNK > 0 in.	segment	NWPS	Y	

Source: FHWA, 2019

Table 13. Tropical Storm Categories.

Type	Algorithm	Extent	Reference	Notify	Icon
Subtropical Depression	From NHC	point	NHC	n/a	SD
Subtropical Storm	From NHC	point	NHC	n/a	SS
Tropical Depression	From NHC	point	NHC	n/a	TD
Tropical Storm	From NHC	point	NHC	n/a	TS
Hurricane	From NHC	point	NHC	n/a	HU
Major Hurricane	From NHC	point	NHC	n/a	MH

APPENDIX E. TRAFFIC DATA INTERFACE SPECIFICATION

Objective

The objective of this interface specification is to provide a standard format for traffic data to be provided to IMRCP from other traffic data records and real-time data services. IMRCP will expect traffic data to conform to this specification for collection and further processing. A deploying agency would create interface data files in this format from its own data records, whether from archived data or as a front-end adapter service for its IMRCP system deployment. Currently, IMRCP does not have the ability to change these formats or collect data using other file formats. During this phase, ATI created an adapter that will be made available on the IMRCP GitHub repository that a transportation agency could modify and use for files that do not meet the IMRCP specification.

Interface Specification

The speed data adapter file format is used to store and transmit speed, volume, and occupancy observations for road network locations and/or segments. The name for each file also has a specific required elements separated by an underscore:
<source>_<observation_type>_<start_time>_<end_time>_<received_time>.txt.
Speed-related data files are commonly generated every 5 or 10 minutes. File content uses UTF-8 character encoding, and can optionally be compressed with gzip, which is recommended. Since the file names are time-based, it is a good idea to impose a year/month/day directory structure in order to prevent thousands of files accumulating in a single directory.
Sample speed file name: here_speed_202301010000_202301010005_202301010005.txt.gz

Table 14. Traffic Speed File Name Format.

File Name Part	Required	Detail	Example
source	required	Typically, the vendor of the data. In this case, HERE.	here
observation_type	required	The expected observation type of the contained data. In this case, speed or speed/volume/occupancy (svo).	speed or svo
start_time	required	The UTC time of the earliest observation starting time within the file in the format yyyyMMddHHmm.	202301010000
end_time	required	The UTC time of the latest observation ending time within the file in the format yyyyMMddHHmm.	202301010005
received_time	required	The UTC time when the file was received from the source in the format yyyyMMddHHmm. Typically, when the file was downloaded or created.	202301010005

The speed data adapter file begins with two required header rows. The first header row is a single double-quoted string, i.e., “version 1.0”, indicating the version of the data format used. The version specifier consists of the lower-case word “version” followed by a single space, the major version number, a period character, and finally the minor version number.
The second header row is the set of column name strings identifying the data intent for each column. For version 1.0 of the speed data, the header row contains id, description, start_time, end_time, speed, volume, occupancy, and location. Whitespace between columns is the space character, and is ignored. The column separator is the pipe “|” character. String type data requires enclosing double-quotes, should not contain control characters, and should backslash escape (\”) double-quote and (\\)backslash. Empty columns contain no characters, which appears as two consecutive pipe characters “||” or only whitespace.
This particular data format allows columns to be variably sized depending on the largest width of the column header or column data, if desired, while remaining relatively simple to implement file writing and reading software. Speed, volume, and occupancy data columns are marked as optional so the file can contain any combination of the available parameters.

Table 15. Traffic Speed File Data Columns.

Column	Required	Type	Units	Detail	Example
id	optional	string	none	Free-form external identifier string. A copy of the source identifier can go here if the source has one.	“AX2348”
description	optional	string	none	Free-form string describing a road segment, such as the names of intersecting roads.	“I-80 W @ Cheyenne”
start_time	required	time stamp	UTC	ISO 8601 formatted date and time when the observations were measured. Inclusive.	2023-01-31T06:30:00
end_time	required	time stamp	UTC	ISO 8601 formatted date and time indicating the extent of the observation measurements. Exclusive.	2023-01-31T06:35:00
speed	optional	decimal number	kph	The aggregate mean traffic speed across all lanes for the segment location.	95.6
volume	optional	integer number	vehicle count	The sum of vehicles crossing all lanes for the segment location within the start_time and end_time period.	1215
occupancy	optional	decimal number	percent	The percentage of total available space occupied by	47.1

Column	Require d	Type	Units	Detail	Example
				vehicles across all lanes of the segment location.	
location	required	decimal number set	decimal degrees	A comma-separated list of bracketed decimal degree longitude and latitude coordinates. Can be a single-point. A set of points is in traffic-flow downstream order.	[-86.781667,33.178333] or [-86.366667,32.383333],[-86.366667,32.383333]

Sample speed file content is at the end of this document.

The event data adapter file format is used to store and transmit event observations such as work zones, crashes, and variable speed limits for road network locations and/or segments. It has the same file name format specification as the speed data file name format, and files should be segregated into a year/month/day directory structure.
File content is also encoded using UTF-8, and recommended to be compressed with gzip. Event-related data files are commonly generated every 5 or 10 minutes, and may have end times in the distant future when they contain work zones planned in advance.
Sample event file name: here_event_202301021115_202301311159_202301021116.txt.gz

Table 16. Event File Name Format.

File Name Part	Required	Detail	Example
source	required	Typically, the vendor of the data. It could be the Work Zone Data Exchange (WZDx) or an agency.	wzdx or ALDOT
observation_type	required	The expected observation type of the contained data. In this case, event.	event
start_time	required	The UTC time of the earliest event starting time within the file in the format yyyyMMddHHmm.	202301021115
end_time	required	The UTC time of the latest event ending time within the file in the format yyyyMMddHHmmss. This can be in the far future for planned work zones.	202301311159
received_time	required	The UTC time when the file was received from the source in the format yyyyMMddHHmm. Typically, when the file was downloaded or created.	202301021116

The event data adapter file begins with two required header rows. The first header row is a single double-quoted string, i.e., “version 1.0”, indicating the version of the data format used. The

version specifier consists of the lower-case word “version” followed by a single space, the major version number, a period character, and finally the minor version number.
The second header row is the set of column name strings identifying the data intent for each column. For version 1.0 of the event data the header row contains id, event_type, description, start_time, end_time, update_time, lanes_affected, speed_limit, and location. Whitespace between columns is the space character, and is ignored. The column separator is the pipe “|” character. String type data requires enclosing double-quotes, should not contain control characters, and should backslash escape (\”) double-quote and (\\)backslash. Empty columns contain no characters, which appears as two consecutive pipe characters “||” or only whitespace. This particular data format allows columns to be variably sized depending on the largest width of the column header or column data, if desired, while remaining relatively simple to implement file writing and reading software.

Table 17. Event File Data Columns.

Column	Required	Type	Units	Detail	Example
id	required	string	none	Events can be modified throughout its duration, so a unique identifier is required to group information about the same event within a single file.	“727ca3c5”
event_type	required	string	none	A short string indicating the type of event recorded.	workzone, incident, speed-change
description	optional	string	none	Optional free-form string providing additional detail for an event.	“Planned construction on Main Street. Detour to 4th Street.”
start_time	required	time stamp	UTC	ISO 8601 formatted date and time when the event began. Inclusive.	2023-02-02T16:00:00
end_time	optional	time stamp	UTC	ISO 8601 formatted date and time indicating when the event concluded. Exclusive. This column can be empty when an event is in progress.	2023-01-31T17:59:59
update_time	required	time stamp	UTC	ISO 8601 formatted date and time indicating when event information has changed. Initially indicates when an event was recorded.	2023-02-02T15:30:00
lanes_affected	optional	integer number	lane count	The number of lanes closed for the segment location during an event. Negative number indicates lanes	1215

Column	Required	Type	Units	Detail	Example
				added for cases like reversible lanes.	
speed_limit	optional	decimal number	kph	The regulatory speed limit applied to the segment location. Used for work zones and variable speed limit signs.	60
location	required	decimal number set	decimal degrees	A comma-separated list of bracketed decimal degree longitude and latitude coordinates. Can be a single-point. A set of points is in traffic-flow downstream order.	[-85.507890,32.602678],[-85.507890,32.607649]

Sample Speed File Content

```
"version 1.0"
              "id"                 |      "description"      |  "start_time (UTC)" |   "end_time (UTC)"  | "speed (kph)" | "volume (veh)" | "occupancy (%)" |          "location ([lon,lat],...)"
"some_kind_of_optional_id"         | "I-80 W @ Cheyenne"     | 2023-01-31T06:30:00 | 2023-01-31T06:34:59 |      100      |      1215      |       47        | [-86.781667,33.178333]
"13b6f73c-556f-48f6-bce0-c1ba377d22b9" | "a useful description" | 2023-01-31T07:10:00 | 2023-01-31T07:14:59 |       70      |      2211      |       69        | [-86.366667,32.383333],[-86.366667,32.383333]
```

Sample Event File Content

```
"version 1.0"
   "id"     |  "event_type"  |      "description"      |  "start_time (UTC)" |   "end_time (UTC)"  | "update_time (UTC)" | "lanes_affected" | "speed_limit (kph)" |           "location ([lon,lat],...)"
"ed65ae06" | "workzone"     | "US-82 @ AL-271"        | 2023-01-01T00:00:00 | 2023-02-28T23:59:59 | 2022-12-17T08:41:22 |        2         |         30          | [-86.193924,32.296420]
"a1661a5c" | "incident"     | "a useful description"  | 2023-02-01T04:36:19 |                     | 2023-02-01T04:36:19 |        1         |                     | [-86.803708,33.525476]
"727ca3c5" | "speed-change" |                         | 2023-02-02T16:00:00 | 2023-01-31T17:59:59 | 2023-02-02T15:30:00 |                  |         60          | [-85.507890,32.602678],[-85.507890,32.607649]
```

APPENDIX F. TILED DATA FILE FORMAT

Background

IMRCP collects heterogeneous data from a variety of sources. To limit the number of store components and to ensure data can be served quickly and efficiently, IMRCP uses a homogenous data format to store all observations and forecasts. These Tiled Data Files are intended to be flexible and handle a wide variety of data. They are indexed in multiple ways: the file directory structure and file names index data by observation type and time, the format of the files index data spatially. TileFileWriter components in the Collect and Computational packages are responsible for converting external data into the Tiled Data File Format. The TileFileReader component in the Store package is responsible for parsing the Tiled Data Files and creating observation and forecast objects for the system to use.

File Format Specification

IMRCP Tiled Data Files are binary files with two sections: the header and tiled data. The header contains metadata and describes what kind of data to expect in the tiled data section. The following table details the header.

Table 18. Event File Data Columns.

Name	Bits	Type	Description
version	8	ubyte	file format version number
lon min	32	int	minimum longitude scaled to 7 decimal places of data
lat min	32	int	minimum latitude scaled to 7 decimal places of data
lon max	32	int	maximum longitude scaled to 7 decimal places of data
lat max	32	int	maximum latitude scaled to 7 decimal places of data
obs type	32	int	Observation type id that gets converted into a base-36 string
obs flags	4	nybble	Tells if there are obs flags present in each obs record. 0 not present, 1 present
value type	4	nybble	Tells how the value for each obs record is stored. 1 ubyte, 2 sbyte, 3 quarter point floating, 4 ushort, 5 sshort, 6 half point floating, 7 uint, 8 sint, 9 single float, 10 ulong, 11 slong, 12 double float, 13 ulonglong, 14 slonglong, 15 quad float

geometry flags	8	byte	Tells the geometry type of obs records. 0 variable (get from obs flag), 1 point, 2 polyline, 3 polygon
obs id size	8	byte	Size of observation id in the obs record.-1 variable (get from obs id length in obs record), 0 no id, 16 uuid, 32 32 bytes generated from sponge algorithm
obj type	4	nybble	System object type to associate the observation with. 1 sensor, 2 node, 3 link, 4 segment, 5 route
time format flags	4	nybble	Tells which times are included in each obs record. 0 not included, 1 include. MSB is bit 0. bit 0 not used, bit 1 received time, bit 2 start time, bit 3 end time
received time	64	long	Received time stamp in milliseconds since Epoch
end time offset	32	int	The number of seconds to add to the received time to get the end time of observations described in the obs records. If the value is the maximum integer value that means the observations do not have a specified end time
start time count	8 (16)	byte (short)	The number of start times offset records to read. If the byte is a negative number, another byte is used to get a short, allowing for larger values.
start time offsets	32 * n	int	n is the value from the start time count record. The number of seconds to add to the received time to get the start time of each time range defined. Offsets are relative to the previous start time.
string table length	32	int	number of bytes in XZ compressed string table. 0 for null table
number of strings	32	int	number of string in the string table. Only written if string table length > 0
strings	n	UTF-8	Array of UTF-8 strings compressed by XZ algorithm
tile zoom level	8	ubyte	zoom level (0-23) used to create data tiles
tile size	8	byte	tile element (pixel) width. Determines ordinate bit size, i.e. 16 bits

tile count	32	uint	Number of tile index records
tile index	64	ushort, ushort, uint	First ushort is the horizontal tile index at specified zoom, second ushort is the vertical tile index at specified zoom, and uint is the XZ compressed tile data byte count
tile data			The following entries are the specification of the XZ compressed tile data
obs flag	8	byte	Optional bit flags, only present if header obs flag .record is 1. Most significant nybble is a bit flag for bridge, mobile, event, and reserved. Least significant nybble is the geo type. This overwrites the geometry type contained in the header.
obs type	32	int	Optional observation type, only present if the observation type in the header is "VARIES"
obs id length	8	byte	Optional observation id length, only present when variable length is set in the header.
obs id	n	bytearray	Optional observation id. Only present if the obs id size is not 0.
string flags	8	byte	Optional string bit flag, indicating which string values are present. Only present if the string table is not null.
string values	n	int	optional 4-byte indices to string table, string flags determines how many
geometry	n	byte	Varied based on geometry type. Point: 2 ushorts representing the x and y tile pixel coordinate of the point. Linestring: 1 short representing the number of points. N 2 ushort records where N is the number of points. The first 2 ushort records are the x and y tile pixel coordinate of the first point. Subsequent 2 ushort records are the x and y tile pixel offset from previous point. Polygon: A polygon is an outer ring followed by inner rings that represent holes. Polygons are expected to be closed (first and last coordinate are NOT the same). 1 short representing the number of rings. Rings that are holes are written in reverse point order to increase compression. For each ring, 1 ushort representing the point count. N 2 ushort records where N is the number of points. The first 2 ushort records are the x

			and y tile pixel coordinate of the first point. Subsequent 2 ushort records are the x and y tile pixel offset from the previous point.
obs values	n	varies	Array of values, the size is determined by the value type in the header. The number of values is number of start times in the header
recv time	32	int	Time the observation was received. Optional 4 byte offset from file received time in seconds
start time	32	int	Time the observation starts being valid. Optional 4 byte offset from file received time in seconds
end time	32	int	Time the observation stops being valid. Optional 4 byte offset from file received time in seconds

U.S. Department of Transportation
Federal Highway Administration
Office of Operations
1200 New Jersey Avenue, SE
Washington, DC 20590

Office of Operations website
https://ops.fhwa.dot.gov