KAJIAN PELAN INDUK SISTEM PENGANGKUTAN PINTAR

Development of ITS System Architecture for Malaysia

Technical Note No. 2

Volume II – Process Specifications

October 2006

TABLE OF CONTENTS

Logical Architecture Framework Volume II - Process Specifications

1 PROCESS SPECIFICATIONS ……1
1.1 Introduction……1
1.2 Process Specifications Summary……..1-529

1 Process Specifications

1.1 Introduction

In Volume I, the Logical Architecture is first illustrated as a single function to show the external inputs (sources) and outputs (sinks) by the top level Data Flow Diagram (DFD). This is then decomposed in lower level of DFDs to show the next level process bubbles in the ITS. These bubbles are then further decomposed into lower levels. At the lowest level of this decomposition are the actual Process Specifications (P-Specs). The Process Specifications of the Malaysian ITS Architecture are summarized in Section 1.2.

For each P-Spec a structured English specification is composed which defines how the P-Spec output data flows are constructed from its input data flows. It is the P-Specs that represent the sources (and sinks) of data flows inside the ITS. Technically, a “data store” can also source and sink flows, however all stores in the Architecture are tightly coupled to a “store controlling” process, which can be considered logically to be the source and sink for data to/from the store.

1.2 Process Specifications Summary

1.1.1.1 Process Traffic Sensor Data

Description:

Overview: This process shall be responsible for collecting surveillance obtained from the roadside, vehicles, pedestrians (travelers using other modes of transport), railroad grade and multimodal crossings. Where any of the data is provided in analog form, the process shall be responsible for converting it into digital form and calibrating. The converted data shall be sent to other processes for distribution, further analysis and storage.

In Subsystem:

Roadway Subsystem

In Deployment Package:

AD4

ATMS01

ATMS03

ATMS04

ATMS05

ATMS08

ATMS13

ATMS14

ATMS17

ATMS29

ATMS30

Data Flows:

Input Flows

fmmc_crossing_close_duration
fmmc_crossing_close_time
fp_pedestrian_data
fp_pedestrian_images
fre_physical_conditions
ft_traffic_data
ft_traffic_images
sensor_configuration_data
surveillance_control_data_for_maintenance

Output Flows

fault_data
hov_lane_data_input
hov_sensor_data
incident_analysis_data
local_sensor_data_for_highways
local_sensor_data_for_roads
multimodal_crossing_sensor_data
pedestrian_presence
pedestrian_sensor_data
sensor_data_archive_input
sensor_data_for_reversible_lanes
sensor_fault_data
sensor_status
traffic_sensor_coordination
traffic_sensor_data

traffic_sensor_maintenance_data
traffic_sensor_maintenance_status
traffic_sensor_status
traffic_video_image
traffic_video_image_for_display

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.2;
USR = 1.6.2.1;
USR = 1.6.2.1.1;
USR = 1.6.2.2;
USR = 1.6.2.2.1;
USR = 1.6.2.3;
USR = 1.6.2.3.1;
USR = 1.6.2.4;
USR = 1.7;
USR = 1.7.1;
USR = 1.7.1.1;
USR = 1.7.1.1.1;
USR = 1.7.1.1.1(a);
USR = 1.8;
USR = 1.8.0;
USR = 1.8.3;
USR = 1.8.3.1;
USR = 1.8.3.1(b);

1.1.1.2 Collect and Process Sensor Fault Data

Description:

Overview: This process shall be responsible for collecting sensor status, identifying faults, and logging faults that have been detected by processes in other parts of the Manage Traffic function. It shall be possible for the faults to have been detected locally at the sensors, or centrally through communications links with the sensors. The process shall pass on new fault data to other processes for communication to the Construction and Maintenance terminator and shall receive fault clearances from the same terminator. It shall also maintain a store of the current fault state of all sensors. The process shall provide facilities that enable traffic operations personnel to review and update the current fault status of all sensors. Details of faulty and fixed equipment shall be passed by the process to the traffic control strategy selection process so that it can adjust its strategy to take account of the fault(s).

In Subsystem:

Traffic Management

In Deployment Package:

ATMS01

ATMS03

ATMS04

Data Flows:

Input Flows

environment_sensor_fault_data
environmental_sensor_status
ftop_sensor_fault_data_input
sensor_fault_data
sensor_fault_data_status
traffic_sensor_status

Input/Output

environmental_sensor_data_store

Output Flows

sensor_fault_data_report
ttop_current_sensor_faults

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.8;
USR = 1.8.0;
USR = 1.8.1;
USR = 1.8.1.4;
USR = 1.8.1.4(a);
USR = 1.8.1.5;
USR = 1.8.2;
USR = 1.8.2.13;
USR = 1.8.3;
USR = 1.8.3.1;

1.1.1.3 Process Environmental Sensor Data

Description:

Overview: This process shall be responsible for collecting data obtained from environmental sensors. Where any of the data is provided in analog form, the process shall be responsible for converting it into digital form and calibrating. The converted data shall be sent to other processes for distribution, further analysis and storage.

In Subsystem:

Roadway Subsystem

In Deployment Package:

AD4

ATMS18

Data Flows:

Input Flows

environment_sensor_configuration_data
fre_environmental_conditions

Output Flows

archive_environmental_sensor_data
environment_sensor_data
environment_sensor_fault_data
environmental_coordination
environmental_maintenance_status
environmental_sensor_status

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2;
USR = 1.2.3;
USR = 1.2.3.2;
USR = 1.2.3.2.3;
USR = 1.7;
USR = 1.7.1;
USR = 1.7.1.1;
USR = 1.7.1.1.1;
USR = 1.7.1.1.1(a);
USR = 1.7.1.1.1(b);
USR = 1.7.1.1.1(g);
USR = 1.7.1.2;
USR = 1.7.1.2.1;
USR = 1.7.1.2.1(b);
USR = 1.8.2;
USR = 1.8.2.1;
USR = 1.8.2.1(e);

1.1.1.4 Manage Data Collection and Monitoring

Description:

Overview: This process shall collect and monitor sensor data from the roadside. The process shall collect the sensor data including sensor status and sensor faults from roadside equipment and distribute it to the Manage Archive Data function. The process shall run when a request for data is received from an external source.

In Subsystem:

Roadway Subsystem

In Deployment Package:

AD1

AD2

AD4

Data Flows:

Input Flows

archive_environmental_sensor_data
fault_data
roadside_archive_control
sensor_data_archive_input
sensor_status

Input/Output

roadside_data_archive

Output Flows

roadside_archive_data

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.1;
USR = 7.1.3.1.1(a);
USR = 7.1.3.1.1(c);
USR = 7.1.3.1.3;
USR = 7.1.3.1.3(e);
USR = 7.1.3.1.7;
USR = 7.1.3.1.7(a);
USR = 7.1.3.1.9;
USR = 7.1.3.1.9(c);

1.1.1.5.1 Collect Roadway Condition Data

Description:

Overview: This process shall be responsible for collecting data obtained from roadway condition sensors. This data will provide information about environmental hazards (e.g. ice, fog, mudslides) affecting the roadway. Where any of the data is provided in analog form, the process shall be responsible for converting it into digital form and calibrating. The converted data shall be sent to other processes for distribution, further analysis and storage.

In Subsystem:

Roadway Subsystem

In Deployment Package:

ATMS20

ATMS21

Data Flows:

Input Flows

fre_environmental_hazards
fre_physical_conditions
roadside_sensor_configuration_data

Output Flows

roadside_condition_data
roadway_condition_coordination
roadway_condition_for_traffic

User Service Requirements Mapped to Pspec:

USR = C2.4.1.1;
USR = C2.4.1.2;
USR = C8.1.1.1;
USR = C8.1.3.1;

1.1.1.5.2 Provide Other Roadside Inputs

Description:

Overview: This process shall exchange data with similar processes in other Roadway Subsystems. The other RS can be adjacent geographically, under control of a different jurisdiction, or part of a more complex hierarchy. The exchange of data shall be triggered by either a request from a remote RS for data from the RS to which the Manage Traffic function belongs, or because data needs to be sent from the local RS to a remote RS. This data shall include traffic conditions around a work zone, environmental conditions (fog detected, etc.), roadway condition information (e.g. debris in roadway),dynamic roadway warning data, a road user protection warning, or a speed restriction request. The data received from other RS's to control local devices is handled by the Recieve Other Roadside Inputs. This process supports the collection of data locally in the roadway that might be needed to trigger an update to a nearby dynamic message sign such as icy conditions in a remote area or workers present in a work zone. This process and its companion process to Receive Other Roadside Inputs support autonomous traffic control without the need for direct control from a Traffic Management Subsystem.

Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Deployment Package:

ATMS26

ATMS30

ATMS25

Data Flows:

Input Flows

environmental_coordination
fors_other_roadside_request
road_user_protection_warning
roadway_condition_coordination
roadway_warning_data_coordination
speed_restriction_request
traffic_sensor_coordination

Output Flows

tors_other_roadside_inputs

User Service Requirements Mapped to Pspec:

USR = C2.6.1.3;
USR = C2.6.1.4;
USR = C2.6.1.5;
USR = C2.7.1.5;
USR = C2.7.2.4;
USR = C2.7.2.5;

1.1.1.5.3 Process Roadway Warning Data

Description:

Overview: This process collects data using sensors alongside the roadway to detect hazardous conditions and sends the information to other processes to warn approaching drivers. The inputs collected by this function include roadway conditions, environmental hazards, obstacles in or near the roadway such as animals, or vehicle characteristics. This process notifies other traffic management functions including enforcement functions to warn drivers to adjust speed to avoid a dangerous driving situation. Inputs from traffic operators may include forecast whether conditions to use in determining appropriate warning parameters. This process detects the characteristics of oncoming vehicles and determines whether conditions dictate a warning such as a high-profile commercial vehicle approaching an area of high cross winds.

Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Deployment Package:

ATMS26

ATMS27

ATMS30

Data Flows:

Input Flows

fe_roadway_environmental_hazards
From_Potential_Obstacles
From_Roadway
fvc_vehicle_characteristics
road_user_dynamic_warning
roadway_warning_sensor_configuration

Output Flows

roadway_warning_data_coordination
roadway_warning_speed_conditions
roadway_warning_traffic_data

User Service Requirements Mapped to Pspec:

USR = C2.6.1.1;
USR = C2.6.1.3;
USR = C2.6.1.4;
USR = C2.6.1.5;

1.1.1.5.4 Process Road User Protection

Description:

Overview: This process collects data using sensors alongside the roadway to detect vehicles in close proximity to other road users such as pedestrians and bicyclists. Data concerning the presence of non-vehicle users is input to this process from the Process Traffic Sensor Data. In some cases this may include the manual press of a pedestrian call button at a cross walk or it could include the automatic detection of such users near the roadway. When this process determines that non-vehicle users are too close to vehicles in the roadway this process generates warning data that is used to generate speed restriction requests or warnings either directly at the roadside or to a function residing at a traffic management centre.

Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Deployment Package:

ATMS29

ATMS30

Data Flows:

Input Flows

fp_pedestrian_presence
fvc_vehicle_presence
pedestrian_presence

Output Flows

road_user_dynamic_warning
road_user_protection_warning
speed_restriction_request

User Service Requirements Mapped to Pspec:

USR = C2.7.1.1;
USR = C2.7.1.2;
USR = C2.7.1.3;
USR = C2.7.1.4;
USR = C2.7.1.5;
USR = C2.7.2.1;
USR = C2.7.2.2;
USR = C2.7.2.3;
USR = C2.7.2.5;

1.1.2.1 Process Traffic Data for Storage

Description:

Overview: This process shall receive data from other processes and store the data into the long term and current data stores. The data shall comprise sensor data, both smoothed and unsmoothed: processed sensor surveillance data, data sent to control indicators (output devices e.g. intersection controllers, pedestrian controllers, dynamic message signs, ramp metering equipment), parking lot management data and other street equipment, the status data received from the indicators, plus current traffic conditions, planned events, current incidents, parking lot states, freeway ramp states, link travel times, roadway conditions provided by vehicle probes, and selected traffic control strategy. The data stored by the process in the current data store shall be the values collected over a relatively short period of time. The data stored in the long term data store shall be retained for a longer period. The data retained in the long term data store may be aggregated so as to reduce the storage requirements for long historical records, the amount of aggregation to be an implementation decision.

In Subsystem:

Traffic Management

In Deployment Package:

AD4

ATMS01

ATMS09

ATMS18

Data Flows:

Input Flows

current_highway_network_data
current_incident_data
current_ramp_state
current_road_network_data
current_road_network_use
hov_lane_data
indicator_control_storage_data_for_highways
indicator_control_storage_data_for_roads
indicator_input_storage_data_for_highways
indicator_input_storage_data_for_roads
link_data_from_avl
link_data_from_tags
parking_lot_current_state
planned_event_data
processed_data
selected_strategy
sensor_output_data
vehicle_smart_probe_data_for_storage
wide_area_pollution_data

Output Flows

current_data
incidents_for_maintenance
long_term_data
traffic_data_for_maintenance

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.2;
USR = 1.6.2.5;
USR = 1.6.2.5.1;
USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.6;
USR = 1.8.1.6(f);
USR = 1.8.2;
USR = 1.8.2.1;
USR = 1.8.2.1(b);

1.1.2.2 Process Traffic Data

Description:

Overview: This process shall receive and process data from sensors (both traffic and environmental) at the roadway. The process distributes data to Provide Device Control processes that are responsible for freeway, highway rail intersections, parking lot, surface street and freeway management. It also sends the data to another Provide Traffic Surveillance process for loading into the stores of current and long term data. Information about the various sensors to aid in this processing and distribution of data is accessed from the data store static_data_for_sensor_processing.

In Subsystem:

Traffic Management

In Deployment Package:

AD4

ATMS01

ATMS03

ATMS09

ATMS18

ATMS29

Data Flows:

Input Flows

environment_sensor_data
hov_sensor_data
hri_sensor_data
multimodal_crossing_sensor_data
pedestrian_sensor_data
roadway_traffic_maintenance_information
roadway_warning_traffic_data
static_data_for_sensor_processing
traffic_sensor_data
traffic_video_image

Output Flows

parking_lot_input_data
processed_data
ramp_data
sensor_output_data
strategy_data_for_highways
strategy_data_for_roads
traffic_surveillance_data
unusual_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.2;
USR = 1.6.2.2;
USR = 1.6.2.2.1;
USR = 1.6.2.3;
USR = 1.6.2.3.2;
USR = 1.6.2.4;
USR = 1.6.2.4.1;
USR = 1.7;
USR = 1.7.1;
USR = 1.7.1.1;
USR = 1.7.1.1.1;
USR = 1.7.1.1.1(a);
USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.4;
USR = 1.8.1.4(a);
USR = 1.8.2;
USR = 1.8.2.3;
USR = 1.8.2.3(a);
USR = 1.8.2.3(d);
USR = 1.10;

1.1.2.3 Update Data Source Static Data

Description:

Overview: This process shall be responsible for the maintenance of the store of static data used in the processing of sensor data. This sensor data shall be used to provide traffic surveillance information for use by other processes within the Manage Traffic function. The store shall contain data showing the relationship between sensors and the surface street and freeway network, i.e. where they are located, to which part(s) of the network their data applies, the type of data, etc.
It shall also hold information about the ownership of each link (that is, the agency or entity responsible for collecting and storing surveillance of the link) in the network which shall be used by processes involved in exchanging surveillance information (and optionally control) with other Traffic Management Subsystems (TMS's). The contents of the store shall be provided by the Plan System Deployment function.

In Subsystem:

Traffic Management

In Deployment Package:

ATMS01

Data Flows:

Input Flows

link_data_update
new_sensor_static_data
request_sensor_static_data

Output Flows

existing_sensor_static_data
link_details
static_data_for_sensor_processing

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.2;
USR = 1.6.2.4;
USR = 1.6.2.4.1;

1.1.2.4 Monitor HOV lane use

Description:

Overview: This process shall be responsible for monitoring the use of High Occupancy Vehicle (HOV) lanes and detecting vehicles that do not have the required number of occupants. The process also provides data on HOV lane usage for storage in the stores of current and long term data.

In Subsystem:

Traffic Management

In Deployment Package:

ATMS05

Data Flows:

Input Flows

hov_lane_data_input
static_data_for_sensor_processing

Output Flows

hov_lane_data
hov_lane_violation

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.3;
USR = 1.6.3.4;
USR = 1.6.3.4(d);
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;
USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.2;
USR = 1.8.1.2(b);
USR = 1.8.1.3;
USR = 1.8.1.3(b);
USR = 1.8.2;
USR = 1.8.2.4;
USR = 1.8.2.4(b);
USR = 1.8.2.11;
USR = 1.8.2.11(b);

1.1.2.5 Process Tag/AVL Data for Link Time Data

Description:

Overview: This process shall be responsible for processing tag and AVL data collected from roadside readers or obtained from an analysis of toll transaction records. The process shall receive the processed tag and AVL data from the data collection process and shall calculate the travel time for the links under tag surveillance that have been traveled by the vehicles carrying the tags. This shall be achieved by noting the successive times at which the tag data is received and calculating the travel time from the difference. The data obtained from the toll tag transaction record analysis and/or tag reader locations shall not need any further processing as it shall contain the average travel times between successive toll collection plazas and tag reading locations. The process shall maintain a data store which contains the average travel time for each link in the surface street and freeway network under tag surveillance calculated from the previously described data. Calculation of the actual average values shall employ some type of aggregation processing (e.g. smoothing or similar technique) and be stored for differing time categories (e.g. times of day, day of week, holidays) in periodic increments. The current delay time for a link shall be the difference between current travel time value and the aggregate processed (e.g. average) value for that time category.

In Subsystem:

Traffic Management

In Deployment Package:

ATMS02

Data Flows:

Input Flows

link_time_calculation_read
probe_data_for_traffic
static_data_for_sensor_processing
transit_probe_data
vehicle_tag_data

Output Flows

link_data_from_avl
link_data_from_tags
link_time_calculation_write

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.2;
USR = 1.6.2.2;
USR = 1.6.2.4;
USR = 1.6.2.4.1;
USR = 1.6.2.5.1;

1.1.2.6 Process Collected Vehicle Smart Probe Data

Description:

Overview: This process shall be responsible for the processing of vehicle smart probe data. The process receives data from vehicle subsystems and processes the data to estimate type and level of roadway conditions and hazards. The process shall send the road condition and hazard estimates to the Provide Device Control facility for output to future passing vehicles. It shall send this data, together with the fixed unit identity and fixed location to the traffic data storage process for loading into the current and long term data stores.

In Subsystem:

Roadway Subsystem

In Deployment Package:

ATMS12

Data Flows:

Input Flows

vehicle_smart_probe_input_data

Output Flows

vehicle_smart_probe_data_for_storage
vehicle_smart_probe_data_indication

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.2;
USR = 1.6.2.2;
USR = 1.6.2.3;
USR = 1.6.2.3.1;
USR = 1.6.2.3.2;
USR = 1.6.2.4;
USR = 1.6.2.5;
USR = 1.8.2;
USR = 1.8.2.10;
USR = 1.8.2.10(b);
USR = 1.9;
USR = 1.9.0;
USR = 1.9.2;
USR = 1.9.2.1;
USR = 1.9.2.1.3;

1.1.2.7 Monitor Reversible Lanes

Description:

Overview: This process shall be responsible for monitoring the use of reversible lanes and detecting wrong-way vehicles in reversible lanes. The process shall monitor sensor data and video images from the reversible lanes, and use this information along with the lane status (which direction it is currently operating) to identify when a vehicle is traveling in the wrong direction on the reversible lane.

In Subsystem:

Traffic Management

In Deployment Package:

ATMS17

Data Flows:

Input Flows

reversible_lane_status
reversible_lane_video_images
sensor_data_for_reversible_lanes
static_data_for_sensor_processing

Output Flows

wrong_way_vehicle_detection

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.1;
USR = 1.7.1.1;
USR = 1.7.1.1.1;
USR = 1.7.1.1.1(a);
USR = 1.7.4;

1.1.2.8 Manage Environmental Data for Traffic

Description:

Overview: This process is responsible for gathering environmental and road condition sensor data. The data gathered by this process shall include the outputs of roadside environmental sensors and road condition sensors. The data may be fused with weather forecasts from the Weather Service, and weather prediction information from the Meteorological Service Provider. The data is fused by the process to create an improved forecast of local weather and pavement conditions. This process is responsible for creating road forecasts using either the fused weather related information or by directly utilizing sensor data. The process shall support the creation of meso and micro-predictions using algorithms to predict road conditions over individual sections of the roadway.

Functional Requirements: none.

In Subsystem:

Traffic Management

In Deployment Package:

ATMS20

ATMS21

ATMS22

Data Flows:

Input Flows

environment_sensor_data
environmental_data_from_maintenance
From_Meteorological_Service_Provider
From_Weather_Service
road_condition_data
roadway_condition_for_traffic

Output Flows

environment_data_from_traffic
environmental_data_from_traffic
tmsp_maintenance_roadway_data
tws_traffic_roadway_data

User Service Requirements Mapped to Pspec:

USR = C2.4.1.2;
USR = C2.4.1.3;
USR = C2.4.1.4;
USR = C8.1.1.1;
USR = C8.1.1.2;
USR = C8.1.1.3;
USR = C8.1.1.4;
USR = C8.1.1.5;
USR = C8.1.2.1;
USR = C8.1.2.2
USR = C8.1.3.1;
USR = C8.1.3.2
USR = C8.1.3.3;
USR = C8.1.3.4;
USR = C8.1.3.5;

1.1.3 Generate Predictive Traffic Model

Description:

Overview: This process shall be responsible for continually producing and updating a predictive model of the traffic flow conditions in the road or freeway network served by the Manage Traffic function that an instance of this process is allocated to. The prediction shall be based on current surveillance, historic traffic data and surveillance, current incidents, planned events, current traffic control strategy, data received from other Traffic Management Subsystems (TMS's) serving other geographic and/or jurisdictional areas, and current and predicted weather conditions. The predictive model of traffic flow produced by this process shall be used by processes in the Manage Traffic function and other ITS functions.

In Subsystem:

Traffic Management

In Deployment Package:

ATMS09

ATMS15

Data Flows:

Input Flows

current_incident_data
fws_predicted_weather
long_term_data
other_traffic_center_data
planned_events
selected_strategy

Input/Output

predictive_model_data

Output Flows

prediction_data
unusual_congestion

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2;
USR = 1.2.3;
USR = 1.2.3.2;
USR = 1.2.3.2.3;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.2;
USR = 1.6.2.5;
USR = 1.6.2.5.2;

1.1.4.1 Retrieve Traffic Data

Description:

Overview: This process shall on request retrieve traffic data from the data stores managed by other processes in the Provide Traffic Surveillance facility of the Manage Traffic function. It shall be possible for requests to originate from traffic operations personnel, the media operator, the Manage Demand facility within the Manage Traffic function, the Plan System Deployment function and the Provide Driver and Traveler Services function. With the exception of those from the Manage Demand facility and the Plan System Deployment function, all requests shall be provided by interface processes. The process shall also generate traffic data for output by other processes to in-vehicle signage functions.

In Subsystem:

Traffic Management

In Deployment Package:

AD4

ATMS01

ATMS02

ATMS06

ATMS18

Data Flows:

Input Flows

current_data
long_term_data
predictive_model_data
request_traffic_media_data
request_traffic_operations_data
traffic_data_demand_request
traffic_data_deployment_request
traffic_data_distribution_request

Output Flows

incident_video_for_emergency_services
operator_log_for_traffic_data
retrieved_traffic_media_data
retrieved_traffic_operations_data
sensor_data_for_distribution
traffic_data_for_demand
traffic_data_for_deployment
traffic_data_for_distribution
traffic_data_for_emergency_services
traffic_data_for_signage
traffic_data_for_transit

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.3;
USR = 1.6.3.4;
USR = 1.6.3.4.1;
USR = 1.6.4;
USR = 1.6.4(a);
USR = 1.6.4(b);
USR = 1.6.4(c);
USR = 1.6.4(d);

1.1.4.2 Provide Traffic Operations Personnel Traffic Data Interface

Description:

Overview: This process shall provide the interface through which traffic operations personnel can obtain access to the data stored by other processes in the Provide Traffic Surveillance facility of the Manage Traffic function, and set up the parameters that govern the data that is available to non-traffic operations people via a separate process to the media operator. This stored data shall comprise current and long term (historic) data on traffic conditions, weather conditions and roadside equipment activity, plus prediction estimates of traffic conditions. The data shall apply to some or all of the surface street and freeway network served by the specific instance of the Manage Traffic function. Where appropriate and/or requested by the traffic operations personnel, the process shall provide the data output in the form of an overlay onto a map of the relevant part(s) of the surface street and freeway network served by the instance of the function. The process shall obtain the map from a local data store, which it shall enable the traffic operations personnel to update as and when required.

In Subsystem:

Traffic Management

In Deployment Package:

ATMS01

ATMS03

ATMS04

ATMS06

ATMS07

Data Flows:

Input Flows

ftop_traffic_data_parameter_updates
ftop_traffic_information_requests
ftop_weather_request_information
map_data_for_traffic_display
retrieved_traffic_operations_data
traffic_video_image_for_display
weather_service_information

Input/Output

operator_log_for_traffic_data

Output Flows

environment_sensor_configuration_data
request_traffic_map_display_update
request_traffic_operations_data
roadway_warning_sensor_configuration
sensor_configuration_data
traffic_data_media_parameters
ttop_traffic_control_information_display
ttop_video_image_output
ttop_weather_information
weather_service_information_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.1;
USR = 1.6.1.7;
USR = 1.6.1.7(a);
USR = 1.6.3;

1.1.4.3 Provide Direct Media Traffic Data Interface

Description:

Overview: This process shall be responsible for providing the interface between the media and the process responsible for obtaining data from the stores of traffic data maintained by other processes within the Provide Traffic Surveillance facility of the Manage Traffic function. The process shall enable the media to request and be provided with current, long term (historic) and predicted traffic data. The data may be provided in one or more formats: as a data stream, as processed and displayed to Traffic Operations Personnel (e.g. graphical summaries of link speeds),or as a display (with data included on a map of relevant part(s) of the road and freeway served by the Manage Traffic function. The media shall only be able to request and see displayed that data that the traffic operations personnel have made available, through the use of the definition in the traffic data media parameters.

In Subsystem:

Traffic Management

In Deployment Package:

ATMS06

Data Flows:

Input Flows

fm_traffic_data_request
map_data_for_traffic_display
retrieved_traffic_media_data
traffic_data_media_parameters

Output Flows

request_traffic_media_data
tm_traffic_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.1;
USR = 1.6.1.7;
USR = 1.6.1.7(a);
USR = 1.7;
USR = 1.7.0;
USR = 1.7.3;
USR = 1.7.3.3;

1.1.4.4 Update Traffic Display Map Data

Description:

Overview: This process shall provide updates to a store of digitized map data when a request is received from traffic operations personnel via their interface process. The map data shall be for use as the background for displays of traffic data requested by traffic operations personnel and media operators through their respective interface processes. This process shall obtain the new map data from either a specialized data supplier or some other appropriate data source.

In Subsystem:

Traffic Management

In Deployment Package:

ATMS01

Data Flows:

Input Flows

fmup_traffic_display_update
request_traffic_map_display_update

Output Flows

map_data_for_traffic_display
tmup_request_traffic_display_update

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.3;
USR = 1.6.3.4(e);
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
URS = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.9;
USR = 7.1.3.1.9(e);

1.1.4.5 Provide Media System Traffic Data Interface

Description:

Overview: This process shall provide the interface through which traffic and incident data can be output to the Media. The output shall comprise traffic and incident data that is suitable for output to the Media System as determined by traffic managers. This interface is only for the output of data that has been requested by the Media.

In Subsystem:

Information Service Provider

In Deployment Package:

ATIS1

ATIS2

ATIS4

Data Flows:

Input Flows

fm_incident_details
fm_incident_information_request
fm_traffic_information_request
information_for_media

Output Flows

incident_details_from_media
tm_incident_information
tm_traffic_information

User Service Requirements Mapped to Pspec:

USR = NA;

1.1.4.6 Provide Traffic Data Retrieval Interface

Description:

Overview: This process shall provide customized sets of traffic data for broadcast, advisories, and personalized data to travelers, traveler information data archive, and the media. This process shall use the parameters in the data store 'traffic_data_retreival_parameters' to define exactly what data shall be retrieved as a result of each request. The process shall select the appropriate subset of traffic data which will be sent to each ITS function which is requesting data. The process shall accept traveler profiles for use in determining what personalized data to send to the traveler. The process shall send kiosk and personal traffic requests to the archival process.

In Subsystem:

Information Service Provider

In Deployment Package:

ATIS6

ATIS8

ATIS1

ATIS2

ATIS4

ATIS5

Data Flows:

Input Flows

current_traffic_pollution_data
incident_details_from_media
maintenance_status_for_travellers
sensor_data_for_distribution
traffic_data_advisory_request
traffic_data_for_distribution
traffic_data_guidance_request
traffic_data_kiosk_request
traffic_data_personal_request
traffic_data_ridesharing_request
traveler_traffic_profile

Input/Output

traffic_data_retrieval_parameters

Output Flows

information_for_media
traffic_data_distribution_request
traffic_data_for_advisory_output
traffic_data_for_broadcast_to_kiosks
traffic_data_for_broadcast_to_personal_devices
traffic_data_for_guidance
traffic_data_for_kiosks
traffic_data_for_personal_devices
traffic_data_for_ridesharing
traffic_data_kiosk_request_for_archive
traffic_data_personal_request_for_archive

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.1;
USR = 1.1.4;
USR = 1.1.4.1;
USR = 1.1.4.1.1;
USR = 1.1.4.1.2;
USR = 1.1.4.1.3;
USR = 1.1.4.1.4;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.3;
USR = 1.6.3.4;
USR = 1.6.3.4.1;
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;
USR = 7.1.3.1.8(g);

1.1.4.7 Manage Traffic Archive Data

Description:

Overview: This process shall collect traffic data and ahs operational data to distribute to the Manage Archive Data function. The process shall run when a request for data is received from an external source, or when fresh data is received.

In Subsystem:

Traffic Management

In Deployment Package:

AD1

AD2

AD4

Data Flows:

Input Flows

ahs_operational_data
ftop_archive_command
static_data_for_archive
traffic_data_for_deployment
traffic_management_archive_request
traffic_management_archive_status

Input/Output

traffic_data_archive

Output Flows

traffic_data_deployment_request
traffic_management_archive_data
ttop_archive_status

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.1;
USR = 7.1.3.1.1(a);
USR = 7.1.3.1.1(b);
USR = 7.1.3.1.1(c);
USR = 7.1.3.1.1(d);
USR = 7.1.3.1.1(e);
USR = 7.1.3.1.2;
USR = 7.1.3.1.3;
USR = 7.1.3.1.3(e);
USR = 7.1.3.1.5;
USR = 7.1.3.1.5(e);
USR = 7.1.3.1.5(g);
USR = 7.1.3.1.8;
USR = 7.1.3.1.8(b);
USR = 7.1.3.1.9;
USR = 7.1.3.1.9(a);

1.1.4.8 Provide Disaster Command Data

Description:

Overview: This process shall be responsible for providing the interface between the Disaster Command Authority and the process responsible for obtaining data from the stores of traffic data maintained by other processes within the Provide Traffic Surveillance facility of the Manage Traffic function. The process shall enable the Disaster Command Authority to request and be provided with current, long term (historic) and predicted traffic data. The data may be provided in one or more formats: as a data stream, as processed and displayed to Traffic Operations Personnel (e.g. graphical summaries of link speeds),or as a display (with data included on a map of relevant part(s) of the road and freeway served by the
Manage Traffic function.

Functional Requirements: none.

In Subsystem:

Traffic Management

In Deployment Package:

EM4

EM5

Data Flows:

Input Flows

disaster_traffic_data
fdca_disaster_traffic_conditions_request

Output Flows

disaster_traffic_request
tdca_disaster_traffic_conditions_data

User Service Requirements Mapped to Pspec:

USR = C6.3.1.4;
USR = C6.3.1.5;
USR = C6.3.2.1;
USR = C6.3.2.2;

1.1.5 Exchange data with Other Traffic Centers

Description:

Overview: This process shall exchange data with similar processes in other Traffic Management Subsystems (TMS's). The other TMS can be adjacent geographically, under control of a different jurisdiction, or part of a more complex hierarchy. The exchange of data shall be triggered by either a request from a remote TMS for data from the TMS to which the Manage Traffic function belongs, or because data needs to be sent from the local TMS to a remote TMS. This data shall include traffic control preemption for vehicle routes which pass through the local network but have a destination in an area served by a remote TMS, or include data about an incident that has an impact on the traffic conditions in the network served by a remote TMS. The data received from remote TMS's shall be used either to vary the current traffic control strategy to give signal preemption to emergency vehicles or enable the passage of commercial vehicles with unusual loads, or as input to the local traffic predictive model estimation process.

In Subsystem:

Traffic Management

In Deployment Package:

ATMS07

ATMS08

ATMS09

CVO10

Data Flows:

Input Flows

control_data_for_highways
control_data_for_roads
current_data
cv_incidents_for_other_TMC
emergency_data_for_other_TMC
fotc_data_request
fotc_identity
fotc_traffic_control_and_status
fotc_transfer_data
link_details
long_term_data
other_status_for_highways
other_status_for_roads
planned_events_local_data
request_other_current_incidents_data
request_other_planned_events_data
request_other_TMC_data

Output Flows

other_control_data_for_highways
other_control_data_for_roads
other_current_incidents
other_planned_events
other_TMC_cv_incidents
other_TMC_emergency_data
other_TMC_strategy_data
other_traffic_center_data
request_local_current_incidents_data
request_local_planned_events_data
status_data_for_highways
status_data_for_roads
totc_data_request
totc_identity
totc_traffic_control_and_status

totc_transfer_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.2;
USR = 1.6.2.5;
USR = 1.6.2.5.2;
USR = 1.6.3;
USR = 1.6.3.6;
USR = 1.6.4;
USR = 1.6.4(a);
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.9;
USR = 7.1.3.1.9(d);

1.1.6 Collect Vehicle Tag Data for Link Time Calculations

Description:

Overview: This process shall collect data from toll and parking tags on passing vehicles. This shall be achieved by transmitting a tag data request message and collecting any tag reply data that is received. This reply data shall be translated by the process into a unique but anonymous ID that does not store or transmit the identity of the tag in any way that is traceable to the tag owner, e.g. credit identity or stored credit value. This ID is then passed on to another process for further link travel time calculation analysis.

In Subsystem:

Roadway Subsystem

In Deployment Package:

ATMS02

ATMS12

Data Flows:

Input Flows

parking_lot_tag_data_input
toll_tag_data_input

Output Flows

parking_lot_tag_data_needed
toll_tag_data_needed
vehicle_tag_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.2;
USR = 1.6.2.2;
USR = 1.6.2.4;
USR = 1.6.2.4.1;
USR = 1.6.2.5.1;

1.1.7 Collect Vehicle Smart Probe Data

Description:

Overview: This process shall collect data from vehicle smart probes. This data shall include information about conditions in the vicinity of the vehicle operating as a smart probe. It shall receive this data from passing vehicles and shall add its own identity and location before sending the data on to the process which outputs the data.

In Subsystem:

Roadway Subsystem

In Deployment Package:

ATMS12

Data Flows:

Input Flows

vehicle_smart_probe_data

Output Flows

vehicle_smart_probe_input_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.2;
USR = 1.6.2.2;
USR = 1.6.2.3;
USR = 1.6.2.3.1;
USR = 1.6.2.3.2;
USR = 1.6.2.4;
USR = 1.6.2.5;

1.2.1 Select Strategy

Description:

Overview: This process shall select the appropriate traffic control strategy to be implemented over a road and/or freeway section served by the specific instance of the Manage Traffic function. The strategy shall be selected by the process from a number that are available, e.g. adaptive control, fixed time control, local operations. The selected strategy shall be passed by the process to the actual control processes for implementation according to the part of the network to which it is to be applied, i.e. surface roads, freeways (i.e. limited access roads), ramps and/or parking lots. The definition of strategy can be extended to include a strategy for the operations of sensors such as video cameras used to provide traffic surveillance data. The process shall make it possible for the current strategy selection to be modified to accommodate the effects of such things as incidents, emergency vehicle preemption, the passage of commercial vehicles with unusual loads, equipment faults and overrides from the traffic operations personnel. The strategy for control of freeways and parking lots is through use of DMS signs and lane indicators. The strategy for control of ramps is through the timing plans for ramp meters. The selected strategy shall be sent to the process within the Provide Traffic Surveillance facility responsible for maintaining the store of long term data.

In Subsystem:

Traffic Management

In Deployment Package:

ATMS03

Data Flows:

Input Flows

current_road_network_use
cv_incident_override
demand_overrides
emergency_traffic_control_request
ftop_strategy_override
ftop_video_camera_strategy_change
incident_strategy_override
indicator_fault_state
indicator_input_state_for_highways
indicator_input_state_for_roads
other_TMC_cv_incidents
other_TMC_emergency_data
other_TMC_strategy_data
special_vehicle_priority_routing
static_data_for_strategy

Output Flows

cv_incidents_for_other_TMC
emergency_data_for_other_TMC
emergency_traffic_control_response
request_other_TMC_data
selected_highway_control_strategy
selected_hri_control_strategy
selected_parking_lot_control_strategy
selected_ramp_control_strategy
selected_road_control_strategy
selected_strategy
video_camera_control_strategy

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.3;

1.2.2.1 Determine Indicator State for Freeway Management

Description:

Overview: This process shall implement selected traffic control strategies and transit vehicle overall priority on some or all of the indicators covering the freeway network served by the Manage Traffic function. It shall implement the strategies only using the indicators (e.g. dynamic message signs (DMS)) specified in the implementation request and shall coordinate its actions with those of the process that controls the road network. The process shall also be capable of monitoring the extra inputs that will arise where tunnels are involved, including the detection of fire and the consequent requirement to re-route traffic.

In Subsystem:

Traffic Management

In Deployment Package:

APTS7

ATMS04

Data Flows:

Input Flows

coordination_data_roads_to_freeways
prediction_data
selected_highway_control_strategy
static_data_for_highways
strategy_data_for_highways
transit_highway_overall_priority

Output Flows

coordination_data_freeways_to_roads
current_highway_network_data
current_highway_network_state
indicator_highway_requested_state
transit_highway_priority_given

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.1;
USR = 1.6.1.1;
USR = 1.6.1.1.1;
USR = 1.6.1.1.2;
USR = 1.6.1.1.3;
USR = 1.6.1.1.4;
USR = 1.6.1.1.5;
USR = 1.6.1.2;
USR = 1.6.1.2.1;
USR = 1.6.1.2.2;
USR = 1.6.1.2.3;
USR = 1.6.1.3;
USR = 1.6.1.4;
USR = 1.6.1.4.1;
USR = 1.6.1.5;
USR = 1.6.1.6;
USR = 1.6.1.7;
USR = 1.6.1.7(b);
USR = 1.6.3;
USR = 1.6.3.1;
USR = 1.6.3.2;
USR = 1.6.3.2.1;

USR = 1.6.3.2.2(a);
USR = 1.6.3.3;
USR = 1.6.3.3.1;
USR = 1.6.3.3.2;
USR = 1.6.3.3.3;
USR = 1.6.3.3.4;
USR = 5.0;
USR = 5.2;
USR = 5.2.3;
USR = 5.2.3.1;

1.2.2.2 Determine Indicator State for Road Management

Description:

Overview: This process shall implement selected traffic control strategies and transit priority on some or all of the indicators covering the road (surface street) network served by the Manage Traffic function. It shall implement the strategies only using the indicators (intersection and pedestrian controllers, variable message signs (dms), etc.) that are specified in the implementation request and shall coordinate its actions with those of the processes that control the freeway network and the ramps that give access to the freeway network.

In Subsystem:

Traffic Management

In Deployment Package:

APTS7

ATMS03

Data Flows:

Input Flows

coordination_data_freeways_to_roads
coordination_data_ramps_to_roads
prediction_data
selected_hri_control_strategy
selected_road_control_strategy
static_data_for_roads
strategy_data_for_roads
transit_road_overall_priority

Output Flows

coordination_data_roads_to_freeways
coordination_data_roads_to_ramps
current_road_network_data
current_road_network_state
indicator_road_requested_state
transit_road_priority_given

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.1;
USR = 1.6.1.1;
USR = 1.6.1.1.1;
USR = 1.6.1.1.2;
USR = 1.6.1.1.3;
USR = 1.6.1.1.4;
USR = 1.6.1.1.5;
USR = 1.6.1.2;
USR = 1.6.1.2.1;
USR = 1.6.1.2.2;
USR = 1.6.1.2.3;
USR = 1.6.1.3;
USR = 1.6.1.4;
USR = 1.6.1.4.1;
USR = 1.6.1.5;
USR = 1.6.1.6;
USR = 1.6.1.7;
USR = 1.6.1.7(b);
USR = 1.6.3;
USR = 1.6.3.1;

USR = 1.6.3.2;
USR = 1.6.3.2.1;
USR = 1.6.3.2.2;
USR = 1.6.3.3;
USR = 1.6.3.3.1;
USR = 1.6.3.3.2;
USR = 1.6.3.3.3;
USR = 1.6.3.3.4;
USR = 5.0;
USR = 5.2;
USR = 5.2.3;
USR = 5.2.3.1;
USR = 5.2.3.2;
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.9;

1.2.3 Determine Ramp State

Description:

Overview: This process shall implement the selected control strategies on some or all of the freeway entry ramps in the freeway network served by the Manage Traffic function. It shall implement the strategies only using the ramps that are specified in the implementation request and shall coordinate its actions with those of the process that controls the road network. The process shall base its ramp metering decisions on the data from sensors and ramp meters monitoring traffic conditions upstream and downstream of the ramps. Data from sensors on the ramp used to detect flow past the meter, extent of queues on the ramp, and the presence of vehicles will also be used as the basis for the ramp metering decisions. The decision making process shall use an algorithm to determine the ramp's state based on the ramp control strategy and the sensor input data received. The process shall coordinate its activities with the process responsible for controlling the road(surface street) network.

In Subsystem:

Traffic Management

In Deployment Package:

APTS7

ATMS04

Data Flows:

Input Flows

coordination_data_roads_to_ramps
ramp_data
selected_ramp_control_strategy
static_data_for_ramps
transit_ramp_overall_priority

Output Flows

coordination_data_ramps_to_roads
current_ramp_state
ramp_signal_state
transit_ramp_priority_given

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.1;
USR = 1.6.1.1;
USR = 1.6.1.1.2;
USR = 1.6.1.2;
USR = 1.6.1.2.1;
USR = 1.6.1.2.3;

1.2.4.1 Output Control Data for Roads

Description:

Overview: This process shall transfer data to processes responsible for controlling equipment located at the roadside within the road (surface street) network served by the Manage Traffic function. This data shall contain outputs for use by roadside indicators, such as intersection and pedestrian controllers, dynamic message signs (DMS), highway advisory radio (HAR), etc. Data for use by in-vehicle signage equipment shall be sent to another process for output to roadside processes. All data shall be sent to this process by processes within the Manage Traffic function. This process shall also
be responsible for the monitoring of input data showing the way in which the indicators are responding to the data that they are being sent, and the reporting of any errors in their responses as faults to the Collect and Process Indicator Fault Data facility within the Manage Traffic function. All output and input data shall be sent by the process to another process in the Manage Traffic function to be loaded into the store of long term data.

In Subsystem:

Traffic Management

In Deployment Package:

AD4

ATMS03

ATMS07

ATMS08

CVO10

ATMS29

Data Flows:

Input Flows

dms_status_for_roads
dms_updates_for_roads
har_status_for_roads
hri_guidance_for_dms
indicator_input_data_from_roads
indicator_road_requested_state
other_control_data_for_roads
parking_guidance_for_dms
static_data_for_road_control
status_data_for_roads
vehicle_pollution_message_for_roads

Output Flows

control_data_for_roads
dms_data_for_roads
har_data_for_roads
indicator_control_data_for_roads
indicator_control_monitoring_data_for_roads
indicator_control_storage_data_for_roads
indicator_data_fault_for_roads
indicator_input_state_for_roads
indicator_input_storage_data_for_roads
indicator_sign_control_data_for_hri
other_status_for_roads
vehicle_sign_data_for_roads
warning_data_for_roads

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.5.2;
USR = 1.5.2.5;
USR = 1.5.2.5(a);
USR = 1.6;
USR = 1.6.0;
USR = 1.6.2;
USR = 1.6.1.2.1;
USR = 1.6.1.4;
USR = 1.6.3;
USR = 1.6.3.3;
USR = 1.6.3.3.1;
USR = 1.6.3.3.2;
USR = 1.6.3.3.3;
USR = 1.6.3.3.4;
USR = 1.6.3.4;
USR = 1.6.3.4(a);
USR = 1.6.3.4.1;
USR = 1.10;
USR = 1.10.3;
USR = 1.10.3.3;
USR = 1.10.3.3.5;
USR = 1.10.5;
USR = 1.10.5.2;
USR = 1.10.5.2.6;

1.2.4.2 Output Control Data for Freeways

Description:

Overview: This process shall transfer data to processes responsible for controlling equipment located at the roadside within the freeway network served by the Manage Traffic function. This data shall contain outputs for use by roadside indicators, such as dynamic message signs (DMS), etc. Data for use by in-vehicle signage equipment shall be sent to another process for output to roadside processes. All data shall have been sent to this process by processes within the Manage Traffic function. This process shall also be responsible for the monitoring of input data showing the way in which the indicators are responding to the data that they are being sent, and the reporting of any errors in their responses as faults to the Collect and Process Indicator Fault Data facility within the Manage Traffic function. All output and input data shall be sent by the process to another process in the Manage Traffic function to be loaded into the store of long term data.

In Subsystem:

Traffic Management

In Deployment Package:

ATMS04

ATMS05

ATMS07

ATMS08

ATMS17

CVO10

Data Flows:

Input Flows

dms_status_for_highways
dms_updates_for_highways
har_status_for_highways
indicator_highway_requested_state
indicator_input_data_from_highways
other_control_data_for_highways
parking_guidance_for_dms
ramp_signal_state
static_data_for_highway_control
status_data_for_highways
vehicle_pollution_message_for_highways

Output Flows

control_data_for_highways
dms_data_for_highways
har_data_for_highways
indicator_control_data_for_highways
indicator_control_monitoring_data_for_highways
indicator_control_storage_data_for_highways
indicator_data_fault_for_highways
indicator_input_state_for_highways
indicator_input_storage_data_for_highways
other_status_for_highways
vehicle_sign_data_for_highways
warning_data_for_highways

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.5.2;
USR = 1.5.2.5;
USR = 1.5.2.5(a);
USR = 1.6;
USR = 1.6.0;
USR = 1.6.2;
USR = 1.6.1.2.1;
USR = 1.6.1.4;
USR = 1.6.3;
USR = 1.6.3.3;
USR = 1.6.3.3.1;
USR = 1.6.3.3.2;
USR = 1.6.3.3.3;
USR = 1.6.3.3.4;
USR = 1.6.3.4;
USR = 1.6.3.4.1;

1.2.4.3 Output In-vehicle Signage Data

Description:

Overview: This process shall format and output data for use by roadside processes in creating in-vehicle signage. This process supports a full range of functionality for in-vehicle signage (from display of signage to location specific advisory data). The process shall be capable of outputting some or all of the following advisory data: link state data, current incidents, planned events, and highway rail intersection status. The process shall be capable of outputting some or all of the following signage data: dynamic message sign contents or fixed signage. The data shall be structured by this process so that it can be output by each roadside process to vehicles for use by in-vehicle signage equipment.

In Subsystem:

Traffic Management

In Deployment Package:

ATIS9

Data Flows:

Input Flows

current_incident_data_for_vehicle_signage
hri_guidance_for_beacon_message
planned_event_data_for_vehicle_signage
static_data_for_vehicle_signage
traffic_data_for_signage
vehicle_sign_data_for_highways
vehicle_sign_data_for_roads

Input/Output

vehicle_signage_incident_data

Output Flows

vehicle_sign_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.2;
USR = 1.6.1.2.1;
USR = 1.6.1.4;
USR = 1.6.3;
USR = 1.6.3.3;
USR = 1.6.3.3.1;
USR = 1.6.3.3.2;
USR = 1.6.3.3.3;
USR = 1.6.3.3.4;
USR = 1.6.3.4;
USR = 1.6.3.4.1;
USR = 1.10;
USR = 1.10.0;
USR = 1.10.1;
USR = 1.10.1.1;

1.2.5.1 Determine Parking Lot State

Description:

Overview: This process shall implement the selected control strategies on some or all of the parking lots in the surface street and freeway network served by the Manage Traffic function. It shall use the current parking lot occupancy provided by another process to determine the parking lot state to be used in sign settings implemented by other processes in the function, when this is not subject to a strategy override. The parking lot state shall be determined using threshold occupancy values contained in the static data provided by the Plan System Deployment function. Fixed thresholds for the states: 'full', 'almost full' and 'available' are in the data flow 'static_data_for_parking_lots'. In addition, threshold transitions can depend on whether the actual occupancy of the lot is increasing or decreasing, allowing hysteresis in the parking lot state transitions, so as to control jitter between parking lot states. The process shall also provide the current parking lot occupancy to other processes in the function.

In Subsystem:

Parking Management

In Deployment Package:

ATMS16

ATMS19

Data Flows:

Input Flows

parking_input_data
parking_lot_calculated_occupancy
parking_lot_operator_input_data
selected_parking_lot_control_strategy

Output Flows

parking_guidance_for_dms
parking_lot_current_state
parking_lot_occupancy
parking_lot_operator_output_data
parking_lot_state_for_archive
parking_output_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.8;
USR = 1.8.0;
USR = 1.8.2;
USR = 1.8.2.1;
USR = 1.8.2.1(c);
USR = 1.8.3;
USR = 1.8.3.1;

1.2.5.2 Coordinate Other Parking Data

Description:

Overview: This process shall process shall continuously communicate and exchange data with parking operators and systems. The exchange of data shall be triggered by either a request from a remote Parking Management Subsystem for data from the operators or systems the Provide Electronic Payment function belongs, or because data needs to be sent from the local Parking Management Substystem to another remote Parking Management Subsystem. This data shall include parking lot state, parking price information, parking availability, etc.

In Subsystem:

Parking Management

In Market Package:

ATMS19

Data Flows:

Input Flows

fop_parking_coordination_data
other_parking_lot_price_data
parking_lot_availability
parking_output_data
parking_transit_update

Output Flows

other_parking_lot_price_data_request
parking_input_data
top_parking_coordination_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;
USR = 1.8.2;
USR = 1.8.2.1;
USR = 1.8.2.1(c);

1.2.5.3 Provide Parking Lot Operator Interface

Description:

Overview: This process shall provide the interface to a local parking lot operator that controls the use of the lot. The operator shall provide inputs of occupancy and/or the current lot state to this process. This process shall provide the operator with outputs that request a change to the lot state, which the operator shall implement by activating local dynamic message signs (DMS) and controlling the use of entry/exit barriers, and data about transit services that provide a park and ride (P+R) operation to be output through local DMS.

In Subsystem:

Parking Management

In Market Package:

ATMS16

ATMS19

Data Flows:

Input Flows

fpo_current_lot_state
fpo_lot_occupancy
parking_lot_operator_output_data
parking_lot_operator_transit_update

Output Flows

parking_lot_operator_input_data
tpo_change_lot_state

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.2.5.4 Determine P+R needs for Transit Management

Description:

Overview: This process shall be responsible for calculating the need for transit services to provide a park and ride (P+R) operation at a parking lot. This calculation shall be based on the rate of change of the current parking lot occupancy. The results of the calculation shall be sent to the Manage Transit function in the form of a request for an additional (or reduced) level of service, depending on demand at the parking lot. The results of the request shall also be passed to other processes within the function.

In Subsystem:

Parking Management

In Market Package:

ATMS19

Data Flows:

Input Flows

parking_lot_occupancy
parking_lot_transit_response

Output Flows

parking_lot_operator_transit_update
parking_lot_transit_request
parking_transit_update

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;
USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.2;
USR = 1.8.1.2(a);

1.2.5.5 Manage Parking Archive Data

Description:

Overview: This process shall obtain parking lot availability and charge data and distribute it to the Manage Archive Data function. The process shall run when a request for data is received from an external source.

In Subsystem:

Parking Management

In Market Package:

AD1

AD2

Data Flows:

Input Flows

fpo_archive_commands
parking_archive_request
parking_archive_status
parking_charge_response_for_archive
parking_lot_state_for_archive

Input/Output

parking_data_archive

Output Flows

parking_archive_data
parking_charge_request_for_archive
tpo_archive_status

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;
USR = 7.1.3.1.8(e);
USR = 7.1.3.1.9;
USR = 7.1.3.1.9(e);
USR = 7.1.3.1.10;

1.2.5.6 Calculate Parking Lot Occupancy

Description:

Overview: This process shall calculate the occupancy of a parking lot based on processed traffic sensor data provided by other processes within the Manage Traffic function. The process shall use the static data for parking lots to determine the part(s) of the supplied data that apply to its entry and exit lanes, so that the numbers of vehicles entering and leaving can be calculated. These calculated flows shall be used by the process to generate the current parking lot occupancy.

In Subsystem:

Parking Management

In Market Package:

ATMS16

Data Flows:

Input Flows

parking_lot_input_data
static_data_for_parking_lots

Output Flows

parking_lot_calculated_occupancy

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.1;
USR = 1.1.0;
USR = 1.1.1;
USR = 1.1.1.1;
USR = 1.1.2;
USR = 1.1.2.1;
USR = 1.1.2.1.6;
USR = 1.5;
USR = 1.5.0;
USR = 1.5.2;
USR = 1.5.2.2;
USR = 1.5.2.2(c);
USR = 1.8;
USR = 1.8.0;
USR = 1.8.1;
USR = 1.8.1.4;
USR = 1.8.1.4(c);
USR = 1.8.2;
USR = 1.8.2.11;
USR = 1.8.2.11(a);
USR = 1.8.3;
USR = 1.8.3.1;
USR = 1.8.3.1(a);

1.2.6.1 Maintain Traffic and Sensor Static Data

Description:

Overview: This process shall maintain the store of static and link data used by other processes within the Manage Traffic function. Link data shall also be sent to the Provide Driver and Traveler Services function to enable it to obtain data about links that are not in the geographic area which it serves.

In Subsystem:

Traffic Management

In Market Package:

ATMS09

Data Flows:

Input Flows

current_incident_static_data
existing_sensor_static_data
ftop_roadway_characteristics
ftop_static_data
static_data_for_traffic_control_copy

Output Flows

link_data_for_guidance
link_data_update
new_sensor_static_data
request_sensor_static_data
static_data_for_traffic_control_update
static_data_store_updated
supply_incident_static_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.2;
USR = 1.8.1.2(a);
USR = 1.8.2;
USR = 1.8.2.13;
USR = 1.8.2.13(a);
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.9;
USR = 7.1.3.1.9(a);

1.2.6.2 Provide Static Data Store Output Interface

Description:

Overview: This process shall provide updates of static data to other processes in the Provide Traffic Control facility of the Manage Traffic function. An update of the data shall only be provided when this process has been notified by another process that the contents of the store of static data has been changed. This process shall provide updates to the map update provider about changes to the static data of a particular region.

In Subsystem:

Traffic Management

In Market Package:

ATMS09

Data Flows:

Input Flows

static_data_for_traffic_control_output
static_data_store_updated

Output Flows

maintenance_link_data
static_data_for_archive
static_data_for_highway_control
static_data_for_highways
static_data_for_parking_lots
static_data_for_ramps
static_data_for_road_control
static_data_for_roads
static_data_for_strategy
static_data_for_vehicle_signage
tmup_map_static_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.9;
USR = 7.1.3.1.9(a);

1.2.7.1 Process Indicator Output Data for Roads

Description:

Overview: This process shall implement the indicator output data generated by other processes within the Manage Traffic function for use on the roads (surface streets) served by the function. It shall perform the functions needed to provide control at intersections or pedestrian crossings, generate the output for dynamic message signs (dms) and highway advisory radios (HAR), or provide the interface for data to be sent to the units (or systems) that manage multimodal crossings. The dms may be either those that display variable text messages, or those that have fixed format display(s)(e.g. vehicle restrictions, or lane open/close).

In Subsystem:

Roadway Subsystem

In Market Package:

APTS7

ATMS03

ATMS06

ATMS13

ATMS17

EM2

Data Flows:

Input Flows

dms_data_for_roads
fmmc_crossing_status_for_roads
har_data_for_roads
hri_device_control
indicator_control_data_for_roads
indicator_preemption_override_for_roads
local_sensor_data_for_roads
other_roadway_data_for_roads
road_device_control_for_maintenance
road_information_maintenance_data
train_sense_data

Output Flows

dms_status_for_roads
har_fault_data_for_roads
har_status_for_roads
hri_device_sense
indicator_input_data_from_roads
indicator_response_data_for_roads
intersection_state_data
tbv_har_broadcast_for_roads
td_dms_indication_for_roads
td_lane_use_indication_for_roads
td_signal_indication
tmmc_crossing_clear_at_roads
tmmc_road_equipment_status
tmmc_stop_alternate_mode_at_roads
tp_cross_request_received
tp_cross_road
tp_dms_indication

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.5.2;
USR = 1.5.2.5;
USR = 1.5.2.5(a);
USR = 1.6;
USR = 1.6.0;
USR = 1.6.3;
USR = 1.6.3.3;
USR = 1.6.3.3.1;
USR = 1.6.3.3.2;
USR = 1.6.3.3.4;
USR = 1.6.3.4;
USR = 1.6.3.4(a);
USR = 1.10;
USR = 1.10.3;
USR = 1.10.3.1;
USR = 1.10.3.3;
USR = 1.10.3.3.2;
USR = 1.10.3.3.3;
USR = 1.10.4;
USR = 1.10.4.1;
USR = 1.10.5;
USR = 1.10.5.1;
USR = 1.10.5.2;
USR = 1.10.5.2.2;

1.2.7.2 Monitor Roadside Equipment Operation for Faults

Description:

Overview: This process shall monitor the operation of the processes that output in-vehicle signage, highway advisory radio, as well as indicator data in the road (surface street) and freeway network. It shall report any instances where the indicator response does not match that expected from the contents of the indicator control data it is receiving, the in-vehicle signage process reports a fault, or the HAR processes report a fault.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS03

ATMS04

Data Flows:

Input Flows

har_fault_data_for_highways
har_fault_data_for_roads
indicator_control_monitoring_data_for_highways
indicator_control_monitoring_data_for_roads
indicator_monitoring_suspend
indicator_response_data_for_highways
indicator_response_data_for_roads
vehicle_sign_data_output_fault
vehicle_smart_probe_data_output_fault

Output Flows

information_device_fault_status
traffic_control_device_status

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.2.7.3 Manage Indicator Preemptions

Description:

Overview: This process shall receive indicator (e.g. signal) preemption and priority requests from other functions within ITS. These requests shall enable the process to give selected vehicles (e.g. those that belong to Transit Authorities or Emergency Services) signal preemption or priority at intersections, pedestrian crossings and multimodal crossings in the surface street and freeway network served by the instance of the Manage Traffic function. Sending of the priority request output shall also generate an output to the monitoring process to suspend its activities while the priority request is being served. This process shall only generate its data flow outputs when input data is received.

In Subsystem:

Roadway Subsystem

In Market Package:

APTS7

EM2

Data Flows:

Input Flows

emergency_vehicle_preemptions
transit_vehicle_roadway_preemptions

Output Flows

indicator_monitoring_suspend
indicator_preemption_override_for_highways
indicator_preemption_override_for_roads

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.8;
USR = 1.8.0;
USR = 1.8.2;
USR = 1.8.2.8;
USR = 1.8.2.8(a);
USR = 1.8.2.8(b);
USR = 1.8.2.8(c);
USR = 1.8.3;
USR = 1.8.3.1;
USR = 1.8.3.1(c);
USR = 5.0;
USR = 5.2;
USR = 5.2.3;

1.2.7.4 Process In-vehicle Signage Data

Description:

Overview: This process shall output data for use by in-vehicle signage equipment on vehicles traveling along the road (surface street) and freeway network served by the Manage Traffic function. This data shall be able to provide information from any of the types of indicators that are supported by the function, e.g. intersection controller, pedestrian controller, dynamic message sign (dms), plus data about incidents and link information such as speed, travel times or roadway conditions. The process shall be responsible for its own fault monitoring, which shall check that output data is being sent and that it is an accurate representation of the input data. When a fault is detected this process shall report it to the process responsible for the monitoring of roadside equipment faults.

In Subsystem:

Roadway Subsystem

In Market Package:

ATIS9

ATMS12

Data Flows:

Input Flows

vehicle_sign_data

Output Flows

vehicle_sign_data_output_fault
vehicle_signage_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.2.7.5 Process Indicator Output Data for Freeways

Description:

Overview: This process shall implement the indicator output data generated by other processes within the Manage Traffic function for use on freeways served by the function. It shall perform the functions needed to output control data to ramp metering controllers and multimodal crossings, generate the output for dynamic message signs (dms), or generate the output for highway advisory radios(HAR). The dms may be either those that display variable text messages, or those that have fixed format display(s), for such things as vehicle restrictions, or lane open/close.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS04

ATMS05

ATMS06

ATMS17

Data Flows:

Input Flows

dms_data_for_highways
fmmc_crossing_status_for_highways
har_data_for_highways
highway_device_control_for_maintenance
highway_information_maintenance_data
indicator_control_data_for_highways
indicator_preemption_override_for_highways
local_sensor_data_for_highways
other_roadway_data_for_highways

Output Flows

dms_status_for_highways
har_fault_data_for_highways
har_status_for_highways
indicator_input_data_from_highways
indicator_response_data_for_highways
tbv_har_broadcast_for_highways
td_dms_indication_for_highways
td_lane_use_indication_for_highways
td_ramp_state_indication
tmmc_crossing_clear_at_highways
tmmc_highway_equipment_status
tmmc_stop_alternate_mode_at_highways

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.5.2;
USR = 1.5.2.5;
USR = 1.5.2.5(a);
USR = 1.6;
USR = 1.6.0;
USR = 1.6.3;
USR = 1.6.3.3;
USR = 1.6.3.3.2;
USR = 1.6.3.3.3;
USR = 1.6.3.4;
USR = 1.6.3.4(b);

1.2.7.6 Provide Intersection Collision Avoidance Data

Description:

Overview: This process shall provide collision avoidance data to vehicles that are approaching intersections served by the Manage Traffic function. The process shall use the data available from traffic sensors to determine any vehicle position conflict(s) that will arise if no action is taken. This process shall output data giving the direction from which the potential collision hazard will arise to the vehicle(s) that is(are) likely to receive the impact.

In Subsystem:

Roadway Subsystem

In Market Package:

AVSS05

AVSS10

Data Flows:

Input Flows

intersection_state_data
local_sensor_data_for_roads

Output Flows

intersection_collision_avoidance_data

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.2;
USR = 5.2.3;
USR = 1.10.3.2;

1.2.7.7 Process Vehicle Smart Probe Data for Output

Description:

Overview: This process shall output data about the conditions on roads and freeways. The process shall be provided with this data by other processes in the Manage Traffic function, which have received and processed data output by smart probes in vehicles. The data shall be output by the process for reception by those vehicles that are passing the deployed instance of this process (e.g. by dedicated short range communications). The process shall perform its own fault detection and report faults that are found to the fault monitoring process.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS12

Data Flows:

Input Flows

vehicle_smart_probe_data_indication

Output Flows

vehicle_smart_probe_data_output
vehicle_smart_probe_data_output_fault

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.2;
USR = 1.6.2.2;
USR = 1.6.2.3;
USR = 1.6.2.3.1;
USR = 1.6.2.3.2;
USR = 1.6.2.4;
USR = 1.6.2.4.1;
USR = 1.6.2.5;

1.2.7.8 Receive Other Roadside Inputs

Description:

Overview: This process shall exchange data with similar processes in other Roadway Subsystems. The other RS can be adjacent geographically, under control of a different jurisdiction, or part of a more complex hierarchy. The exchange of data shall be triggered by either a request for data or because data is received from the remote RS. This data shall include traffic conditions around a work zone, environmental conditions (fog detected, etc.), roadway condition information (e.g. debris in roadway), dynamic roadway warning data, or a road user protection warning. This process sends control information needed to trigger an update of nearby dynamic message sign such as icy conditions in a remote area or workerspresent in a work zone. This process and its companion process to Provide Other Roadside Inputs support autonomous traffic control without the need for direct control from a Traffic Management Subsystem.

Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS26

ATMS30

Data Flows:

Input Flows

fors_other_roadway_sensor_controls

Output Flows

other_roadway_data_for_highways
other_roadway_data_for_roads
other_roadway_warning_data
tors_other_roadway_sensor_request

User Service Requirements Mapped to Pspec:

USR = C2.6.1.3;
USR = C2.6.1.4;
USR = C2.6.1.5;
USR = C2.7.2.4;

1.2.7.9 Display Roadway Warnings

Description:

Overview: This process shall output data for use by in-vehicle signage equipment on vehicles traveling along the road (surface street) and freeway network served by the Manage Traffic function or directly to drivers via dynamic message signing or highway advisory radio transmission. This data shall be able to provide information from any of the types of indicators that are supported by the function, e.g. intersection controller, pedestrian controller, dynamic warning inputs from traffic management centres or directly from other roadside equipment. Data about incidents and link information such as speed, travel times or roadway conditions may also be used to generate the warning.

Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS26

ATMS29

ATMS30

Data Flows:

Input Flows

other_roadway_warning_data
warning_data_for_highways
warning_data_for_roads

Output Flows

td_roadway_warning
vehicle_roadway_warning_data_output

User Service Requirements Mapped to Pspec:

USR = C2.6.1.2;
USR = C2.6.1.4;

1.2.8.1 Collect Indicator Fault Data

Description:

Overview: This process shall collect data about faults in the operation of indicators (e.g. signals, dms, har) that have been detected by processes in other parts of the Manage Traffic function. It shall be possible for the faults to be detected locally at the indicators, or centrally through communications links with the indicators.

In Subsystem:

Traffic Management

In Market Package:

ATMS01

ATMS03

ATMS04

Data Flows:

Input Flows

indicator_data_fault_for_highways
indicator_data_fault_for_roads
information_device_fault_status
traffic_control_device_status

Output Flows

indicator_new_fault_update

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.2.8.2 Maintain Indicator Fault Data Store

Description:

Overview: This process shall collect data about indicator faults that have been detected by processes in other parts of the Manage Traffic function. It shall be possible for the faults to have been detected locally at the indicators, or centrally through communications links with the indicators. The process shall pass on new fault data to another process for communication to the Construction and Maintenance terminator and shall receive fault clearances from the same process communicating with that terminator. It shall also maintain a store of the current fault state of all indicators. The process shall provide facilities that enable traffic operations personnel to review and update the current fault status of all indicators. Details of faulty and fixed equipment shall be passed by the process to the traffic control strategy selection process so that it can adjust its strategy to take account of the current fault(s).

In Subsystem:

Traffic Management

In Market Package:

ATMS01

ATMS03

ATMS04

Data Flows:

Input Flows

indicator_current_fault_update
indicator_fault_clearance_update
indicator_new_fault_data
indicator_new_fault_update

Input/Output

indicator_current_faults_list

Output Flows

indicator_current_fault_data
indicator_fault_state
indicator_new_fault

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.2.8.3 Provide Indicator Fault Interface for C and M

Description:

Overview: This process shall provide an interface for the exchange of data with the Construction and Maintenance terminator. The interface shall be used to both send data containing details of new indicator equipment faults, and to receive clearances when the faults are cleared. The details of new equipment faults and the clearances shall be received from and sent to another process.

In Subsystem:

Traffic Management

In Market Package:

ATMS01

ATMS03

ATMS04

Data Flows:

Input Flows

fcm_fault_clearance
indicator_new_fault

Output Flows

indicator_fault_clearance_update
tcm_fault_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.2.8.4 Provide Traffic Operations Personnel Indicator Fault Interface

Description:

Overview: This process shall provide the interface through which traffic operations personnel access data about faults on indicator equipment controlled by the Manage Traffic function. The process shall enable the personnel to monitor all indicator equipment faults that have been detected, and if necessary, amend that data. It shall also enable the traffic operations personnel to manually input faults in cases where they cannot otherwise be detected.

In Subsystem:

Traffic Management

In Market Package:

ATMS01

ATMS03

ATMS04

Data Flows:

Input Flows

ftop_indicator_fault_data_input
ftop_indicator_fault_data_request
ftop_indicator_fault_data_update
indicator_current_fault_data

Output Flows

indicator_current_fault_update
indicator_new_fault_data
ttop_current_indicator_faults

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.3.1.1 Analyze Traffic Data for Incidents

Description:

Overview: This process shall analyze traffic sensor data, vehicle probe data, or video images for anomalies that could indicate occurrence of an incident. The data may be collected from roads(surface street) and/or highways served by the Manage Traffic function. The process shall pass on any anomalies that it detects to another process in the Manage Incidents facility as possible detected incidents.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

ATMS29

Data Flows:

Input Flows

current_road_network_use
hri_incident_data
incident_analysis_data
static_data_for_incident_management
traffic_image_data
unusual_data

Output Flows

possible_detected_incidents
reversible_lane_status

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.1;
USR = 1.7.1.2;
USR = 1.7.1.2.1;
USR = 1.7.1.2.1(e);

1.3.1.2 Maintain Static Data for Incident Management

Description:

Overview: This process shall maintain the store of static data (data about the location and features of the road or highway links in the transportation network). This data store is used by another process within the Manage Incidents facility to identify and locate incidents. The static data shall be input to this process from another process within the Planning for Deployment function, and it shall be possible for that process to request a copy of the current static data.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

Data Flows:

Input Flows

supply_incident_static_data

Input/Output

static_data_for_incident_management

Output Flows

current_incident_static_data

User Service Requirements Mapped to Pspec:

USR = NA;

1.3.1.3 Process Traffic Images

Description:

Overview: This process shall process raw traffic image data received from sensors located on the road (surface street) and freeway network served by the Manage Traffic function. The process shall transform the raw data into images that can be sent to another process within the Manage Incidents facility. It shall also act as the control interface through which the images of traffic conditions which are analyzed for incidents can be changed by the traffic operations personnel, who shall also be supplied with images for viewing.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS01

Data Flows:

Input Flows

ft_traffic_images
incident_video_image_control

Output Flows

incident_video_image
reversible_lane_video_images
traffic_image_data

User Service Requirements Mapped to Pspec:

USR = 1.7;
USR = 1.7.1;
USR = 1.7.1.2;
USR = 1.7.1.2.2;
USR = 1.7.1.2.2(a);
USR = 2.0;
USR = 2.2;
USR = 2.2.1;
USR = 2.2.1.2;
USR = 2.2.1.2.1;
USR = 2.2.1.2.1.3;
USR = 2.2.1.2.2;
USR = 2.2.1.2.2.1;
USR = 2.4;
USR = 2.4.2;
USR = 2.4.2.2;

1.3.2.1 Store Possible Incident Data

Description:

Overview: This process shall receive data on possible incidents from other processes within the Manage Incidents facility and other ITS functions. The process shall load all data that it receives into the store of possible incidents. As part of the loading activity, the process shall enter the data into the relevant parts of the standard format for incident data, and shall assign a level of confidence (e.g. related to the source of the data or time of its detection) to that data.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

ATMS18

Data Flows:

Input Flows

fep_event_information
fws_current_weather
fws_predicted_weather
logged_special_vehicle_route
maintenance_incident_information
maintenance_vehicle_status
media_incident_data_updates
pollution_incident
possible_detected_incidents

Output Flows

possible_incident_data_update
possible_incidents

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.1;
USR = 1.7.1.1;
USR = 1.7.1.1.1;
USR = 1.7.1.1.2;
USR = 1.7.1.1.2(a);
USR = 1.7.1.1.2(b);
USR = 1.7.1.1.2(c);
USR = 1.7.1.1.2(d);
USR = 1.7.1.1.2(e);
USR = 1.7.1.1.3;
USR = 1.7.1.2;
USR = 1.7.1.2.1;
USR = 1.7.1.2.1(e);
USR = 1.7.1.2.1(g);
USR = 1.7.1.2.2;
USR = 1.7.1.2.3;
USR = 1.7.2;
USR = 1.7.2.1;

1.3.2.2 Review and Classify Possible Incidents

Description:

Overview: This process shall review input data about possible incidents and provide verification of the incident. The process shall have the capability of using algorithms to automatically identify and verify an incident. The process shall have the capability to classify an incident as current incident or a planned event and shall be load the data into the store of possible incidents as either current incidents or planned events. The process shall report any incidents that it is unable to verify or classify to the traffic operations personnel for manual verification and classification. The process shall allow the traffic operations personnel to request all possible incidents and carry out the verification and classification process manually.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

ATMS18

Data Flows:

Input Flows

incident_details
operations_incident_data_updates
possible_incident_data_update
possible_incidents
request_possible_incidents_data
work_zone_status

Output Flows

current_incidents_new_data
incident_data_update
incident_details_request
maintenance_incident_coordination
planned_event_data
planned_events
planned_events_new_data
possible_incidents_data_output
tep_event_confirmation

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.1;
USR = 1.7.1.1;
USR = 1.7.1.1.1;
USR = 1.7.1.1.1(a);
USR = 1.7.1.1.1(b);
USR = 1.7.1.1.1(c);
USR = 1.7.1.1.1(d);
USR = 1.7.1.1.1(e);
USR = 1.7.1.1.1(f);
USR = 1.7.1.1.1(g);
USR = 1.7.1.1.1(h);
USR = 1.7.1.2;
USR = 1.7.1.2.1;
USR = 1.7.1.2.1(a);
USR = 1.7.1.2.1(b);
USR = 1.7.1.2.1(c);
USR = 1.7.1.2.1(d);

USR = 1.7.1.2.1(e);
USR = 1.7.1.2.1(f);
USR = 1.7.1.2.1(g);
USR = 1.7.1.2.2;
USR = 1.7.1.2.2(a);
USR = 1.7.1.2.2(b);
USR = 1.7.1.2.2(c);

1.3.2.3 Review and Classify Planned Events

Description:

Overview: This process shall receive updates of planned events and review the complete list of them to determine when an incident should be reclassified from planned event to current incident. It shall carry out the re-classification process automatically either upon receiving notice that the store of planned events has been updated, or at some periodic rate. The criteria for reclassifying an incident could be that the planned start time of the event has passed. The process shall request details of planned events from the process that manages their data store and shall send details of any new (re-classified) current incidents to the process that manages their data store. It shall also provide updates of planned events and current incidents to other ITS functions, and details of any new planned events to the process responsible for the output of data to vehicle signage functions.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

ATMS08

CVO10

Data Flows:

Input Flows

current_incidents_data
current_incidents_request
incident_data_update
incident_response_status
planned_events_data
reclassify_incidents

Output Flows

current_incident_data
current_incidents
current_incidents_data_output
current_incidents_data_request
current_incidents_data_update
planned_event_data_for_vehicle_signage
request_planned_events_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.1;
USR = 1.7.1.2;
USR = 1.7.1.2.2;
USR = 1.7.1.2.2(a);
USR = 1.7.1.2.2(b);
USR = 1.7.4;

1.3.2.4 Provide Planned Events Store Interface

Description:

Overview: This process shall provide the interface to, and manage the use of the store containing details of planned events. The process shall enter details of all new planned events into the store, retrieve details on request, and delete details of an incident when it has been re-classified as a current incident.. The process shall be able to receive details of planned events from within the local Manage Incidents facility, and from similar facilities in other Traffic Management Subsystems (TMS's). When requested, the process shall also be able to provide details of its planned events to the Manage Incidents facilities in other TMS's.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

Data Flows:

Input Flows

other_planned_events
planned_events_new_data
request_local_planned_events_data
request_planned_events_data

Input/Output

planned_events_store

Output Flows

planned_events_data
planned_events_data_output
planned_events_local_data
request_other_planned_events_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.3.2.5 Provide Current Incidents Store Interface

Description:

Overview: This process shall provide the interface to, and manage the use of the store of current incident details. The process shall enter the details of all new current incidents into the store, retrieve details on request, and delete details of incidents when they cease to be current. The process shall be able to receive details of current incidents from within the local Manage Incidents facility, and from similar facilities in other Traffic Management Subsystems (TMS's). When requested, the process shall also be able to provide details of its current incidents to the Manage Incidents facilities in other TMS's.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

Data Flows:

Input Flows

current_incidents_data_request
current_incidents_data_update
current_incidents_new_data
other_current_incidents
request_local_current_incidents_data

Input/Output

current_incidents_store

Output Flows

current_incidents_data
request_other_current_incidents_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.3.3 Respond to Current Incidents

Description:

Overview: This process shall provide responses to incidents that become current, i.e. active. Three general strategies for response to incidents can be supported by the process: 1) Operator enters response (there is no set of predetermined responses), 2) the operator selects response from a set of predetermined responses (possibly modifying the response), and 3) the process automatically accesses and implements a response from a set of predetermined responses (while informing the operator of the actions taken). Where predetermined responses are utilized, the operator shall have the capability to view, modify, or override the predetermined response. The predetermined response to each type of incident shall be defined for the process in the store defined_responses_data. If the process cannot find a predetermined response for a particular incident, it shall send the details of the incident to the traffic operations personnel so that they can provide an update to the store of predetermined responses. The process shall output the predetermined responses to an incident when it receives notification from another process in the Manage Incidents facility that a new current incident has occurred. At the same time it shall also output the incident data to the process responsible for providing broadcast data to roadside processes. The other process in the Manage Incidents facility shall also provide details of incidents that have ceased to be current (terminated) so that this process can send out data to clear the actions requested and roadside broadcast information output in response to its occurrence.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

CVO10

Data Flows:

Input Flows

current_incidents_data_output
defined_responses_data

Output Flows

current_incident_data_for_vehicle_signage
cv_incident_override
dms_updates_for_highways
dms_updates_for_roads
hri_strategy_override
incident_alert
incident_response_clear
incident_response_log
incident_strategy_override
undefined_incident_response

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.1;
USR = 1.7.1.2;
USR = 1.7.1.2.3;
USR = 1.7.2;
USR = 1.7.2.2;
USR = 1.7.2.3;
USR = 1.7.2.4;
USR = 1.7.2.5;
USR = 1.7.3;
USR = 1.7.3.1;
USR = 1.7.3.1(a);
USR = 1.7.3.2;
USR = 1.7.3.3;

1.3.4.1 Retrieve Incident Data

Description:

Overview: This process shall retrieve incident data from the stores of planned events and current incidents that are managed by other processes in the Manage Incidents facility of the Manage Traffic function. The process shall retrieve data as the result of a request which may come from the traffic operations personnel or the media operator. The output shall be returned to the source of the request, except where the media operator has specified that the data should be output to the media system.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

CVO10

Data Flows:

Input Flows

current_incidents
map_data_for_incident_display
planned_events_data_output
possible_incidents_data_output
request_incident_media_data
request_incident_operations_data

Output Flows

current_incidents_request
request_possible_incidents_data
retrieved_incident_media_data
retrieved_incident_operations_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.1;
USR = 1.7.1.2;
USR = 1.7.1.2.1;
USR = 1.7.1.2.1(c);
USR = 1.7.1.2.1(d);
USR = 1.7.1.2.1(f);
USR = 1.7.1.2.2;
USR = 1.7.1.2.2(d);
USR = 1.7.4;

1.3.4.2 Provide Traffic Operations Personnel Incident Data Interface

Description:

Overview: This process shall provide the interface between the traffic operations personnel and the Manage Incidents facility of the Manage Traffic function. It shall enable the personnel to request and amend details of predicted and current incidents and predetermined incident responses, obtain and control incident video image data and manually re-classify incidents as possible or current or a planned event. It shall also output to the traffic operations personnel incident details to which no predetermined response currently exists. The process shall support inputs from and outputs to the traffic operations personnel Where appropriate and/or requested by the traffic operations personnel, the process shall provide the output 'display' in a form incorporating a map of the relevant part(s) of the surface street and freeway network served by the function. The process shall obtain the map from a local data store, which it shall request to be updated by another process as and when required.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

ATMS08

ATMS17

ATMS18

CVO10

Data Flows:

Input Flows

defined_incident_response_data
ftop_defined_incident_response_data_request
ftop_defined_incident_response_data_update
ftop_incident_camera_action_request
ftop_incident_data_amendment
ftop_incident_information_requests
ftop_output_possible_defined_responses
ftop_request_possible_incidents_data
ftop_resource_request
ftop_update_defined_incident_responses
incident_video_image
possible_defined_responses_output
remote_video_image_control
retrieved_incident_operations_data
traffic_operations_resource_response
undefined_incident_response
video_camera_control_strategy
wrong_way_vehicle_detection

Input/Output

operator_log_for_incidents_data

Output Flows

defined_incident_response_data_request
defined_incident_response_update_request
defined_incident_response_updates
incident_video_image_control
operations_incident_data_updates
possible_defined_responses_output_request
reclassify_incidents
request_incident_map_display_update
request_incident_operations_data
traffic_operations_resource_request

ttop_defined_incident_responses_data
ttop_incident_information_display
ttop_incident_video_image_output
ttop_possible_defined_response_output
ttop_possible_incidents_data
ttop_resource_response
ttop_undefined_response_details
ttop_wrong_way_detection

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.3.4.3 Provide Media Incident Data Interface

Description:

Overview: This process shall provide the interface between the Media and the Manage Incidents facility. It shall enable the media to request details of incidents and shall allow transmission of incident information to the media. The media shall also provide raw input data on possible incidents. The process shall enable the output to incorporate a map of the area to which the incidents relate.

In Subsystem:

Traffic Management

In Market Package:

ATMS06

ATMS08

Data Flows:

Input Flows

fm_incident_data_request
fm_incident_information
retrieved_incident_media_data

Output Flows

media_incident_data_updates
request_incident_media_data
tm_incident_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.1;
USR = 1.7.1.1;
USR = 1.7.1.1.1;
USR = 1.7.1.1.1(d);
USR = 1.7.4;

1.3.4.4 Update Incident Display Map Data

Description:

Overview: This process shall provide updates to the store of digitized map data used with displays of incident data produced by processes in the Manage Incidents facility of the Manage Traffic function. The process shall obtain the new data from a map provider or other appropriate data source, on receiving an update request from the traffic operations personnel interface process within the Manage Incidents facility.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

CVO10

Data Flows:

Input Flows

fmup_incident_display_update
request_incident_map_display_update

Output Flows

map_data_for_incident_display
tmup_request_incident_display_update

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.3.4.5 Manage Resources for Incidents

Description:

Overview: This process shall provide the capability for the Manage Traffic function to generate and receive requests for resources in responding to incidents. The process shall provide the capability for traffic operations personnel to request resources from the Construction and Maintenance to provide equipment and support for incident response and clean up. The process shall be able to receive resource requests from the Manage Emergency function and respond with the status of the response by Construction and Maintenance or the traffic operations personnel.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

ATMS18

CVO10

Data Flows:

Input Flows

maintenance_resource_response
resource_request
traffic_operations_resource_request

Output Flows

maintenance_resource_request
operator_log_for_incidents_data
resource_deployment_status
traffic_operations_resource_response

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.1;
USR = 1.7.1.1;
USR = 1.7.1.1.1;
USR = 1.7.1.1.1(c);
USR = 1.7.1.2;
USR = 1.7.1.2.2;
USR = 1.7.1.2.2(e);
USR = 1.7.3;
USR = 1.7.3.1;
USR = 1.7.3.1(b);

1.3.5 Manage Possible Predetermined Responses Store

Description:

Overview: This process shall manage the data store containing possible predetermined responses to incidents used within the Manage Incidents facility. These responses shall be those that another process within the facility has found to be worth including in the store of predetermined responses from an analysis of the incident response log. This process shall enable retrieval of the data from the store for presentation to traffic operations personnel and its possible transfer to the process that manages the store of predetermined incident responses that are actually used by other processes in the Manage Incidents facility.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

CVO10

Data Flows:

Input Flows

defined_incident_response_update_request
possible_defined_responses_data
possible_defined_responses_output_request

Input/Output

possible_defined_responses

Output Flows

defined_incident_response_changes
possible_defined_responses_output

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.3.6 Manage Predetermined Incident Response Data

Description:

Overview: This process shall manage data held in the store of predetermined incident responses that are used by processes within the Manage Incidents facility of the Manage Traffic function. The process shall provide details of the current predetermined responses in response to requests from traffic operations personnel, and shall also update the store with new responses received from the process that manages the store of possible predetermined responses.

In Subsystem:

Traffic Management

In Market Package:

ATMS08

CVO10

Data Flows:

Input Flows

defined_incident_response_changes
defined_incident_response_data_request
defined_incident_response_updates

Input/Output

defined_responses_data

Output Flows

defined_incident_response_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.3.7 Analyze Incident Response Log

Description:

In Subsystem:

Traffic Management

In Market Package:

ATMS08

CVO10

Data Flows:

Input Flows

incident_response_log

Output Flows

possible_defined_responses_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7;
USR = 1.7.0;
USR = 1.7.4;

1.4.1 Provide Traffic Operations Personnel Demand Interface

Description:

Overview: This process shall provide the interface between the traffic operations personnel and the processes and data stores used within the Manage Demand facility of the Manage Traffic function. It shall enable the traffic operations personnel to access the data used as input by the demand forecasting process and the results of that process, to request that the input data be updated, set the policies used as input to the Calculate Forecast Demand process, to request that the demand forecasting process runs, and to run the process that implements the results. Where appropriate and/or requested by the traffic operations personnel, the process shall provide the output in a form that includes a map of the relevant part(s) of the road and freeway network served by the Manage Travel Demand function. The process shall obtain the map from a local data store, which it shall request to be updated by another process when required.

In Subsystem:

Traffic Management

In Market Package:

ATMS09

Data Flows:

Input Flows

demand_forecast_data
demand_forecast_result
demand_input_data
demand_management_result
ftop_demand_data_request
ftop_demand_data_update_request
ftop_demand_forecast_request
ftop_demand_policy_activation
ftop_demand_policy_information_request
ftop_demand_policy_updates
map_data_for_demand_display

Input/Output

demand_policy_data

Output Flows

demand_data_update_request
demand_forecast_request
demand_management_activate
request_demand_display_update
ttop_demand_data
ttop_demand_forecast_data
ttop_demand_forecast_result
ttop_demand_policy_activation_result
ttop_demand_policy_information

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.8;
USR = 1.8.0;
USR = 1.8.1;
USR = 1.8.1.1;

1.4.2 Collect Demand Forecast Data

Description:

Overview: This process shall collect data from other ITS functions for use as input to the demand forecasting process within the Manage Demand facility of the Manage Traffic function. The process shall support data retrieval from other functions on request from the traffic operations personnel and through the receipt of unsolicited data from ITS functions. It shall load all the data that it receives in a consistent format into the input store used by the demand forecasting process.

In Subsystem:

Traffic Management

In Market Package:

APTS7

ATMS09

Data Flows:

Input Flows

current_other_routes_use
current_transit_routes_use
demand_data_update_request
fws_current_weather
fws_predicted_weather
hri_status_for_traffic_demand
parking_lot_charge_details
parking_lot_charge_direct_details
pollution_state_data
toll_price_details
toll_price_direct_details
traffic_data_for_demand
transit_fare_details
transit_fare_direct_details
transit_running_data_for_demand
transit_services_for_demand
unusual_congestion
weather_service_information_request

Output Flows

demand_input_data
parking_lot_charge_direct_request
parking_lot_charge_request
pollution_state_data_request
toll_price_direct_request
toll_price_request
traffic_data_demand_request
transit_conditions_demand_request
transit_fare_direct_request
transit_fare_request
transit_services_demand_request
weather_service_information

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.8;
USR = 1.8.0;
USR = 1.8.1;
USR = 1.8.1.1;
USR = 1.8.1.2;
USR = 1.8.1.2(e);
USR = 1.8.1.2(f);
USR = 1.8.1.3;
USR = 1.8.1.3(e);
USR = 1.8.1.3(f);
USR = 1.8.1.4;
USR = 1.8.1.5;
USR = 1.8.1.5(a);
USR = 1.8.1.5(c);
USR = 1.8.2;
USR = 1.8.2.1;
USR = 1.8.2.1(a);
USR = 1.8.2.10;
USR = 1.8.2.11;
USR = 1.8.2.12;
USR = 1.8.2.13;
USR = 1.8.2.14;
USR = 1.8.2.2;
USR = 1.8.2.3;
USR = 1.8.2.4;
USR = 1.8.2.4(f);
USR = 1.8.2.5;
USR = 1.8.2.5(a);
USR = 1.8.2.5(b);
USR = 1.8.2.5(c);
USR = 1.8.2.5(d);
USR = 1.8.2.5(e);
USR = 1.8.2.6;
USR = 1.8.2.7;
USR = 1.8.2.7(a);
USR = 1.8.2.7(b);
USR = 1.8.2.7(c);
USR = 1.8.2.8;
USR = 1.8.2.8(a);
USR = 1.8.2.9;
USR = 1.8.2.9(a);
USR = 1.8.2.9(b);
USR = 1.8.2.9(c);
USR = 1.8.3;
USR = 3.0;
USR = 3.1;
USR = 3.1.5;
USR = 3.1.5.1;
USR = 3.1.5.1.1;

1.4.3 Update Demand Display Map Data

Description:

Overview: This process shall provide updates to the store of map data used for displays of forecast traffic and travel demand produced by processes in the Manage Travel Demand facility of the Manage Traffic function. The process shall obtain the new data from a specialist map data supplier or some other appropriate source, on receiving an update request from the traffic operations personnel interface process within the Manage Travel Demand facility.

In Subsystem:

Traffic Management

In Market Package:

ATMS09

Data Flows:

Input Flows

fmup_demand_display_update
request_demand_display_update

Output Flows

map_data_for_demand_display
tmup_request_demand_display_update

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.1;

1.4.4 Implement Demand Management Policy

Description:

Overview: This process shall implement the traffic and travel demand forecast data produced by the demand forecasting process in the Manage Travel Demand facility of the Manage Traffic function. The new demand forecast data shall be implemented in such a way that it can influence the demand from travelers for various types of services provided by ITS functions. The process shall when required, request changes to transit services, and/or the charges for tolls, and/or the use of parking lot spaces (as per the locally determined demand policy). It shall communicate the results of its policy implementation to the process that provides the interface to the traffic operations personnel.

In Subsystem:

Traffic Management

In Market Package:

ATMS09

ATMS09

Data Flows:

Input Flows

demand_forecast_data
demand_management_activate
parking_lot_charge_change_response
toll_price_changes_response
transit_services_changes_response

Output Flows

ahs_control_data
demand_management_result
demand_overrides
parking_lot_charge_change_request
toll_price_changes_request
transit_services_changes_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.8;
USR = 1.8.0;
USR = 1.8.1;
USR = 1.8.1.1;
USR = 1.8.1.2;
USR = 1.8.1.2(e);
USR = 1.8.1.2(f);
USR = 1.8.1.3;
USR = 1.8.1.3(e);
USR = 1.8.1.3(f);
USR = 1.8.1.4;
USR = 1.8.1.5;
USR = 1.8.1.5(a);
USR = 1.8.1.5(b);
USR = 1.8.1.6;
USR = 1.8.1.6(d);
USR = 1.8.2;
USR = 1.8.2.1;
USR = 1.8.2.4;
USR = 1.8.2.4(f);
USR = 1.8.2.10;
USR = 1.8.2.11;
USR = 1.8.2.12;
USR = 1.8.2.13;

USR = 1.8.2.14;
USR = 1.8.2.14(a);
USR = 1.8.2.14(b);
USR = 1.8.2.14(c);
USR = 1.8.2.2;
USR = 1.8.2.3;
USR = 1.8.2.3(a);
USR = 1.8.2.3(b);
USR = 1.8.2.3(c);
USR = 1.8.2.3(d);
USR = 1.8.2.4;
USR = 1.8.2.5;
USR = 1.8.2.5(a);
USR = 1.8.2.5(b);
USR = 1.8.2.5(c);
USR = 1.8.2.5(d);
USR = 1.8.2.5(e);
USR = 1.8.2.6;
USR = 1.8.2.7;
USR = 1.8.2.7(a);
USR = 1.8.2.7(b);
USR = 1.8.2.7(c);
USR = 1.8.2.8;
USR = 1.8.2.8(a);
USR = 1.8.2.9;
USR = 1.8.2.9(a);
USR = 1.8.2.9(b);
USR = 1.8.2.9(c);
USR = 1.8.3;
USR = 3.0;
USR = 3.1;
USR = 3.1.5;
USR = 3.1.5.1;
USR = 3.1.5.1.1;
USR = 3.1.5.2;

1.4.5 Calculate Forecast Demand

Description:

Overview: This process shall provide a forecast of traffic and travel demand in the geographic area served by the Manage Traffic function to which this instance of the Manage Travel Demand facility belongs. The process shall base its forecast on the current and predicted traffic levels traveler demand patterns obtained from an analysis of data obtained from elsewhere within the Manage Traffic function and from other ITS functions as well as locally determined demand policy. The process shall produce a demand forecast that changes the way that services are provided by ITS functions according to locally determined demand policy.

In Subsystem:

Traffic Management

In Market Package:

ATMS09

Data Flows:

Input Flows

demand_forecast_request
demand_input_data
demand_policy_data

Output Flows

demand_forecast_data
demand_forecast_result

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.8;
USR = 1.8.0;
USR = 1.8.1;
USR = 1.8.1.1;
USR = 1.8.1.2;
USR = 1.8.1.3;
USR = 1.8.1.4;
USR = 1.8.1.5;
USR = 1.8.1.5(b);
USR = 1.8.2;
USR = 1.8.2.1;
USR = 1.8.2.1(f);
USR = 1.8.2.10;
USR = 1.8.2.11;
USR = 1.8.2.12;
USR = 1.8.2.13;
USR = 1.8.2.14;
USR = 1.8.2.2;
USR = 1.8.2.3;
USR = 1.8.2.3(a);
USR = 1.8.2.4;
USR = 1.8.2.5;
USR = 1.8.2.5(a);
USR = 1.8.2.5(b);
USR = 1.8.2.5(c);
USR = 1.8.2.5(d);
USR = 1.8.2.5(e);
USR = 1.8.2.6;
USR = 1.8.2.7;
USR = 1.8.2.7(a);
USR = 1.8.2.7(b);

USR = 1.8.2.7(c);
USR = 1.8.2.8;
USR = 1.8.2.8(a);
USR = 1.8.2.9;
USR = 1.8.2.9(a);
USR = 1.8.2.9(b);
USR = 1.8.2.9(c);
USR = 1.8.3;
USR = 3.0;
USR = 3.1;
USR = 3.1.5;
USR = 3.1.5.1;
USR = 3.1.5.1.1;
USR = 3.1.5.2;

1.5.1 Provide Traffic Operations Personnel Pollution Data Interface

Description:

Overview: This process shall provide the interface between the traffic operations personnel and the processes and data stores used within the Manage Emissions facility of the Manage Traffic function. The process shall enable the personnel to access and update the pollution reference data used by other processes within the facility, and to access the pollution state data provided by those processes. The process shall support inputs from the traffic operations personnel. Where appropriate and/or requested by the traffic operations personnel, the process shall incorporate map data of the relevant part(s) of the surface street and freeway network served by the Manage Traffic function. The process shall obtain the map from a local data store, which it shall request to be updated by another process as and when required.

In Subsystem:

Environment Management

In Market Package:

ATMS11

Data Flows:

Input Flows

ftop_pollution_data_information_request
ftop_pollution_parameter_updates
map_data_for_pollution_display
pollution_reference_data_output
pollution_state_data_output

Output Flows

pollution_reference_data_request
pollution_reference_data_update
pollution_state_data_output_request
request_pollution_map_display_update
ttop_pollution_data_display

User Service Requirements Mapped to Pspec:

USR = 1.8.2;
USR = 1.8.2.2;
USR = 1.8.2.2(a);
USR = 1.8.2.2(c);
USR = 1.9.0;
USR = 1.9.1;
USR = 1.9.1.2;
USR = 1.9.1.2.1;
USR = 1.9.1.2.2;
USR = 1.9.2.2.1;
USR = 1.9.2.2.3;

1.5.2 Process Pollution Data

Description:

Overview: This process shall process the pollution data being collected from sensors in the geographic area being served by the Manage Traffic function. The process shall integrate data from distributed roadside sensors (provided by another process) with that obtained directly from sensors looking at the general (wide area) environment. The data shall be checked by the process against the pollution levels that have been set up as reference points. If the process finds that the detected levels of pollution exceed the reference levels it shall generate pollution warnings. The process shall send these warnings to other processes in the Manage Traffic function for output to drivers and travelers.

In Subsystem:

Environment Management

In Market Package:

ATMS11

Data Flows:

Input Flows

fe_area_pollutant_levels
pollution_state_roadside_collection
pollution_state_static_acceptance_criteria

Output Flows

archive_pollution_data
current_traffic_pollution_data
pollution_incident
pollution_state_static_collection
pollution_state_static_log_data
tm_pollution_data
wide_area_pollution_data

User Service Requirements Mapped to Pspec:

USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.4;
USR = 1.8.1.4(d);
USR = 1.9.0;
USR = 1.9.1;
USR = 1.9.1.1;
USR = 1.9.1.1.1;
USR = 1.9.1.1.2;
USR = 1.9.1.1.3;
USR = 1.9.1.2;
USR = 1.9.1.2.1;
USR = 1.9.1.2.2;

1.5.3 Update Pollution Display Map Data

Description:

Overview: This process shall provide updates to the map data used in displays of pollution data produced by processes in the Manage Emissions facility of the Manage Traffic function. The process shall obtain the map data from a specialist map data supplier or some other appropriate data source, on receiving an update request from the traffic operations personnel interface process within the Manage Emissions facility.

In Subsystem:

Environment Management

In Market Package:

ATMS11

Data Flows:

Input Flows

fmup_pollution_display_update
request_pollution_map_display_update

Output Flows

map_data_for_pollution_display
tmup_request_pollution_display_update

User Service Requirements Mapped to Pspec:

USR = 1.9.0;
USR = 1.9.1;
USR = 1.9.1.2.1;
USR = 1.9.1.2.2;
USR = 1.9.2.2.1;
USR = 1.9.2.2.3;

1.5.4 Manage Pollution State Data Store

Description:

Overview: This process shall manage the store of pollution state data in the Manage Emissions facility of the Manage Traffic function. The data in the store shall be that which has been received by the process from other processes within the facility. The process shall manage the data in the store to enable its contents to be available to other processes within the Manage Traffic function, and to traffic operations personnel, via an interface process within the Manage Emissions facility.

In Subsystem:

Environment Management

In Market Package:

ATMS11

Data Flows:

Input Flows

pollution_state_data_output_request
pollution_state_data_request
pollution_state_static_collection
pollution_state_vehicle_collection

Input/Output

pollution_state

Output Flows

archive_pollution_state_data
pollution_state_data
pollution_state_data_output

User Service Requirements Mapped to Pspec:

USR = 1.8.2;
USR = 1.8.2.2;
USR = 1.8.2.2(b);
USR = 1.9.0;
USR = 1.9.1;
USR = 1.9.1.2;
USR = 1.9.1.2.1;
USR = 1.9.1.2.2;
USR = 1.9.2.2.1;
USR = 1.9.2.2.3;

1.5.5 Process Vehicle Pollution Data

Description:

Overview: This process shall obtain pollution data about individual vehicles and analyze it against reference data obtained from another process within the Manage Emissions facility of the Manage Traffic function. The process shall use this reference data to determine whether or not a vehicle is possibly violating the acceptable levels of pollution output. When the process determines that a possible violation has occurred, it shall send the detected pollution levels and the vehicle identity to the process responsible for law enforcement in the Manage Emergency Services function for action.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS11

Data Flows:

Input Flows

From_Vehicle_Characteristics
ft_vehicle_pollutant_levels
pollution_state_vehicle_acceptance_criteria
vehicle_status_details_for_emissions

Output Flows

pollution_state_vehicle_collection
pollution_state_vehicle_log_data
vehicle_pollution_alert
vehicle_pollution_message_for_highways
vehicle_pollution_message_for_roads

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.4;
USR = 1.8.1.4(b);
USR = 1.8.2;
USR = 1.8.2.13;
USR = 1.8.2.13(b);
USR = 1.8.3;
USR = 1.8.3.1;
USR = 1.8.3.1(d);
USR = 1.9;
USR = 1.9.0;
USR = 1.9.2;
USR = 1.9.2.1;
USR = 1.9.2.1.1;
USR = 1.9.2.1.2;
USR = 1.9.2.1.4;
USR = 1.9.2.1.5;
USR = 1.9.2.2;
USR = 1.9.2.2.1;
USR = 1.9.2.2.2;
USR = 1.9.2.2.3;

1.5.6 Detect Roadside Pollution Levels

Description:

Overview: This process shall process the local area pollution data analyzed by sensors looking at the levels of pollution at the roadside within the geographic area served by the Manage Traffic function. The process shall pass the data on to another process within the Manage Emissions facility for integration with wide area pollution data and comparison with thresholds for pollution incidents.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS11

Data Flows:

Input Flows

fe_roadside_pollutant_levels

Output Flows

pollution_state_roadside_collection

User Service Requirements Mapped to Pspec:

USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.4;
USR = 1.8.1.4(d);
USR = 1.9.0;
USR = 1.9.1;
USR = 1.9.1.1;
USR = 1.9.1.1.1;
USR = 1.9.1.1.2;
USR = 1.9.1.1.3;
USR = 1.9.1.2;
USR = 1.9.1.2.1;
USR = 1.9.1.2.2;

1.5.7 Manage Pollution Data Log

Description:

Overview: This process shall manage the log of pollution data within the Manage Emissions facility of the Manage Traffic function. The process shall receive data for entry into the log from other processes within the facility. It shall also send the contents of the log to the Manage Archive Data function for use in planning future modifications to the ITS network.

In Subsystem:

Environment Management

In Market Package:

ATMS11

Data Flows:

Input Flows

pollution_state_static_log_data
pollution_state_vehicle_log_data

Input/Output

pollution_data_log

Output Flows

pollution_archive_data_log

User Service Requirements Mapped to Pspec:

USR = NA;

1.5.8 Manage Pollution Reference Data Store

Description:

Overview: This process shall manage the store of pollution reference data within the Manage Emissions facility of the Manage Traffic function. It shall make the contents of the store available to other processes within the facility that are responsible for emissions management, and on request to the traffic operations personnel interface process. The process shall accept updates to the stored data from the traffic operations personnel interface process.

In Subsystem:

Environment Management

In Market Package:

ATMS11

Data Flows:

Input Flows

pollution_reference_data_archive_request
pollution_reference_data_request
pollution_reference_data_update

Input/Output

pollution_reference_data

Output Flows

archive_pollution_reference_data
pollution_reference_data_output
pollution_state_static_acceptance_criteria
pollution_state_vehicle_acceptance_criteria

User Service Requirements Mapped to Pspec:

USR = 1.9.0;
USR = 1.9.1;
USR = 1.9.1.2;
USR = 1.9.1.2.1;
USR = 1.9.1.2.2;
USR = 1.9.2.2.1;
USR = 1.9.2.2.3;

1.5.9 Manage Pollution Archive Data

Description:

Overview: This process shall collect and store the pollution data being collected from sensors in the geographic area being served by the Manage Traffic function. The process shall integrate data from distributed roadside sensors (provided by another process) with that obtained directly from sensors looking at the general (wide area) environment.

In Subsystem:

Environment Management

In Market Package:

AD1

AD2

Data Flows:

Input Flows

archive_pollution_data
archive_pollution_reference_data
archive_pollution_state_data
emissions_archive_request
emissions_archive_status
pollution_archive_data_log

Input/Output

emissions_data_archive

Output Flows

emissions_archive_data
pollution_reference_data_archive_request

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.7;
USR = 7.1.3.1.7(a);

1.6.1.1 Detect Roadway Events

Description:

Overview: This process is responsible for monitoring local sensor data obtained from traffic surveillance and then determining and reporting the current state of all traffic in the HRI vicinity. The process provides triggers for other processes within Manage HRI Traffic Volume. It also monitors the device controls as they are initiated by the Activate HRI Device Controls process.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS14

Data Flows:

Input Flows

approaching_train_data
device_control_state
hri_device_sense
hri_status
hri_traffic_surveillance
indicator_sign_control_data_for_hri

Output Flows

current_hri_state
event_notice
roadway_status
train_sense_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3;
USR = 1.3.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.2.1(d);
USR = 1.3.1.2.1(d).1;
USR = 1.10;
USR = 1.10.0;
USR = 1.10.1;
USR = 1.10.1.7;
USR = 1.10.2;
USR = 1.10.2.1;
USR = 1.10.2.1.1;
USR = 1.10.3;
USR = 1.10.3.3;
USR = 1.10.3.3.3;

1.6.1.2.1 Control HRI Traffic Signals

Description:

Overview: This process is responsible for interpreting the hri_control message and safely directing the activation of the appropriate devices. This process will both directly command devices at the HRI and will disseminate necessary control information to the Process Indicator Output Data for Roads function to allow integrated control of adjacent traffic signals. Data will also be sent to SSR and/or HSR Device Control functions to control these specialized devices at the crossing. When sensor data indicates an approaching train this process notifies the Process Indicator Output Data for Roads function to allow the signal timing to be adjusted and dynamic message signs, if available, to be updated. This allows the traffic signals in the area adjacent to an HRI to be used to clear the Storage Area in advance of an approaching train and to manage traffic around the intersection.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS13

Data Flows:

Input Flows

hri_control_message

Output Flows

barrier_control_request
hsr_control_request
ssr_control_request
traffic_device_control
traffic_device_control_state

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.1;
USR = 1.10.1.7;
USR = 1.10.3;
USR = 1.10.3.1;
USR = 1.10.3.2;
USR = 1.10.3.3;
USR = 1.10.3.3.1;

1.6.1.2.2 Control HRI Warnings and Barriers

Description:

Overview: This process is responsible for initiating the activation of HRI barriers at active vehicular and pedestrian grade crossings. When a request is sent to activate the HRI barriers perhaps because of a detection of an oncoming train, this process sends the device control signal to the Manage Device Controls process to activate the barriers. This process also returns state information to the Maintain Device State process concerning the commands that have been initiated by this process. Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS13

Data Flows:

Input Flows

barrier_control_request

Output Flows

barrier_device_control
barrier_device_control_state

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.3;
USR = 1.10.3.3;
USR = 1.10.3.3.2;

1.6.1.2.3 Provide SSR Device Controls

Description:

Overview: This process is responsible for initiating the activation of HRI Standard Speed Rail control devices at active vehicular and pedestrian grade crossings. This process responds to requests sent by the Control HRI Traffic Signals process based on detection of an oncoming train. This process sends command information to the Manage Device Control containing control signals and commands that are unique to the SSR functions. State information is also sent to the Maintain Device State process to monitor the last known state of the controls commands being processed. Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS13

Data Flows:

Input Flows

ssr_control_request

Output Flows

ssr_device_control
ssr_device_control_state

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.4;
USR = 1.10.4.1;

1.6.1.2.4 Provide HSR Device Controls

Description:

Overview: This process is responsible for initiating the activation of HRI devices, barriers and other special safety features for High Speed Rail at active vehicular and pedestrian grade crossings. This process responds to requests sent by the Control HRI Traffic Signals process based on detection of an oncoming train. This process sends command information to the Manage Device Control containing control signals and commands that are unique to the HSR functions, such as trapped vehicle detection. State information is also sent to the Maintain Device State process to monitor the last known state of the controls commands being processed.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS14

Data Flows:

Input Flows

hsr_control_request

Output Flows

hsr_device_control
hsr_device_control_state

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.5;
USR = 1.10.5.1;
USR = 1.10.5.2;
USR = 1.10.5.2.2;

1.6.1.2.5 Manage Device Control

Description:

Overview: This process is responsible for managing and selecting the appropriate device control messages. This process gathers the control signals from the other Activate HRI Device Control processes and forwards them as needed to the Process Indicator Output Data for Roads process within Provide Device Control. These control signals are used to activate all of the HRI unique roadside devices such as gates or other barriers, lights, adjacent traffic signals, message signs or in-vehicle signage beacons.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS13

Data Flows:

Input Flows

barrier_device_control
hsr_device_control
ssr_device_control
traffic_device_control

Output Flows

hri_device_control

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.4;
USR = 1.10.4.1;

1.6.1.2.6 Maintain Device State

Description:

Overview: This process is responsible for managing and selecting the appropriate device control state messages. This process collects the device state messages that are produced by the other Activate HRI Device Controls processes and forwards the appropriate signals to the Detect Roadway Events process that monitors the status of the HRI commands being processed. This information is also used in the equipment diagnostic monitoring and testing.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS13

Data Flows:

Input Flows

barrier_device_control_state
hsr_device_control_state
ssr_device_control_state
traffic_device_control_state

Output Flows

device_control_state

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.4;
USR = 1.10.4.1;

1.6.1.3 Perform Equipment Self-Test

Description:

Overview: This process is responsible for performing real-time equipment checks and reporting the status of the equipment associated with an active grade crossing. Based on receipt of the sensor data of the surrounding highway and rail traffic and receipt of any near term events this process can execute a real-time check of the equipment and determine the relative health and status of the active grade crossing equipment. The output is sent onto the Monitor HRI Status process for further processing with other diagnostic data.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS13

Data Flows:

Input Flows

hri_device_sense
near_term_status

Output Flows

device_status

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.3;
USR = 1.10.3.3;
USR = 1.10.3.3.4;

1.6.1.4.1 Generate Alerts and Advisories

Description:

Overview: This process is responsible for generating the messages to advise and protect motorists, travelers and train crews approaching and crossing railroad grade crossings. Based on the severity of the hazard condition sent by the Detect HRI Hazards process this process will either send an hri_advisory command for non-time critical data or an hri_alert command for time critical data to the Report Alerts and Advisories. These users that will receive these messages include drivers, bicyclists, and pedestrians.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS14

Data Flows:

Input Flows

hazard_condition

Output Flows

hri_advisory
hri_alert

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.1;
USR = 1.10.1.5;

1.6.1.4.2 Provide Closure Parameters

Description:

Overview: This process is responsible for providing the HRI predicted time to closure to be used in broadcast message alerts to approaching vehicles. This time is calculated from data provided by the Detect HRI Hazards process.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS13

Data Flows:

Input Flows

hazard_condition

Output Flows

time_to_closing

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.2;
USR = 1.10.2.2;
USR = 1.10.2.2.4;

1.6.1.4.3 Report Alerts and Advisories

Description:

Overview: This process is responsible for reporting real-time HRI traffic volume advisories and real-time highway traffic alerts. Depending on the input received from the Generate Alerts and Advisories process, this process sends alerts or advisories to a train to describe the operational status of the intersection and alerts about any hazards. This process also sends the commands to Output Control Data for Roads process that will control the dynamic message signs in the area of an HRI to display the appropriate alert or advisory. Messages for local beacon broadcast are processed and sent to the Report HRI Status on Approach process. Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS14

Data Flows:

Input Flows

hri_advisory
hri_alert

Output Flows

approach_warning
hri_guidance_for_dms
train_message

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.3;
USR = 1.10.3.3;
USR = 1.10.3.3.5;

1.6.1.4.4 Report HRI Status on Approach

Description:

Overview: This process is responsible for providing real-time HRI status to vehicles as they approach an HRI. It must discriminate between vehicles near, but not approaching, the HRI (e.g. on parallel side streets, etc.). This process develops the message to be broadcast to nearby vehicles by receiving time_to_closing data and the hazard_condition signal and calculating the appropriate window of time to display the message. The message is built from the approach_warning data received from the Report Alerts and Advisories process. Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS13

Data Flows:

Input Flows

approach_warning
hazard_condition
time_to_closing

Output Flows

hri_guidance_for_beacon_message

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.5;
USR = 1.10.5.2;
USR = 1.10.5.2.6;

1.6.1.5 Detect HRI Hazards

Description:

Overview: This process is responsible for detecting real-time HRI blockages or collisions in the vicinity of an HRI that create a blockage or other hazard at the HRI. Based upon information received from the Provide Advance Warnings process this process can send a request to the Control Traffic Volume at Active HRI that the local signal strategy be preempted. A hazard condition message can also be sent to the Generate Alerts and Advisories process for further action or the Provide Closures Parameters process to possibly adjust the time to closing.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS14

Data Flows:

Input Flows

hri_hazard

Output Flows

hazard_condition
intersection_blocked
strategy_preemption

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.3;
USR = 1.10.3.3;
USR = 1.10.3.3.3;
USR = 1.10.6;

1.6.1.6.1 Close HRI on Detection

Description:

Overview: This process is responsible for protecting highway vehicles approaching and crossing railroad grade crossings by initiating the closure up to 3 minutes before train arrival. This process receives the near term status of the crossing including any approaching trains or trapped vehicles. With this information along with the local control plan data the predicted hri state is computed and sent to the Detect Imminent Vehicle/Train Collision process. If a hri_predicted_collision message is returned then this process sends out an hri_hazard message to the Detect HRI Hazard which will in turn result in a change to the device control strategy. This process also receives rail operations advisories for processing along with the state and control plan data. As needed this process will output any rail_operations_message data to the Interact with Rail Operations process.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS14

Data Flows:

Input Flows

current_hri_state
hri_predicted_collision
local_control_plan
rail_operations_advisories

Output Flows

hri_blockage
hri_hazard
near_term_status
predicted_hri_state
rail_operations_message

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.1;
USR = 1.10.1.4;
USR = 1.10.5;
USR = 1.10.5.2;
USR = 1.10.5.2.1;

1.6.1.6.2 Detect Imminent Vehicle/Train Collision

Description:

Overview: This process is responsible for detecting imminent collisions between vehicles and trains at railroad grade crossings. Using the data contained in the predicted_hri_state message this process performs the necessary calculations to determine whether a collision is imminent. If so, this process returns a hri_predicted_collision message to the Close_HRI_on_Detection process. Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS14

Data Flows:

Input Flows

predicted_hri_state

Output Flows

hri_predicted_collision

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.3;
USR = 1.10.3.1;

1.6.1.7.1 Control Vehicle Traffic at Passive HRI

Description:

Overview: This process is responsible for controlling traffic volume at passive grade crossings. It provides a mechanism for rail operations to close grade crossings that have active traffic devices but no real-time train detection mechanisms. This process also will allow for a train crew member to manually activate closure of the crossing. In such an event a crew_close_hri signal is sent to the Close_HRI_on_Command process.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS13

Data Flows:

Input Flows

rail_operations_device_command

Output Flows

crew_close_hri

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.4;
USR = 1.10.4.2;
USR = 1.10.4.2.1;

1.6.1.7.2 Control Vehicle Traffic at Active HRI

Description:

Overview: This process is responsible for controlling vehicular traffic at an active HRI by controlling the operation of traffic control devices in accordance with a predetermined local control plan. The local_control_plan is communicated to the Close_HRI_on_Detection process. This local control plan can be preempted by a strategy_preemption message from the Detect_HRI_Hazards process or by such inputs as an event_notice from the Detect_Roadway_Events process or hri_traffic_surveillance data. The outputs of this process include the command messages to close the HRI, requests for information from the Manage Traffic function, and information about the current hri_traffic_data.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS13

Data Flows:

Input Flows

event_notice
hri_traffic_surveillance
preemption_command
strategy_preemption

Output Flows

close_hri
hri_traffic_data
local_control_plan
traffic_management_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.3;

1.6.1.7.3 Close HRI on Command

Description:

Overview: This process is responsible for closing the HRI to vehicular traffic, either on command from the Control Traffic Volume at Active HRI process, or from direct command from rail operations (as an override). A third mechanism for closing the HRI is defined for passive crossings, i.e. crossings without active train detection systems. Upon command from rail operations, or via manual operation by a train crewman, active traffic devices at an otherwise passive grade crossing may be activated to close the crossing.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS13

Data Flows:

Input Flows

close_hri
crew_close_hri
rail_operations_device_command

Output Flows

hri_control_message

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.4;
USR = 1.10.4.1;
USR = 1.10.5;
USR = 1.10.5.2;
USR = 1.10.5.2.1;

1.6.2.1 Exchange Data with Rail Operations

Description:

Overview: This process is responsible for exchanging routine data with rail operations. Such data being sent to the rail operators includes event schedules, requests for information from the Rail Operators, incident notification based on rail operations messages received from Close_HRI_on_Detection process and hri_priority_message data received from the Manage Alerts and Advisories process. This process receives maintenance schedules, train schedules, and incident notifications from the rail operators. This information is used to develop the rail operations update data that is passed onto the Manage Rail Traffic Control Data process and the rail operations priority data that is sent to the Manage Alerts and Advisories process. Functional Requirements: none.

In Subsystem:

Traffic Management

In Market Package:

ATMS13

ATMS14

ATMS15

Data Flows:

Input Flows

fro_incident_notification
fro_maintenance_schedules
fro_train_schedules
hri_priority_message
rail_operations_message

Output Flows

rail_operations_device_command
rail_operations_priority_data
rail_operations_update
ro_requests
tro_equipment_status
tro_event_schedules
tro_incident_notification

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.2;
USR = 1.10.2.1;

1.6.2.2 Manage Alerts and Advisories

Description:

Overview: This process is responsible for acquiring HRI advisory or alert data from rail operations and for providing HRI status to rail operations. The data managed by this process may be time critical, as in the case of alerts or priority messages, or not time critical, as in the case of advisories.

In Subsystem:

Traffic Management

In Market Package:

ATMS13

ATMS14

Data Flows:

Input Flows

hri_blockage
hri_status
rail_operations_data
rail_operations_priority_data

Output Flows

hri_priority_message
rail_operations_advisories
rail_operations_query

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.2;
USR = 1.10.2.2;
USR = 1.10.2.2.1;
USR = 1.10.5;
USR = 1.10.5.2;
USR = 1.10.5.2.4;

1.6.2.3 Manage Rail Traffic Control Data

Description:

Overview: This process is responsible for providing and maintaining a current store of rail operations data. The data is assembled from the rail_operations_update information sent by the Exchange Data with Rail Operations process. Queries for this information are received from the Manage Alerts and Advisories process and the Interact with Vehicle Traffic Management processes. Functional Requirements: none.

In Subsystem:

Traffic Management

In Market Package:

ATMS15

Data Flows:

Input Flows

rail_operations_query
rail_operations_update
request_rail_schedules_data

Input/Output

rail_traffic_control_data

Output Flows

rail_operations_data
rail_schedules_data

rail_traffic_control_data = rail_operations_update + rail_operations_query;
rail_operations_data = rail_operations_query;
rail_schedules_data = request_rail_schedules_data;

User Service Requirements Mapped to Pspec:

USR = 1.0

USR = 1.10

USR = 1.10.2

USR = 1.10.2.2

USR = 1.10.2.2.2;

1.6.3.1 Interact with Wayside Systems

Description:

Overview: This process is responsible for interfacing to railroad owned and maintained wayside equipment, such as Wayside Interface Units, Crossing Gate Controllers, etc. All these devices are expected to provide real-time information to the HRI about approaching trains and their own health. In addition, advanced implementations will make use of a communications path back to approaching trains provided by the railroad's equipment. Functional Requirements: none.

In Subsystem:

Roadway Management

In Market Package:

ATMS13

ATMS14

Data Flows:

Input Flows

ats alert
fwe_approaching_train_announcement
fwe_train_data
fwe_wayside_equipment_status
hri_reporting_data

Output Flows

approaching_train_announcement
approaching_train_data
ats_status
twe_hri_status
twe_stop_highway_indication
twe_stop_train_indication
wayside_status

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.1;
USR = 1.10.1.1;
USR = 1.10.1.2;
USR = 1.10.1.3;
USR = 1.10.1.6;
USR = 1.10.1.7;
USR = 1.10.2;
USR = 1.10.2.2;

1.6.3.2 Advise and Protect Train Crews

Description:

Overview: This process is responsible for generating advisories/ alerts that are routed to the wayside equipment for transmission to the train crews. If the intersection is blocked, or there is an incident at the intersection this information will be passed to the Interact with Wayside Systems process for routing to the wayside equipment. The wayside equipment can then route the information directly to the train crews, or to rail operations.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS14

Data Flows:

Input Flows

approaching_train_announcement
ats_warning_notification
hri_status
train_message

Output Flows

ats_advisory
hri_reporting_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.1;
USR = 1.10.1.6;

1.6.3.3 Provide ATS Alerts

Description:

Overview: This process is responsible for automatically protecting commuter, intercity, transit and freight trains as they approach and cross grade crossings. It also reports HRI rail traffic advisories to traffic management and rail operations. It is responsible for verifying and reporting overall HRI status to approaching trains so that crews can act within safe service braking distances. It provides for notification of Automatic Train Stop systems (ATS, PTS, etc) with sufficient advance warning to allow emergency brake application time to stop a train before it encounters an HRI hazard. Finally, it provides automatic status indications about the HRI to the crews of approaching trains.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS14

Data Flows:

Input Flows

ats_advisory
ats_status

Output Flows

ats_alert
ats_warning_notification
hri_rail_alert

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.1;
USR = 1.10.1.2;
USR = 1.10.1.3;
USR = 1.10.3;
USR = 1.10.3.3;
USR = 1.10.3.3.5;
USR = 1.10.5;
USR = 1.10.5.2;
USR = 1.10.5.2.3;
USR = 1.10.5.2.5;

1.6.4.1 Manage HRI Closures

Description:

Overview: This process is responsible for coordination and managing of HRI closures at the Traffic management Center. It interfaces with Manage Incidents process to provide incident information and to receive strategy overrides as required by the larger incident management function. Functional Requirements: none.

In Subsystem:

Traffic Management

In Market Package:

ATMS13

ATMS14

Data Flows:

Input Flows

hri_strategy_override
hri_traffic_data
train_ops_plan

Output Flows

closure_event_data
hri_incident_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.2;
USR = 1.10.2.1;
USR = 1.10.2.1.3;

1.6.4.2 Exchange Data with Traffic Management

Description:

Overview: This process is responsible for interacting with traffic management processes. It collects data from processes that are within the HRI elements located at the roadside and forwards the data as needed to other processes within traffic management. It also acts as the interface between rail operations and traffic management processes through its interface with the Interact with Rail Operations process. Functional Requirements: none.

In Subsystem:

Traffic Management

In Market Package:

ATMS13

ATMS14

Data Flows:

Input Flows

closure_event_data
hri_status
intersection_blocked
rail_schedules_data
traffic_management_request
traffic_surveillance_data

Output Flows

hri_sensor_data
hri_status_for_traffic_demand
hri_traffic_surveillance
request_rail_schedules_data
tms_requests
train_ops_plan

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.2;
USR = 1.10.2.1;
USR = 1.10.2.1.2;
USR = 1.10.2.2;
USR = 1.10.2.2.4;

1.6.5.1 Provide Interactive Interface

Description:

Overview: This process is responsible for initiating reports of the health status of the HRI to both Traffic Management and Rail Operations. In addition the process initiates reporting of the health status of the HRI to the wayside interface equipment (and ultimately to the train when the advanced HRI functionality is in place).

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS14

Data Flows:

Input Flows

hri_closure_data_response
hri_state
ro_requests
tms_requests

Output Flows

hri_status
request_hri_closure_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.2;
USR = 1.10.2.2;
USR = 1.10.2.2.1;
USR = 1.10.2.2.2;
USR = 1.10.2.2.3;

1.6.5.2 Determine HRI Status

Description:

Overview: This process is responsible for monitoring critical HRI functions and merging them into a single coherent picture of the state of the hri. It also is responsible for assuring that the HRI always reverts to the safest possible operating condition in the event of any operational malfunctions.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS13

Data Flows:

Input Flows

device_status
hri_rail_alert
roadway_status
wayside_status

Output Flows

hri_state
preemption_command

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.3;
USR = 1.10.3.1;

1.6.5.3 Maintain HRI Closure Data

Description:

Overview: This process is responsible for managing a log of the HRI operation for use in strategy planning, demand management and traffic management. Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS13

Data Flows:

Input Flows

hri_state
request_hri_closure_data

Input/Output

hri_closure_data

Output Flows

hri_closure_data_response

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.10;
USR = 1.10.2;
USR = 1.10.2.1;
USR = 1.10.2.1.3;

1.7.1.1 Gather O&M Data

Description:

Overview: This process is responsible for gathering data to support the Operations and Maintenance function. The data gathered by this process includes incident information, traffic information, weather predictions, and static data about the makeup of the roadway network being maintained. This process collects information concerning scheduled maintenance activities in the region as well as data concerning the maintenance infrastructure. This data is collected and forwarded to other processes that archive the information or disseminate it to other agencies. This data also includes information concerning the management of disasters to be used to coordinate area wide responses.

Functional Requirements: none.

In Subsystem:

Maintenance Management

In Market Package:

ATMS24

ATMS25

EM5

Data Flows:

Input Flows

environmental_maintenance_data
fdca_disaster_maintenance_advisory
incidents_for_maintenance
infrastructure_information
maintenance_activity_data
maintenance_data_request
maintenance_link_data
traffic_data_for_maintenance

Output Flows

asset_management_data
maintenance_data_for_archive
maintenance_data_for_dissemination

User Service Requirements Mapped to Pspec:

USR = C2.4.1.2;
USR = C2.4.1.4;
USR = C2.5.1.1;
USR = C2.5.1.5;
USR = C2.5.2.1;
USR = C6.3.2.1;

1.7.1.2 Manage O&M Archive Data

Description:

Overview: This process shall be responsible for processing requests for archive data of operations and maintenance. This process shall receive maintenance data that has been gathered from the roadway, traffic, and maintenance sources. This process shall receive and respond to requests from the Manage Archived Data process for either a catalog of the data contained with the O&M data stores or for the data itself. Additionally this process shall be able to produce sample products of the data available. As data is received into this process quality control metrics shall be assigned. The appropriate meta-data shall be generated and stored along with the data. The process runs when a request for data is received from an external source, or when fresh data is received.

Functional Requirements: none.

In Subsystem:

Maintenance Management

In Market Package:

ATMS24

Data Flows:

Input Flows

asset_management_storage_data
maintenance_archive_request
maintenance_archive_status
maintenance_data_for_archive
O_and_M_archive_data

Output Flows

archived_maintenance_data
maintenance_archive_data
O_and_M_archive_data

User Service Requirements Mapped to Pspec:

USR = C2.5.1.2;
USR = C2.5.1.3;

1.7.1.3 Manage O&M Assets

Description:

Overview: This process supports the management of maintenance fleets as well as contruction work zone activity. This process maintains an inventory of the assets within the infrastructure owned and managed by the region. This process collects inputs from traffic surveillance and traffic device control functions as they report on faults on traffic devices. This process tracks those faults and reports their status. This process provides data concerning the asset inventory as requested to support the scheduling activity and the information dissemination functions.

Functional Requirements: none.

In Subsystem:

Maintenance Management

In Market Package:

ATMS24

ATMS25

Data Flows:

Input Flows

asset_management_data
indicator_new_fault
infrastructure_data
roadway_asset_status
sensor_fault_data_report

Output Flows

asset_management_plan
asset_management_storage_data
indicator_fault_clearance_update
infrastructure_data
infrastructure_information
sensor_fault_data_status

User Service Requirements Mapped to Pspec:

USR = C2.5.1.2;
USR = C2.5.1.3;
USR = C2.5.2.1;

1.7.1.4.1 Gather and Process Environmental Data

Description:

Overview: This process is responsible for gathering environmental and road condition sensor data to support the Operations and Maintenance function. The data gathered by this process includes the outputs of environmental sensors and road condition sensors. The data comes from sensors at the roadside, or from sensors on maintenance vehicles. The sensor data may be processed to create a single view of environmental or road conditions. This data is collected and forwarded to other processes that archive the information or disseminate it to other agencies, or fuse the data with weather data from weather sources to create a more complete picture of the environment at the roadway.

Functional Requirements: none.

In Subsystem:

Maintenance Management

In Market Package:

ATMS20

ATMS21

Data Flows:

Input Flows

environmental_sensor_data_for_maintenance
environmental_sensor_data_from_vehicle
environmental_sensor_status_for_maintenance
environmental_sensor_status_from_vehicle
roadside_condition_data

Output Flows

environmental_sensor_control_to_vehicle
processed_environment_data
processed_environmental_data_for_fusing
processed_environmental_data_for_road_forecasting
tmsp_maintenance_roadway_data
tws_maintenance_roadway_data

User Service Requirements Mapped to Pspec:

USR = C2.4.1.2;
USR = C2.4.1.4;
USR = C8.1.1.1;

1.7.1.4.2 Create Road Forecasts

Description:

Overview: This process is responsible for creating road forecasts using either the fused weather related information or by directly utilizing sensor data. The process shall support the creation of meso and micro-predictions using algorithms to predict road conditions over individual sections of the roadway.

Functional Requirements: none.

In Subsystem:

Maintenance Management

In Market Package:

ATMS21

ATMS23

Data Flows:

Input Flows

fused_environmental_data_for_road_forcasting
processed_environmental_data_for_road_forecasting

Output Flows

road_forecasts

User Service Requirements Mapped to Pspec:

USR = C2.4.1.3;
USR = C8.1.2.1;
USR = C8.1.3.1;
USR = C8.1.3.2;
USR = C8.1.3.3
USR = C8.1.3.4;
USR = C8.1.3.5;

1.7.1.4.3 Fuse Environmental Data

Description:

Overview: This process is responsible for fusing weather related information to support the Operations and Maintenance function. The data gathered by this process includes the processed outputs of environmental sensors and road condition sensors, weather forecasts from the Weather Service, and weather prediction information from the Meteorological Service Provider. The data is fused by the process to create an improved forecast of local weather and pavement conditions.

Functional Requirements: none.

In Subsystem:

Maintenance Management

In Market Package:

ATMS22

Data Flows:

Input Flows

fmsp_roadway_micro_prediction
fmsp_weather_prediction
fws_current_weather
fws_predicted_weather
processed_environmental_data_for_fusing

Output Flows

fused_environment_data
fused_environmental_data_for_road_forcasting

User Service Requirements Mapped to Pspec:

USR = C2.4.1.3;
USR = C2.4.1.4;
USR = C8.1.1.1;
USR = C8.1.1.2;
USR = C8.1.1.3;
USR = C8.1.1.4;
USR = C8.1.1.5;
USR = C8.1.3.1;
USR = C8.1.3.2;

1.7.2.1 Schedule Maintenance Activity

Description:

Overview: This process shall generate new maintenance schedules for use by maintenance vehicles, maintenance operators, and for information coordination purposes with other processes within Manage Traffic. The process shall use parameters and input data set up by the maintenance operator, incident data for the current traffic situation, data gathered from the roadway, data input from the maintenance vehicles and knowledge of the assets within the infrastructure. The process shall also respond to requests for resources from the Manage Incidents function as part of the scheduling system. The generation of new schedules by the process shall be initiated as a result of data received from the maintenance operator interface or a request for maintenance resources. The process shall send its output to the Gather Data function and to the maintenance vehicles to dispatch and route them appropriately.

Functional Requirements: none.

In Subsystem:

Maintenance Management

In Market Package:

ATMS24

ATMS25

EM4

Data Flows:

Input Flows

archived_maintenance_data
asset_management_plan
fdca_disaster_maintenance_request
maintenance_incident_coordination
maintenance_resource_request
maintenance_vehicle_activity_status
maintenance_vehicle_status
operator_schedule_inputs
roadway_maintenance_activity_data

Output Flows

maintenance_activity_data
maintenance_activity_device_control
maintenance_incident_information
maintenance_resource_response
maintenance_vehicle_dispatch
operator_schedule_outputs
tdca_disaster_maintenance_response

User Service Requirements Mapped to Pspec:

USR = C2.5.1.1;
USR = C2.5.1.2;
USR = C2.5.1.3;
USR = C2.5.1.4;
USR = C2.5.1.6;
USR = C2.5.1.9;
USR = C2.5.2.1;
USR = C2.5.2.2;
USR = C2.5.2.3;
USR = C2.5.2.4;
USR = C2.5.2.5;
USR = C2.5.2.7;

1.7.2.2 Manage and Control On-board Maintenance Systems

Description:

Overview: This process shall collect and process data available to sensors on-board maintenance vehicles. This data shall be sent by this process to other processes in the Manage O&M function for use in determining vehicle schedule deviations, maintenance activity status, or status of the on-board equipment itself. The data shall also include environmental sensor data obtained from sensors on-board the vehicle. This process receives inputs from the Process Vehicle Locate Data to determine the current position of the maintenance vehicle. It also receives inputs from the roadway to sense the conditions of the road that may be pertinent to the operation. This process receives dispatch information from the Schedule Maintenance Activity function and inputs from the Maintenance Vehicle Driver Interface function to determine a course of action.

Functional Requirements: none.

In Subsystem:

Maintenance Vehicle

In Market Package:

ATMS24

ATMS25

Data Flows:

Input Flows

environmental_sensor_control_to_vehicle
fmv_maintenance_measures
From_Roadway
maintenance_vehicle_dispatch
maintenance_vehicle_operator_controls
onboard_environmental_data
onboard_environmental_sensor_status
onboard_sensor_control_by_driver
vehicle_location_for_maintenance

Output Flows

environmental_sensor_data_from_vehicle
environmental_sensor_status_from_vehicle
maintenance_status_for_driver
maintenance_vehicle_activity_status
maintenance_vehicle_status
onboard_environmental_sensor_control
tmv_maintenance_control

User Service Requirements Mapped to Pspec:

USR = C2.4.1.1;
USR = C2.4.1.2;
USR = C2.5.1.4;
USR = C2.5.1.6;
USR = C2.5.2.2;
USR = C2.5.2.3;
USR = C2.5.2.4;
USR = C2.5.2.5;
USR = C8.1.1.1;
USR = C8.1.3.1;

1.7.2.3 Manage Maintenance Operator Interface

Description:

Overview: This process shall manage the interface to the Operator of the maintenance systems. This process receives inputs from the Maintenance Operator concerning schedule parameters or new information concerning maintenance resources in the area. This process also displays outputs to the operator such as schedule updates or flashes of new requests from other management functions within ITS.

Functional Requirements: none.

In Subsystem:

Maintenance Management

In Market Package:

ATMS24

ATMS25

Data Flows:

Input Flows

fmo_maintenance_operator_inputs
operator_schedule_outputs

Output Flows

operator_schedule_inputs
tmo_maintenance_operator_outputs

User Service Requirements Mapped to Pspec:

USR = C2.5.1.2;
USR = C2.5.1.3;
USR = C2.5.1.6;
USR = C2.5.1.7;
USR = C2.5.1.9;
USR = C2.5.2.4;
USR = C2.5.2.7;

1.7.2.4 Manage Maintenance Driver Interface

Description:

Overview: This process shall manage the interface to the Driver of the maintenance vehicles. This process receives inputs from the Maintenance Vehicle Driver such as requests for status from on-board systems, responses to queries for driver status, and requests for information from the Schedule Maintenance Activity function such as new dispatch orders or an update to weather conditions in the area. This process then formulates the response to be output to the driver based on received input from the on-board systems. This process formats the data to be presented to the driver either in digital screen displays or audio formats. This process will also retrieve status updates from the driver concerning maintenance activities and requests for follow-up support.

Functional Requirements: none.

In Subsystem:

Maintenance Vehicle

In Market Package:

ATMS24

ATMS25

Data Flows:

Input Flows

fmvd_maintenance_driver_inputs
maintenance_status_for_driver

Output Flows

maintenance_vehicle_operator_controls
onboard_sensor_control_by_driver
tmvd_maintenance_driver_data

User Service Requirements Mapped to Pspec:

USR = C2.5.1.4;
USR = C2.5.1.5;
USR = C2.5.1.6;
USR = C2.5.2.3;

1.7.2.5 Collect Environmental Data On-board

Description:

Overview: This process shall be responsible for collecting data obtained from environmental sensors which are on-board the maintenance vehicle. Where any of the data is provided in analog form, the process shall be responsible for converting it into digital form and calibrating. The converted data shall be sent to other processes for distribution, further analysis and storage.

Functional Requirements: None.

In Subsystem:

Maintenance Vehicle

In Market Package:

ATMS20

ATMS21

Data Flows:

Input Flows

fre_roadside_data
fre_roadside_hazards
onboard_environmental_sensor_control

Output Flows

onboard_environmental_data
onboard_environmental_sensor_status

User Service Requirements Mapped to Pspec:

USR = C2.4.1.1;
USR = C2.4.1.2;
USR = C8.1.1.1;
USR = C8.1.3.1;

1.7.3 Control Roadway Maintenance Devices

Description:

Overview: This process is controlling assets located at or alongside the roadway that can be used to monitor and control traffic round work zones. This process takes in traffic and environmental data from the field and based on inputs from the Schedule Maintenance Activity function sends control data out to the traffic devices in the field. This process can send traveller information via dynamic message signs or highway advisory radios. This process sends updated information regarding the roadway assets being used to the Manage Assets process.

Functional Requirements: none.

In Subsystem:

Maintenance Management

In Market Package:

ATMS25

Data Flows:

Input Flows

environmental_maintenance_status
information_device_fault_status
maintenance_activity_device_control
traffic_control_device_status
traffic_sensor_maintenance_data
traffic_sensor_maintenance_status

Output Flows

highway_device_control_for_maintenance
highway_information_maintenance_data
road_device_control_for_maintenance
road_information_maintenance_data
roadway_asset_status
roadway_maintenance_activity_data
surveillance_control_data_for_maintenance

User Service Requirements Mapped to Pspec:

USR = C2.5.2.1;
USR = C2.5.2.2;
USR = C2.5.2.3;
USR = C2.5.2.5;
USR = C2.5.2.6;

1.7.4 Disseminate O&M Information

Description:

Overview: This process is responsible for disseminating information to other agencies within the region being served by the Manage ITS function. The data is received from the Gather O&M Data function and is packaged for the appropriate using functions such as transit, emergency, traffic, and traveller information.

Functional Requirements: none.

In Subsystem:

Maintenance Management

In Market Package:

ATMS24

Data Flows:

Input Flows

maintenance_data_for_dissemination

Output Flows

maintenance_data_request
maintenance_incidents_for_emergencies
maintenance_status_for_transit
maintenance_status_for_travellers
maintenance_vehicle_status
roadway_traffic_maintenance_information
work_zone_status

User Service Requirements Mapped to Pspec:

USR = C2.5.1.8;
USR = C2.5.1.9;

2.1.1 Manage Commercial Fleet Electronic Credentials and Tax Filing

Description:

Overview: This process shall be responsible for providing the commercial vehicle manager with the ability to manage the activities of commercial vehicles. The process shall provide the capability for the manager to obtain commercial vehicle routes. When a route has been confirmed, the process shall enable the manager to enroll commercial vehicles for electronic clearance at roadside check station facilities, to process and pay for electronic credential and tax filing, to send tag data to the Provide Commercial Vehicle On-board Data facility, and to provide vehicle route instructions for use by the commercial vehicle driver. Periodically it shall also send reports about taxes that have been paid to the Administer Commercial Vehicles facility. The process shall also enable the manager to obtain commercial vehicle activity reports from the logs provided by roadside checkstation facilities. It shall be possible to obtain these reports either on request or at periodic intervals.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO04

CVO05

CVO10

Data Flows:

Input Flows

cf_enrollment_information
cf_enrollment_payment_confirmation
cf_hazmat_request
cf_manager_activity_report_request
cf_manager_enrollment_payment_request
cf_manager_enrollment_request
cf_manager_route_request
cf_manager_storage_request
cf_periodic_activity_report
cf_roadside_activity_report
cf_route
cf_static_route_data
cf_tag_data
freight_cargo_data

Input/Output

cf_retained_data
cf_route_details

Output Flows

cf_driver_route
cf_enrollment_payment_request
cf_enrollment_request
cf_hazmat_route_information
cf_hazmat_vehicle_information
cf_manager_activity_report
cf_manager_enrollment_information
cf_manager_enrollment_payment_confirmation
cf_manager_route_data
cf_request_activity_report
cf_route_request
cf_static_route_request
cf_tag_initialization_data
cf_tax_audit_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.4;
USR = 4.4.1;
USR = 4.4.1(e);
USR = 4.4.1(f);
USR = 4.4.1(g);
USR = 4.6;
USR = 4.6.1;

2.1.2 Provide Commercial Fleet Static Route

Description:

Overview: This process shall be responsible for providing a static commercial vehicle route using data provided by the commercial vehicle manager. A static route is one which is based on geographic data and therefore takes no account of current or predicted traffic conditions, incidents, etc. The process shall provide the route using its own route generation algorithms and data from its own store of digitized map information.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO01

Data Flows:

Input Flows

cf_static_route_request

Input/Output

map_data_for_fleet_managers

Output Flows

cf_static_route_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.4.0;
USR = 4.4.1;
USR = 4.4.1(c);
USR = 4.4.1(d);

2.1.3 Provide Flt Mgr Electronic Credentials and Tax Filing Interface

Description:

Overview: This process shall be responsible for providing an interface for the commercial vehicle manager. The process shall enable this interface to provide the manager with facilities for the input of data used to set up commercial vehicle routes, to pay the necessary taxes and duties so that a commercial vehicle can be enrolled for a particular route, to exchange general information messages with a driver in a vehicle, and to set up instructions for a driver to take a vehicle on a particular route. It shall be possible for the driver's route instructions input by the manager to include details of the cargo to be picked up and/or dropped off at each point along the route. The enrollment activity supported by the process shall enable a commercial vehicle to pass through the roadside checkstations along its route without stopping, unless safety checks are required. The process shall support inputs from the commercial vehicle manager in both manual and audio form, and shall provide its outputs in audible and visual forms. It shall enable the visual output to be in hardcopy, or as a display.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO04

CVO05

Data Flows:

Input Flows

cf_driver_route_instructions_output
cf_manager_activity_report
cf_manager_credit_identity
cf_manager_enrollment_information
cf_manager_enrollment_payment_confirmation
cf_manager_route_data
cf_vehicle_data
fcvm_enrollment_payment_request
fcvm_enrollment_request
fcvm_other_data_input
fcvm_preclearance_data
fcvm_request_driver_route_instructions
fcvm_request_on_board_vehicle_data
fcvm_roadside_activity_report_request
fcvm_route_data
fcvm_route_function_request
fcvm_update_driver_route_instructions

Output Flows

cf_driver_instructions_request
cf_driver_load_data
cf_manager_activity_report_request
cf_manager_enrollment_cost
cf_manager_enrollment_payment_request
cf_manager_enrollment_request
cf_manager_route_request
cf_manager_storage_request
cf_request_vehicle_data
tcvm_data_input_request
tcvm_driver_route_instructions
tcvm_enrollment_confirmation
tcvm_enrollment_payment_confirmation
tcvm_other_data_request
tcvm_preclearance_results
tcvm_roadside_activity_report
tcvm_route_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.6;
USR = 4.6.1;
USR = 4.6.2;

2.1.4 Provide Fleet Manager Commercial Vehicle Communications

Description:

Overview: This process shall be responsible for providing the communications interface and data storage facility for data that is exchanged between the commercial vehicle manager and commercial vehicle drivers in their vehicles. The process shall support the receipt of data from the vehicle consisting of that processed from input received by sensors on board the vehicle and text data used to exchange general information with the driver. Only the output to the vehicle of the data that contains the general text message shall be supported by the process. The process shall enable access to the store of received data by the manager through the manager's interface process.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO01

Data Flows:

Input Flows

cf_on_board_vehicle_data
cf_request_vehicle_data
cv_vehicle_location
cv_vehicle_location_request

Input/Output

cf_retrieved_vehicle_data

Output Flows

cf_request_on_board_vehicle_data
cf_vehicle_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.6;
USR = 4.6.0;
USR = 4.6.1;
USR = 4.6.2;
USR = 4.6.2(a);
USR = 4.6.2(b);

2.1.5 Provide Commercial Vehicle Driver Routing Interface

Description:

Overview: This process shall be responsible for providing the communications interface through which a commercial vehicle driver can obtain details of the vehicle route that has been provided by the commercial vehicle manager. The process shall enable the output of the route instructions in audio and/or visual form. It shall be possible for the visual form to be either hardcopy output, or in the form of a display. The process shall retain the data for a particular route internally, so that successive requests for details of the same route do not require use of the communications network.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO01

CVO08

CVO09

Data Flows:

Input Flows

cf_driver_route_instructions
cf_intermodal_dispatch_assigment
cf_intermodal_route_instructions
fcvd_request_routing_instructions

Output Flows

cf_driver_route_instructions_request
tcvd_routing_instructions

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.6;
USR = 4.6.0;
USR = 4.6.1;
USR = 4.6.2;
USR = 4.6.2(a);
USR = 4.6.2(b);
USR = 4.6.2(c);
USR = C5.5.2.4;

2.1.6 Manage Driver Instruction Store

Description:

Overview: This process shall be responsible for managing the store of driver route instructions so that they can be loaded with data for retrieval by the commercial vehicle driver. The data for loading into the store shall be sent to the process from other processes in the Manage Commercial Vehicle Fleet Operations facility of the Manage Commercial Vehicles function. The process shall enable the data to comprise vehicle route data and vehicle load information, including the points along the route at which identified cargo is to be picked up and/or dropped off. The process shall support the retrieval of this data by the commercial vehicle driver through the driver's interface process.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO01

CVO01

CVO09

Data Flows:

Input Flows

cf_driver_instructions_request
cf_driver_load_data
cf_driver_route
cf_driver_route_instructions_request
intermodal_dispatch_assignment

Input/Output

cf_driver_instructions

Output Flows

cf_driver_route_instructions
cf_driver_route_instructions_output
cf_intermodal_dispatch_assigment
cf_intermodal_route_instructions

User Service Requirements Mapped to Pspec:

USR = C5.5.2.4;

2.2.1 Manage CV Electronic Credential and Tax Filing Interface

Description:

Overview: This process shall be responsible for providing the commercial vehicle driver with the ability to manage the activities of a commercial vehicle. In this instance the driver is assumed to be acting in the role of a commercial vehicle manager, and is therefore probably the owner/driver of the vehicle. The process shall provide the capability for the driver to obtain commercial vehicle routes, to enroll commercial vehicles for electronic clearance at roadside check station facilities, and to process and pay for electronic credential and tax filing.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO04

CVO05

Data Flows:

Input Flows

cv_driver_enrollment_payment_request
cv_driver_enrollment_request
cv_driver_route_request
cv_driver_storage_request
cv_enrollment_information
cv_enrollment_payment_confirmation
cv_route
cv_static_route_data

Input/Output

cv_route_details

Output Flows

cv_driver_enrollment_information
cv_driver_enrollment_payment_confirmation
cv_driver_route_data
cv_enrollment_payment_request
cv_enrollment_request
cv_route_request
cv_static_route_request

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.4;
USR = 4.4.1;
USR = 4.6;
USR = 4.6.1;

2.2.2 Provide Vehicle Static Route

Description:

Overview: This process shall be responsible for providing a static commercial vehicle route using data provided by the commercial vehicle driver. A static route is one which is based on geographic data and therefore takes no account of current or predicted traffic conditions, incidents, etc. The process shall provide the route using its own route generation algorithms and data from its own store of digitized map information. In this instance the driver is assumed to be acting in the role of a commercial vehicle manager, and is therefore likely to be the owner/driver of the vehicle.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO01

CVO08

CVO09

Data Flows:

Input Flows

cv_static_route_request

Input/Output

map_data_for_cv_drivers

Output Flows

cv_static_route_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.4;
USR = 4.4.1;

2.2.3 Provide CV Driver Electronic Credential and Tax Filing Interface

Description:

Overview: This process shall be responsible for providing an interface for the commercial vehicle fleet manager. In this instance the driver is assumed to be acting in the role of a commercial vehicle manager, and is therefore likely to be the owner/driver of the vehicle. The process shall enable this interface to provide the driver with facilities for the input of data used to set up commercial vehicle routes, to pay all the necessary taxes and duties so that a commercial vehicle can be enrolled for a particular route, and to obtain a copy of the data collected by processes on-board the vehicle. The enrollment activity supported by the process shall enable a commercial vehicle to pass through the roadside checkstations along its route without stopping, unless safety checks are required. The process shall support inputs from the commercial vehicle driver in both manual and audio form, and shall provide its outputs in audible and visual forms. It shall enable the visual output to be in hardcopy, or as a display.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO03

CVO05

CVO06

CVO07

Data Flows:

Input Flows

cv_driver_credit_identity
cv_driver_enrollment_information
cv_driver_enrollment_payment_confirmation
cv_driver_route_data
cv_vehicle_data
fcvd_activity_request
fcvd_enrollment_payment_request
fcvd_enrollment_request
fcvd_other_data_input
fcvd_route_data
fcvd_route_request

Output Flows

cv_driver_enrollment_cost
cv_driver_enrollment_payment_request
cv_driver_enrollment_request
cv_driver_route_request
cv_driver_storage_request
cv_request_vehicle_data
tcvd_data_request
tcvd_enrollment_confirmation
tcvd_enrollment_payment_confirmation
tcvd_other_data_request
tcvd_route_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.3;
USR = 4.3.2;
USR = 4.3.2.1;
USR = 4.4;
USR = 4.4.1;
USR = 4.4.2;
USR = 4.4.3;
USR = 4.4.3.1;
USR = 4.4.3.2;
USR = 4.4.3.2(a);
USR = 4.4.3.2(b);
USR = 4.4.3.2(c);
USR = 4.4.3.2(d);
USR = 4.4.3.2(e);
USR = 4.6;

2.2.4 Provide Commercial Vehicle Driver Communications

Description:

Overview: This process shall be responsible for providing communications between the commercial vehicle driver and the commercial vehicle. In this instance the driver is acting in the role of vehicle manager, and is therefore likely to be the owner/driver of the vehicle. The process shall support the receipt of data from the vehicle consisting of that processed from input received by sensors on board the vehicle The process shall enable access to the store of received data by the driver through the driver's interface process.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO01

CVO08

CVO09

Data Flows:

Input Flows

cv_on_board_vehicle_data
cv_request_vehicle_data

Input/Output

cv_received_vehicle_data

Output Flows

cv_request_on_board_vehicle_data
cv_vehicle_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.6;
USR = 4.6.0;
USR = 4.6.1;
USR = 4.6.2;
USR = 4.6.2(a);
USR = 4.6.2(b);
USR = 4.6.2(c);

2.3.1 Produce Commercial Vehicle Driver Message at Roadside

Description:

Overview: This process shall be responsible for the output of pull-in or pass messages to commercial vehicle drivers as they approach the commercial vehicle roadside checkstation or border crossing facilities. The process shall support the use of roadside equipment such as dynamic message signs (DMS), or simple red-green lights, flashing orange lights, etc. to provide the output. These output messages shall be received by the process from other processes responsible for roadside facilities within the Manage Commercial Vehicles function. The process shall support pull-in messages that are the result of checks on a commercial vehicle's electronic credentials, safety and border crossing data, the result of the vehicle's tag not being properly read, or the result of a general pull-in decision for all vehicles being issued by inspectors at the roadside facility. The process shall also generate a message to be sent to the vehicle so that an indication can be output directly to the driver at the same time as it appears on the roadside equipment.

In Subsystem:

Commercial Vehicle Check

In Market Package:

CVO03

Data Flows:

Input Flows

cv_border_pull_in_output
cv_general_pull_in_output
cv_safety_pull_in_output
cv_screening_pull_in_output

Output Flows

cv_on_board_pull_in_output
tcvd_border_pull_in_output
tcvd_clearance_pull_in_output
tcvd_general_pull_in_output
tcvd_safety_pull_in_output

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1;
USR = 4.1.1.4;
USR = 4.1.2;
USR = 4.1.2.2;
USR = 4.3;
USR = 4.3.1;
USR = 4.3.1.2;
USR = 4.3.1.7;

2.3.2.1 Administer Commercial Vehicle Roadside Credentials Database

Description:

Overview: This process shall be responsible for receiving the electronic credentials sent to the roadside checkstation facility as part of a commercial vehicle's enrollment process. The process shall store the data for use by another process and shall also enable the inspector in the roadside facility to obtain a copy of the data in the store. If the requested data is not in the store, the process shall request it from another process in the commercial vehicle administration facility.

In Subsystem:

Commercial Vehicle Check

In Market Package:

CVO03

Data Flows:

Input Flows

cv_credentials_data_request
cv_credentials_database_update
cv_credentials_information_response

Input/Output

cv_roadside_credentials_database

Output Flows

cv_credentials_data_output
cv_credentials_information_request

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1;
USR = 4.1.1.8;

2.3.2.2 Process Screening Transactions

Description:

Overview: This process shall be responsible for checking commercial vehicle credentials against those held in a store maintained by another process in the roadside checkstation facility. The process shall send the result of each check to the roadside inspector interface process so that an override input can be generated if required. The process shall send a request for the commercial vehicle to pull-in if the vehicle's credentials do not match those in the store, and shall also send a record of each decision to the process that maintains the commercial vehicle roadside checkstation facility log.

In Subsystem:

Commercial Vehicle Check

In Market Package:

CVO03

Data Flows:

Input Flows

cv_roadside_credentials_database
cv_screening_data
cv_screening_override

Output Flows

cv_on_board_screening_record
cv_screening_decision
cv_screening_pull_in_output
cv_screening_record

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1;
USR = 4.1.1.4;
USR = 4.1.1.5;
USR = 4.1.1.7;
USR = 4.1.1.8;

2.3.3.1 Provide Commercial Vehicle Checkstation Communications

Description:

Overview: This process shall be responsible for providing an interface through which a commercial vehicles roadside checkstation facility can communicate with a passing commercial vehicle. When a request for on-board data is received from another process within the facility, the process shall issue a data request to the identified commercial vehicle. The data received by the process from the vehicle shall be stored in the store of collected data for use by the roadside inspection process.

In Subsystem:

Commercial Vehicle Check

In Market Package:

CVO07

Data Flows:

Input Flows

cv_get_on_board_data
cv_inspection_data
cv_on_board_data

Output Flows

cv_inspection_data_output
cv_request_on_board_data
cv_roadside_collected_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1.4;
USR = 4.1.1.7;
USR = 4.2;
USR = 4.2.2;
USR = 4.2.2.1;
USR = 4.2.2.7;
USR = 4.2.2.7(a);
USR = 4.2.2.7(b);
USR = 4.2.2.7(c);
USR = 4.2.2.7(d);
USR = 4.3;
USR = 4.3.1;
USR = 4.3.1.1;
USR = 4.3.1.3;
USR = 4.3.1.4;
USR = 4.3.1.6;
USR = 4.3.1.7;

2.3.3.2 Provide Commercial Vehicle Inspector Handheld Terminal Interface

Description:

Overview: This process shall be responsible for providing an interface for a hand held terminal which can be used by a commercial vehicle inspector. The process shall enable the inspector to start a commercial vehicle roadside inspection, to review the results, and to add comments to the results data. The process shall support inputs from the inspectors in both manual and audio form, and shall provide its outputs in audible and visual forms. It shall enable the form of the visual output to be in hardcopy, or as a display.

In Subsystem:

Commercial Vehicle Check

In Market Package:

CVO07

Data Flows:

Input Flows

cv_inspection_results
fci_inspection_data_input
fci_start_inspection

Output Flows

cv_inspector_safety_data_input
cv_start_inspection
tci_inspection_report

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.2;
USR = 4.2.2;
USR = 4.2.2.3;

2.3.3.3 Administer Commercial Vehicle Roadside Safety Datadase

Description:

Overview: This process shall be responsible for maintaining in the commercial vehicle roadside checkstation facility a database of credentials for commercial vehicles with safety problems. This process shall store the data about these vehicles received from the commercial vehicle administration facility. It shall enable this data to be used by another process and shall also enable the inspector in the roadside facility to obtain a copy of the data in the store. If the requested data is not in the store, the process shall request it from another process in the commercial vehicle administration facility.

In Subsystem:

Commercial Vehicle Check

In Market Package:

CVO07

Data Flows:

Input Flows

cv_roadside_safety_database_read
cv_safety_data_request
cv_safety_database_update
cv_safety_information_response

Output Flows

cv_roadside_safety_database_write
cv_safety_data_response
cv_safety_information_request

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.2;
USR = 4.2.2;
USR = 4.2.2.4;
USR = 4.2.2.6;

2.3.3.4 Carry-out Commercial Vehicle Roadside Safety Screening

Description:

Overview: This process shall be responsible for checking commercial vehicle credentials against the list of those known to have safety problems held in a store maintained by another process in the roadside checkstation facility. The process shall send the result of each check to the roadside inspector interface process so that an override input can be generated if required. The process shall send a request for the commercial vehicle to pull-in if the vehicle's credentials are in the list of those with safety problems, and shall also send a record of each decision to the process that maintains the commercial vehicle roadside checkstation facility log.

In Subsystem:

Commercial Vehicle Check

In Market Package:

CVO03

CVO07

CVO07

CVO08

Data Flows:

Input Flows

cv_roadside_safety_database_output
cv_safety_data
cv_safety_override

Output Flows

cv_archived_safety_data
cv_safety_decision
cv_safety_pull_in_output

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1;
USR = 4.1.1.1;
USR = 4.1.1.2;
USR = 4.1.1.3;
USR = 4.1.1.4;
USR = 4.1.1.6;
USR = 4.1.1.8;
USR = 4.2;
USR = 4.2.2;
USR = 4.2.2.6;
USR = 4.3;
USR = 4.3.1;
USR = 4.3.1.1;
USR = 4.3.1.1(a);
USR = 4.3.1.1(b);
USR = 4.3.1.3;
USR = 4.3.1.4;
USR = 4.4;
USR = 4.4.2;

2.3.3.5 Carry-out Commercial Vehicle Roadside Inspection

Description:

Overview: This process shall be responsible for carrying out roadside safety inspections at the request of the roadside facility inspector. The result of the inspection shall be sent by the process to the inspector, the commercial vehicle driver, the roadside checkstation facility log, and the commercial vehicle itself. The process shall enable the inspector to add comments to the result of the inspection before it is sent to the above outputs. These comments shall be received by the process in the form of data input from the inspector's hand held terminal interface.

In Subsystem:

Commercial Vehicle Check

In Market Package:

CVO07

Data Flows:

Input Flows

cv_inspector_safety_data_input
cv_roadside_collected_data
cv_start_inspection

Output Flows

cv_archived_inspection_data
cv_get_on_board_data
cv_inspection_data
cv_inspection_results
cv_roadside_safety_database_update
cv_update_safety_problems_list
tcvd_inspection_results

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.2;
USR = 4.2.2;
USR = 4.2.2.1;
USR = 4.2.2.2;
USR = 4.2.2.3;
USR = 4.2.2.6;
USR = 4.2.2.7;
USR = 4.2.2.7(a);

2.3.4 Detect Commercial Vehicle

Description:

Overview: This process shall be responsible for detecting the presence of commercial vehicles through the use of sensors that can differentiate between the different types of vehicle. The process shall use the sensors to determine the number of axles, gross vehicle weight and weight per axle data for use by inspectors at the roadside checkstation facilities. When a commercial vehicle is detected, the process shall transmit a request for its on-board tag data, which when received shall be passed to other processes within the roadside facility. If no tag data is received, or the data cannot be interpreted correctly, the process shall send a request for the vehicle to pull-in to be output by another process in the roadside checkstation facility.

In Subsystem:

Commercial Vehicle Check

In Market Package:

CVO03

CVO06

Data Flows:

Input Flows

cv_electronic_clearance_data
cv_general_override
cv_manual_pull_in
fcv_vehicle_characteristics

Output Flows

cv_border_data
cv_general_decision
cv_general_pull_in_output
cv_request_electronic_clearance_data
cv_safety_data
cv_screening_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.0;
USR = 4.1.1;
USR = 4.1.1.7;
USR = 4.2;
USR = 4.2.0;
USR = 4.2.1;
USR = 4.2.3;
USR = 4.2.3.2;
USR = 4.2.3.5;
USR = 4.3;
USR = 4.3.0;
USR = 4.3.1;
USR = 4.3.1.7;

2.3.5 Provide Commercial Vehicle RoadsideOperator Interface

Description:

Overview: This process shall be responsible for providing the commercial vehicle inspector interface at the roadside checkstation facility. The process shall provide an interface which enables the inspector to monitor and if necessary override the pull-in decisions made by those of the border crossing, credentials and safety data checking processes that are present in the facility. The process shall also make it possible for the inspector to issue a manual general pull-in request for all commercial vehicles to pull into the roadside checkstation facility, to have access the contents of the facility's log, and to obtain credentials or safety data on a selected combination of carrier, driver, and vehicle. The process shall support inputs from the traffic operations personnel in both manual and audio form, and shall provide its outputs in audible and visual forms. It shall enable the visual output to be in hardcopy, or as a display.

In Subsystem:

Commercial Vehicle Check

In Market Package:

CVO03

CVO07

Data Flows:

Input Flows

cv_border_decision
cv_credentials_data_output
cv_general_decision
cv_roadside_operator_output
cv_safety_data_response
cv_safety_decision
cv_screening_decision
fci_credentials_data_request
fci_pull_in_action
fci_request_log_report
fci_safety_data_request

Output Flows

cv_border_override
cv_credentials_data_request
cv_general_override
cv_manual_pull_in
cv_roadside_operator_data_request
cv_safety_data_request
cv_safety_override
cv_screening_override
tci_credentials_data_output
tci_output_log_report
tci_pull_in_information
tci_safety_data_output

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1;
USR = 4.1.1.4; USR = 4.1.1.4(a); USR = 4.1.1.4(b);
USR = 4.1.1.4(c);
USR = 4.1.1.5;
USR = 4.2;
USR = 4.2.2;
USR = 4.2.2.1;
USR = 4.2.2.2;
USR = 4.2.2.3;
USR = 4.2.2.3(a);
USR = 4.2.2.3(b);
USR = 4.2.2.3(c);
USR = 4.2.2.4;
USR = 4.2.2.6;
USR = 4.3;

2.3.6 Provide Commercial Vehicle Reports

Description:

Overview: This process shall be responsible for collecting data from those of the border crossing, credential and safety checking processes that are present in a commercial vehicle roadside checkstation facility. The data shall be stored by the process in a roadside facility log, to which the roadside inspector interface process shall have access. Once a day the process shall make a copy of the roadside facility log and send it to the commercial vehicle administration facility for further processing.

In Subsystem:

Commercial Vehicle Check

In Market Package:

CVO03

Data Flows:

Input Flows

cv_archived_inspection_data
cv_archived_safety_data
cv_border_record
cv_roadside_operator_data_request
cv_screening_record

Input/Output

cv_roadside_facility_log

Output Flows

cv_border_daily_log
cv_roadside_daily_log
cv_roadside_operator_output

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.2;
USR = 4.2.2;
USR = 4.2.2.4;
USR = 4.3;
USR = 4.3.1;
USR = 4.3.1.3;
USR = 4.4;
USR = 4.4.2;
USR = 4.4.2(b);

2.3.7 Produce Commercial Vehicle Driver Message on Vehicle

Description:

Overview: This process shall be responsible for the output of the pull-in or pass messages to commercial vehicle drivers directly in their vehicles as they approach a commercial vehicle roadside checkstation facility. These messages shall be generated by other processes within the facility that are responsible for checking the commercial vehicle's credentials (including those for border crossing) and safety, or may be the result of the vehicle's tag not being properly read, or may be the result of a general pull-in decision for all vehicles being issued by inspectors at the roadside checkstation facility.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO03

CVO05

CVO06

CVO07

Data Flows:

Input Flows

cv_on_board_pull_in_output

Output Flows

tcvd_on_board_pull_in_output

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1;
USR = 4.1.1.4;
USR = 4.3;
USR = 4.3.1;
USR = 4.3.1.2;
USR = 4.3.1.7;

2.3.8 Provide Commercial Vehicle Border Screening

Description:

Overview: This process shall be responsible for checking a commercial vehicle and its cargo through a border crossing point. The checks carried out by the process shall comprise a comparison of the trip identity already provided by the commercial vehicle administration processes, and held in a local data store. A check shall also be made by the process to see if the lock tag attached to the vehicle's cargo has been changed. If either of these two checks produce negative results then the process shall request the vehicle to pull-in, otherwise the vehicle shall be allowed to pass. The process shall send its decision to the process that provides the roadside inspectors' interface, to enable an override to be applied if required. The decision of the process (with the override if it is applied) shall be sent to the message output process and be written back to the vehicle's on-board tag.

In Subsystem:

Commercial Vehicle Check

In Market Package:

CVO05

Data Flows:

Input Flows

cv_border_data
cv_border_database_update
cv_border_override

Input/Output

cv_roadside_border_database

Output Flows

cv_border_decision
cv_border_pull_in_output
cv_border_record
cv_on_board_border_record

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.2;
USR = 4.1.2.2;
USR = 4.4.3;
USR = 4.4.3.1;
USR = 4.4.3.1(a);
USR = 4.4.3.1(b);
USR = 4.4.3.1(c);
USR = 4.4.3.2;
USR = 4.4.3.2(a);
USR = 4.4.3.2(b);
USR = 4.4.3.2(c);
USR = 4.4.3.2(d);
USR = 4.4.3.2(e);
USR = 4.4.3.2(f);
USR = 4.4.3.2(g);
USR = 4.4.3.2(h);

2.4.1 Communicate Commercial Vehicle On-board Data to Roadside

Description:

Overview: This process shall be responsible for providing the commercial vehicle end of the communications link between itself and a commercial vehicle roadside checkstation facility. The process shall enable an inspector at the facility or elsewhere to have access to the data accumulated on-board the vehicle for use in a vehicle inspection. It shall also enable the inspector to send back data about the result of the inspection for storage on-board the vehicle.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO01

CVO08

CVO08

CVO09

Data Flows:

Input Flows

cv_inspection_data_output
cv_on_board_data_current_copy
cv_request_on_board_data

Output Flows

cv_inspection_data_update
cv_on_board_data
cv_on_board_data_needed

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.2;
USR = 4.2.3;
USR = 4.2.3.3;
USR = 4.2.3.7;
USR = 4.3;
USR = 4.3.1;
USR = 4.3.1.6;
USR = 4.3.2;
USR = 4.3.2.3;

2.4.2 Collect On-board Commercial Vehicle Sensor Data

Description:

Overview: This process shall be responsible for continuously monitoring the conditions on-board a commercial vehicle. These inputs shall be processed by sensors, and if required converted from analog into a digital form. The process shall load all collected into an on-board vehicle data store for use by other processes in the vehicle.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO01

CVO03

CVO05

CVO06

CVO07

CVO08

CVO08

CVO09

Data Flows:

Input Flows

fcv_brake_condition
fcv_cargo_safety_status
fcv_distance_traveled
fcv_driver_safety_status
fcv_driver_status
fcv_vehicle_safety_status
fcv_weight

Output Flows

cv_on_board_stored_sensor_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1;
USR = 4.1.1.6;
USR = 4.1.2;
USR = 4.1.2.1;
USR = 4.2;
USR = 4.2.0;
USR = 4.2.3;
USR = 4.2.3.1;
USR = 4.2.3.2;
USR = 4.2.3.3;
USR = 4.2.3.4;
USR = 4.2.3.5;
USR = 4.2.3.6;
USR = 4.3;
USR = 4.3.1;
USR = 4.3.1.2;
USR = 4.3.2;

2.4.3 Analyze Commercial Vehicle On-board Data

Description:

Overview: This process shall be responsible for analyzing the data collected on-board a commercial vehicle, and sending it to another process for loading into a store on-board the vehicle. If the analysis of the data carried out by the process shows that there is a critical safety problem, the process shall send data to the driver's interface process for output to the driver. The process shall also accept input of data from the commercial vehicle driver via the interface process and load it into the same store.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO01

CVO03

CVO05

CVO06

CVO07

CVO08

CVO08

CVO09

Data Flows:

Input Flows

cv_driver_data_input
cv_on_board_stored_sensor_data

Output Flows

cv_critical_safety_problem
cv_driver_data_output
cv_on_board_data_update

User Service Requirements Mapped to Pspec:

USR = 4.2.3.1;
USR = 4.2.3.2;
USR = 4.2.3.4;
USR = 4.2.3.5;
USR = 4.3.2.1;
USR = 4.3.2.1(a);
USR = 4.3.2.1(b);
USR = 4.3.2.1(c);
USR = 4.3.2.1(d);
USR = 4.3.2.1(e);
USR = 4.3.2.2;
USR = 4.4.3.2;

2.4.4 Provide Commercial Vehicle Driver Interface

Description:

Overview: This process shall be responsible for providing the interface between the commercial vehicle driver and processes on-board the commercial vehicle. The process shall provide interfaces to the processes responsible for collecting, analyzing and storing data about the vehicle, its cargo, the driver, etc., and for the exchange of data with the commercial vehicle manager. The process shall support inputs from the driver in both manual and audio form, and shall provide its outputs in audible and visual forms. It shall enable the visual output to be in hardcopy, or as a display.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO01

CVO03

CVO05

CVO06

CVO07

CVO08

CVO08

CVO09

Data Flows:

Input Flows

cv_critical_safety_problem
cv_driver_data_output
cv_general_input_message
fcvd_driver_data_input
fcvd_driver_general_message
fcvd_driver_input_type

Output Flows

cv_driver_data_input
cv_general_output_message
cv_output_on_board_vehicle_data
tcvd_critical_safety_problem
tcvd_data_input_request
tcvd_output_data
tcvd_type_input_request

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1;
USR = 4.1.1.6;
USR = 4.1.1.6(a);
USR = 4.1.1.6(b);
USR = 4.1.1.6(c);
USR = 4.1.1.6(d);
USR = 4.1.1.6(e);
USR = 4.0;
USR = 4.3;
USR = 4.3.2;
USR = 4.3.2.1;
USR = 4.4;
USR = 4.4.1;
USR = 4.4.2;
USR = 4.4.3;
USR = 4.4.3.1;
USR = 4.4.3.1(a);
USR = 4.4.3.1(b);
USR = 4.4.3.1(c);
USR = 4.4.3.2;
USR = 4.4.3.2(a);
USR = 4.4.3.2(b);
USR = 4.4.3.2(c);
USR = 4.4.3.2(d);
USR = 4.4.3.2(e);
USR = 4.4.3.2(f);
USR = 4.4.3.2(g);
USR = 4.4.3.2(h);
USR = 4.6;

2.4.5 Communicate Commercial Vehicle On-board Data to Vehicle Manager

Description:

Overview: This process shall be responsible for providing the communications interface through which the commercial vehicle manager (or commercial vehicle driver acting in the role of the manager) can access the data stored on-board a commercial vehicle. The process shall also support the exchange of unformatted messages between the commercial vehicle manager and driver, and the ability of the driver to send the on-board data to the manager as an unsolicited data flow.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO01

CVO08

CVO09

Data Flows:

Input Flows

cf_request_on_board_vehicle_data
cv_general_output_message
cv_on_board_data_output
cv_output_on_board_vehicle_data
cv_request_on_board_vehicle_data
cv_vehicle_location_request
vehicle_location_for_cv

Output Flows

cf_on_board_vehicle_data
cv_general_input_message
cv_on_board_data_required
cv_on_board_vehicle_data
cv_vehicle_location

User Service Requirements Mapped to Pspec:

USR = 4.2.3.1;
USR = 4.2.3.2;
USR = 4.2.3.4;
USR = 4.2.3.5;
USR = 4.3.2.1;
USR = 4.3.2.2;
USR = 4.4.3.2;

2.4.6 Provide Commercial Vehicle On-board Data Store Interface

Description:

Overview: This process shall be responsible for providing the interface through which data can be written to and read from the store of data that is held on-board a commercial vehicle. The data shall be provided by and on request from, other processes within the Manage Commercial Vehicles function that are on-board the vehicle.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO03

CVO05

CVO06

CVO07

Data Flows:

Input Flows

cv_inspection_data_update
cv_on_board_data_needed
cv_on_board_data_required
cv_on_board_data_update
cv_provide_credentials_data_for_inspections

Input/Output

cv_on_board_stored_data

Output Flows

cv_on_board_data_current_copy
cv_on_board_data_output

User Service Requirements Mapped to Pspec:

USR = 4.2.3.1;
USR = 4.2.3.2;
USR = 4.2.3.4;
USR = 4.2.3.5;
USR = 4.3.2.1;
USR = 4.3.2.2;
USR = 4.4.3.2;

2.5.1 Manage Commercial Vehicle Trips and Clearances

Description:

Overview: This process shall be responsible for the advance acquisition of electronic credentials and tax filing for commercial vehicles. The process will support the payment of the necessary taxes and duties that will enable a vehicle to be cleared through the credentials checks at the roadside checkstation facilities along its route, including those at border crossings. For this activity the process uses information about the vehicle's route provided by the commercial vehicle manager, or by the driver acting in that role when the vehicle is owned and operated by the driver. The actual payment activity and the subsequent notification of the roadside facilities along the route is carried out by other processes. Where the roadside facilities are outside the area served by the local ITS functions, the process requests that the necessary vehicle data is passed to the similar processes serving the appropriate areas.

In Subsystem:

Commercial Vehicle Administration

In Market Package:

CVO04

CVO05

Data Flows:

Input Flows

cf_enrollment_payment_request
cf_enrollment_request
cf_tax_audit_data
cv_check_credentials_response
cv_confirmed_enrollment
cv_enrollment_payment_request
cv_enrollment_request
cv_remote_enrollment_confirmation
cv_request_enrollment_data
cv_tax_and_credential_fees
cv_update_new_credentials_response

Output Flows

cf_enrollment_information
cf_enrollment_payment_confirmation
cv_check_credentials_request
cv_enrollment_information
cv_enrollment_list
cv_enrollment_payment_confirmation
cv_provide_enrollment_data
cv_remote_enrollment_request
cv_request_permits_and_duties_update
cv_update_new_credentials_request

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.6;
USR = 4.6.1;

2.5.2 Obtain Electronic Credential and Tax Filing Payment

Description:

Overview: This process shall be responsible for making payment for electronic credential and tax filing. The data on which the payment is based shall be that for a commercial vehicle's route as provided by the commercial vehicle manager or the commercial vehicle driver who is also the owner of the vehicle. The actual payment activity will be carried out by another process in the Provide Electronic Payment Services function.

In Subsystem:

Commercial Vehicle Administration

In Market Package:

CVO04

Data Flows:

Input Flows

cv_enrollment_list
cv_tax_and_credential_fees
financial_response

Output Flows

cv_confirmed_enrollment
financial_request

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.6;
USR = 4.6.1;

2.5.3 Update Permits and Duties Store

Description:

Overview: This process shall be responsible for receiving data from Government Administrators. This data comprises updates to the list of electronic credentials and tax filing required for a commercial vehicle to pass each roadside checkstation facility, plus carrier safety ratings for use in roadside safety inspections. These updates are both loaded into a store used by other process in the commercial vehicle administration facility.

In Subsystem:

Commercial Vehicle Administration

In Market Package:

CVO04

Data Flows:

Input Flows

cv_request_permits_and_duties_update
fga_carrier_safety_ratings
fga_roadside_facility_locations
fga_tax_and_credential_fees

Input/Output

cv_roadside_facility_locations

Output Flows

cv_roadside_facility_location
cv_safety_history_update
cv_tax_and_credential_fees
tga_request_fees_updates

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1;
USR = 4.1.1.2;
USR = 4.4.0;
USR = 4.4.1;
USR = 4.4.1(a);
USR = 4.4.1(b);
USR = 4.4.3;
USR = 4.4.3.2;

2.5.4 Communicate with Other Commercial Vehicle Administration System

Description:

Overview: This process shall be responsible for communicating with commercial vehicle administration facilities in ITS functions that serve areas outside that which is served by the local function. The communications supported by the process shall enable the local function to enroll commercial vehicles in other areas, and for those other areas to enroll their commercial vehicles in the local area. The process shall thus support the coordination and the determination of electronic credentials and tax filing across geographic and jurisdictional boundaries.

In Subsystem:

Commercial Vehicle Administration

In Market Package:

CVO03

CVO04

CVO04

CVO05

CVO07

CVO08

Data Flows:

Input Flows

cv_commit_remote_enrollment
cv_provide_enrollment_data
cv_remote_enrollment_request
focvas_commit_local_enrollment
focvas_data_table
focvas_enrollment_confirmation
focvas_enrollment_request
focvas_provide_data

Output Flows

cv_commit_local_enrollment
cv_remote_enrollment_confirmation
cv_request_enrollment_data
tocvas_commit_remote_enrollment
tocvas_data_table
tocvas_enrollment_confirmation
tocvas_enrollment_request
tocvas_provide_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1;
USR = 4.1.1.3;
USR = 4.6;
USR = 4.6.1;

2.5.5 Manage Commercial Vehicle Credentials and Enrollment

Description:

Overview: This process shall be responsible for enabling commercial vehicle managers and drivers (who are owners) to enroll the electronic credentials for their vehicles. This enrollment shall be achieved by loading the credentials data into a data store from which such data shall be downloaded to the commercial vehicle roadside checkstation and border crossing facilities by another process. When the roadside facility is located in the area not served by the local Manage Commercial Vehicles function, the process sends the data to another process that is responsible for communicating with a similar function in other geographic and/or jurisdictional areas. The process shall also be able to accept commercial vehicle enrollment requests from similar functions in other areas, query enforcement agency databases for outstanding prosecutions, and shall be able to respond to requests for information from authorized entities, such as insurance underwriters.

In Subsystem:

Commercial Vehicle Administration

In Market Package:

CVO04

CVO07

CVO08

Data Flows:

Input Flows

cv_check_credentials_request
cv_commit_local_enrollment
cv_roadside_facility_location
cv_safety_history
cv_update_new_credentials_request
fcvoir_request_for_information
fea_cv_enforcement_agency_response

Output Flows

cv_check_credentials_response
cv_commit_remote_enrollment
cv_credentials_enrollment_data
cv_database
cv_update_new_credentials_response
tcvoir_carrier_or_vehicle_information
tea_cv_request_for_information

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1;
USR = 4.1.1.8;
USR = 4.2;
USR = 4.2.2;
USR = 4.2.2.1;
USR = 4.2.2.4;
USR = 4.4.0;
USR = 4.4.1;
USR = 4.4.1(a);
USR = 4.4.1(b);
USR = 4.4.2;

2.5.6 Output Commercial Vehicle Enrollment Data to Roadside Facilities

Description:

Overview: This process shall be responsible for providing credentials, safety and border crossing data to commercial vehicle roadside checkstation and border crossing facilities. This data shall be output by the process periodically (e.g. daily) from an interrogation of the stores of safety history and credentials, and sent to the roadside facilities served by the local Manage Commercial Vehicles function. The process shall also provide selected credentials and safety data on request from the commercial vehicle inspectors at particular roadside checkstation facilities.

In Subsystem:

Commercial Vehicle Administration

In Market Package:

CVO03

CVO04

CVO04

CVO05

CVO05

CVO07

CVO08

Data Flows:

Input Flows

cv_credentials_information_request
cv_database
cv_facility_log
cv_safety_history_read
cv_safety_information_request

Output Flows

cv_border_database_update
cv_credentials_database_update
cv_credentials_information_response
cv_safety_database_update
cv_safety_information_response

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1;
USR = 4.1.1.8;
USR = 4.2;
USR = 4.2.2;
USR = 4.2.2.1;
USR = 4.2.2.5;
USR = 4.2.2.5(a);
USR = 4.2.2.5(b);
USR = 4.2.2.5(c);
USR = 4.2.2.5(d);
USR = 4.2.2.4;

2.5.7 Process Commercial Vehicle Violations

Description:

Overview: This process shall be responsible for sending details of commercial vehicle carriers and drivers that require prosecution to a process in the Manage Emergency Services function. The receiving process in that function will be responsible for sending the data to the appropriate law enforcement agency. This process shall obtain the data by periodically (e.g. daily) scanning the data in the log obtained from the commercial vehicle roadside checkstation facilities.

In Subsystem:

Commercial Vehicle Administration

In Market Package:

CVO04

Data Flows:

Input Flows

cv_facility_log

Output Flows

cv_violation_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.4;
USR = 4.4.3;
USR = 4.4.3.2;
USR = 4.2.0;
USR = 4.2.1;
USR = 4.2.2;

2.5.8 Process Data Received from Roadside Facilities

Description:

Overview: This process shall be responsible for the examination of the daily logs received periodically from the commercial vehicle checkstation and border crossing facilities. It shall also be responsible for the receipt in real time of data about commercial vehicles that have failed their safety inspections. The examination of the received data shall lead the process to update the local stores containing the facility logs and vehicle safety history. This process shall also issue quarterly reports for use by government administrators, send details of the activity at the roadside facility to the Plan System Deployment function. It shall also provide responses to requests from the commercial vehicle manager for reports of fleet activity through roadside facilities, either on-demand or as periodic summaries..

In Subsystem:

Commercial Vehicle Administration

In Market Package:

CVO04

CVO05

Data Flows:

Input Flows

cf_request_activity_report
cv_border_daily_log
cv_roadside_daily_log
cv_update_safety_problems_list

Input/Output

cv_facility_log

Output Flows

cf_periodic_activity_report
cf_roadside_activity_report
cv_daily_logs
cv_safety_history_write
tga_quarterly_reports

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.1;
USR = 4.1.1.8;
USR = 4.2;
USR = 4.2.2;
USR = 4.2.2.1;
USR = 4.2.2.4;
USR = 4.2.2.5;
USR = 4.4;
USR = 4.4.2;
USR = 4.4.2(a);
USR = 4.4.2(c);
USR = 4.4.2(d);

2.5.9 Manage Commercial Vehicle Archive Data

Description:

Overview: This process shall be responsible for processing request for archive data of commericial vehicle operations. This process shall receive operational data from the roadside check systems and administration and credentials data. This process shall receive and respond to requests from the Manage Archived Data process for either a catalog of the data contained with the commercial vehicle data stores or for the data itself. Additionally this process shall be able to produce sample products of the data available. As data is received into this process quality control metrics shall be assigned. The appropriate meta-data shall be generated and store along with the data. A catalog of the data shall be maintained to allow requesters to know what data is available from the archive store. The process shall run when a request for data is received from an external source, or when fresh data is received.

In Subsystem:

Commercial Vehicle Administration

In Market Package:

AD1

AD2

Data Flows:

Input Flows

cv_archive_request
cv_archive_status
cv_credentials_enrollment_data
cv_daily_logs

Input/Output

cv_data_archive

Output Flows

cv_archive_data

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.6;
USR = 7.1.3.1.6(a);
USR = 7.1.3.1.6(b);
USR = 7.1.3.1.6(c);
USR = 7.1.3.1.6(d);
USR = 7.1.3.1.6(e);
USR = 7.1.3.1.6(f);

2.6.1 Provide Commercial Vehicle Manager Tag Data Interface

Description:

Overview: This process shall be responsible for providing an interface through which the commercial vehicle manager can set up the data in the tag on-board a commercial vehicle. The data that the process enables the manager to write to the tag will be that which identifies the carrier, driver and vehicle. The process shall also enable the manager to read this data from the tag, but shall not enable the reading of any other data from the tag. Data provided by the manager shall also be sent by the process to the tag the process that manages electronic credentials and tax filing for use by the manager in future enrollments. The process shall support inputs from the commercial vehicle manager in both manual and audio form, and shall provide its outputs in audible and visual forms. It shall enable the visual output to be in hardcopy, or as a display.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO01

Data Flows:

Input Flows

cf_tag_data_store_output
cf_tag_initialization_data
fcvm_carrier_number
fcvm_driver_number
fcvm_request_tag_data_output
fcvm_trip_identity
fcvm_vehicle_number

Output Flows

cf_tag_data
cf_tag_data_store_request
cf_tag_data_store_write
tcvm_confirm_enrollment_data_stored
tcvm_output_tag_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.2;
USR = 4.1.2.2;

2.6.2 Transmit Commercial Vehicle Tag Data

Description:

Overview: This process shall be responsible for providing the output of the data that has been previously stored on-board a commercial vehicle's tag on request from a commercial vehicle roadside checkstation facility. The process shall also provide the current status of the lock tag, if one is attached to the vehicle's cargo. The data shall only be sent by the process to the commercial vehicle roadside checkstation or border crossing facility that made the request. The output mechanism used by the process shall be an implementation issue, but it could be by radio, beacon, or a visual mechanism, such as a bar code.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO03

CVO05

CVO06

CVO07

Data Flows:

Input Flows

cv_lock_tag_data
cv_on_board_border_record
cv_on_board_screening_record
cv_request_electronic_clearance_data
cv_tag_data_store_read

Output Flows

cv_electronic_clearance_data
cv_request_lock_tag_data
cv_tag_data_store_needed
cv_tag_data_store_update

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.2;
USR = 4.1.2.1;
USR = 4.1.2.2;

2.6.3 Provide Commercial Driver Tag Data Interface

Description:

Overview: This process shall be responsible for providing the interface through which the commercial vehicle driver can set up the data in an on-board vehicle unit (e.g. an electronic tag). In this instance the driver is assumed to be acting in the role of a commercial vehicle manager, and is thus likely to be the owner of the vehicle. The data the process enables the manager to write to the tag will be that which identifies the carrier, driver and vehicle. The process shall also enable the read this data from the tag, but shall not enable the manager to read any other data from the tag. The process shall support inputs from the commercial vehicle driver in both manual and audio form, and shall provide its outputs in audible and visual forms. It shall enable the visual output to be in hardcopy, or as a display.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO03

CVO05

CVO06

CVO07

Data Flows:

Input Flows

cv_tag_data_store_output
fcvd_carrier_number
fcvd_driver_number
fcvd_request_tag_data_output
fcvd_trip_identity
fcvd_vehicle_number

Output Flows

cv_tag_data_store_request
cv_tag_data_store_write
tcvd_confirm_data_stored
tcvd_output_tag_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.2;
USR = 4.1.2.2;

2.6.4 Provide Lock Tag Data Interface

Description:

Overview: This process shall be responsible for producing an output of the current status of a lock tag that is being carried by the cargo of a commercial vehicle. The process shall only produce the output in response to a request for data that is received from the other process on-board the vehicle that is responsible for communication with commercial vehicle roadside checkstation facilities. The actual output mechanism used by the process shall be an implementation issue, but it could be by radio or beacon.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO03

CVO05

CVO06

CVO07

Data Flows:

Input Flows

cv_request_lock_tag_data
fcv_lock_tag_data

Output Flows

cv_lock_tag_data
tcv_lock_tag_data_request

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.2;
USR = 4.1.2.2;

2.6.5 Manage Commercial Vehicle Tag Data Store

Description:

Overview: This process shall be responsible for managing the store of data that is held by a commercial vehicle's on-board tag. It shall manage all of the transactions that either write data to the store and read data from it, to ensure that the data retains its consistency. The process shall ensure that the commercial vehicle manager or driver can only read the data that they are enabled to write to the store, and that the store only contains data from the last two roadside checkstation facilities passed by the commercial vehicle.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO03

CVO05

CVO06

CVO07

Data Flows:

Input Flows

cf_tag_data_store_request
cf_tag_data_store_write
cv_tag_data_store_needed
cv_tag_data_store_request
cv_tag_data_store_update
cv_tag_data_store_write

Input/Output

cv_tag_data_store

Output Flows

cf_tag_data_store_output
cv_provide_credentials_data_for_inspections
cv_tag_data_store_output
cv_tag_data_store_read

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.1;
USR = 4.1.2;
USR = 4.1.2.1;
USR = 4.1.2.2;

2.7.1.1 Manage Terminal Access

Description:

Overview: This process shall be responsible for maintaining site security through the monitoring of entry and exit of commercial vehicles tothe Intermodal Terminal. In addition the process shall provide routing instructions within the Terminal for the correct location for pickup or delivery of an intermodal container.

Functional Requirements: None.

In Subsystem:

Intermodal Terminal

In Market Package:

CVO12

Data Flows:

Input Flows

container_location_information_for_driver
container_transfer_location_request
cv_vehicle_identification
entry_request
exit_request
release_container_to_driver

Output Flows

container_transfer_location
driver_credentials
entry_permission
exit_permission

User Service Requirements Mapped to Pspec:

USR = C5.5.2.4;
USR = C5.5.2.5;
USR = C5.5.2.7;

2.7.1.2 Manage Container Pickup and Delivery

Description:

Overview: This process shall be responsible for managing the pickup and delivery of containers. The process shall communicate with the Intermodal Fleet to coordinate pickup and delivery.

Functional Requirements: None.

In Subsystem:

Intermodal Terminal

In Market Package:

CVO12

Data Flows:

Input Flows

container_delivery_request
container_pickup_confirmation
release_status

Output Flows

container_delivery_confirmation
container_pickup_request

User Service Requirements Mapped to Pspec:

USR = C5.5.2.1;
USR = C5.5.2.2;
USR = C5.5.2.8;

2.7.1.3 Manage Container Release

Description:

Overview: This process shall be responsible for managing release of delivered intermodal containers. The process shall coordinate the release of the container with the Other Mode Shipper and with the Customs Agency. The process shall provide information on release status to Distribution and Logistics Management Providers and Fleet Managers.

Functional Requirements: None.

In Subsystem:

Intermodal Terminal

In Market Package:

CVO12

Data Flows:

Input Flows

container_location_information
driver_credentials
fca_container_release
fdlmp_container_availability_request
foifms_shipper_container_release
information_on_container_release_status

Output Flows

container_information_request
container_location_information_for_driver
release_container_to_driver
release_information
release_status
tca_container_arrival_information
tdlmp_container_availability

User Service Requirements Mapped to Pspec:

USR = C5.5.2.3;
USR = C5.5.2.8;

2.7.1.4 Provide Intermodal Terminal Operator I/F

Description:

Overview: This process shall provide the interface to the Intermodal Terminal Operator. The process shall accept requests for information from interested parties, and provide identification of the information to be released to the parties. It shall monitor the release status of containers.

Functional Requirements: None.

In Subsystem:

Intermodal Terminal

In Market Package:

CVO12

Data Flows:

Input Flows

container_information_request
intermodal_terminal_operator_inputs
release_information

Output Flows

information_on_container_release_status
intermodal_terminal_operator_data

User Service Requirements Mapped to Pspec:

USR = C5.5.2.4;

2.7.1.5 Provide Container Tracking

Description:

Overview: This process shall track the location of intermodal containers within the Intermodal Terminal. The process shall monitor the status of the container, including container integrity. The process shall provide this location and status to other processes in the Intermodal Terminal.

Functional Requirements: None.

In Subsystem:

Intermodal Terminal

In Market Package:

CVO12

Data Flows:

Input Flows

container_identification_for_terminal
container_location_for_terminal
container_status_for_terminal

Output Flows

container_location_information

User Service Requirements Mapped to Pspec:

USR = C5.5.2.4;
USR = C5.5.2.6;

2.7.2.1 Determine Container Status

Description:

Overview: This process shall be responsible for determining the status and/or location of an intermodal container. The process shall monitor parameters of the container (e.g. temperature, vibration, and humidity). The process shall monitor the status of the container seal. The process shall accept an input from separately monitored cargo within the container. The process shall allow output status only to a process in the Commercial Vehicle Subystem. In this case an independent location capability is not required. The process shall allow operation in a standalone mode, where status and location are transmitted directly to the Distribution and Logistics Management Provider. The process shall provide the ability to output the individual parameters, or to output a general status indication such as 'green'/'red'. The process shall have the ability to test its monitoring capability and provide an output on the status of its monitoring function.

Functional Requirements: None.

In Subsystem:

Intermodal Container

In Market Package:

CVO11

Data Flows:

Input Flows

cargo_measures
electronic_seal_data
From_Location_Data_Source

Output Flows

cargo_data
container_identification_for_terminal
container_information
container_location_for_terminal
container_monitoring_status
container_status
electronic_seal_data

User Service Requirements Mapped to Pspec:

USR = C5.5.1.1;
USR = C5.5.1.2;

2.7.2.2 Determine Cargo Status

Description:

Overview: This process shall be responsible for determining the status and/or location of cargo which is inside an intermodal container. The process shall monitor parameters of the cargo (e.g. temperature, vibration, and humidity). The process shall provide an input to an overall container status. In this case an independent location capability is not required. The process shall allow operation in a standalone mode, where status and location are transmitted directly to the Distribution and Logistics Management Provider. The process shall provide the ability to output the individual parameters, or to output a general status indication such as 'green'/'red'. The process shall have the ability to test its monitoring capability and provide an output on the status of its monitoring function.

Functional Requirements: None.

In Subsystem:

Intermodal Container

In Market Package:

CVO11

Data Flows:

Input Flows

From_Location_Data_Source

Output Flows

cargo_information
cargo_measures

User Service Requirements Mapped to Pspec:

USR = C5.5.1.1;
USR = C5.5.1.2;

2.7.2.3 Provide Cargo Status Interface

Description:

Overview: This process shall be responsible for providing the interface for direct query of cargo status or location by the Distribution and Logistics Management Provider. The process shall provide the ability to output the individual parameters, or to output a general status indication such as 'green'/'red'. The process shall have the ability to output the status of the cargo monitoring function (output of cargo monitoring self test).

Functional Requirements: None.

In Subsystem:

Intermodal Container

In Market Package:

CVO11

Data Flows:

Input Flows

cargo_information
cargo_monitoring_status
fdlmp_cargo_status_request

Output Flows

fdlmp_container_status_request
tdlmp_cargo_location

User Service Requirements Mapped to Pspec:

USR = C5.5.1.3;
USR = C5.5.1.4;

2.7.2.4 Provide Container Status Interface

Description:

Overview: This process shall be responsible for providing the interface for direct query of container status or location by the Distribution and Logistics Management Provider. The process shall provide the ability to output the individual parameters, or to output a general status indication such as 'green'/'red'. The process shall have the ability to output the status of the container monitoring function (output of container monitoring self test).

Functional Requirements: None.

In Subsystem:

Intermodal Container

In Market Package:

CVO11

Data Flows:

Input Flows

container_information
container_monitoring_status
fdlmp_container_status_request

Output Flows

tdlmp_container_status

User Service Requirements Mapped to Pspec:

USR = C5.5.1.3;
USR = C5.5.1.4;

2.7.2.5 Manage Customs Interface

Description:

Overview: This process shall be responsible for providing the interface for direct query of container seal status or manifest information by the Customs Agency. The process shall have the ability to output the status of the container seal monitoring function.

Functional Requirements: None.

In Subsystem:

Intermodal Container

In Market Package:

CVO12

Data Flows:

Input Flows

electronic_seal_data
fca_container_seal_status_request
fca_manifest_request

Output Flows

tca_container_seal_status
tca_manifest_data

User Service Requirements Mapped to Pspec:

USR = C5.5.2.3;

2.7.3.1 Provide Driver Interface for Intermodal Freight Dispatch

Description:

Overview: This process shall be responsible for providing the interface between the Fleet and Freight Management function and the Commercial Vehicle Driver for the assignment and routing of intermodal shipment pickup and delivery.

Functional Requirements: None.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO12

Data Flows:

Input Flows

container_transfer_location_information
intermodal_container_dispatch_assignment
intermodal_route

Output Flows

tcvd_container_transfer_location
tcvd_intermodal_container_transport_assignment
tcvd_intermodal_dispatch_route

User Service Requirements Mapped to Pspec:

USR = C5.5.2.1;
USR = C5.5.2.2;

2.7.3.2 Monitor Container Status

Description:

Overview: This process shall be responsible for determining the status of an intermodal container. The process shall monitor measures from the container systems, such as refrigeration, shock measurement, controlled atmosphere sensing, and electronic seal data (both an electronic manifest and the status of the electronic seal). The process shall provide the ability to output the individual system measures, or to output a general status operation readiness indication such as 'green'/'red'. The process shall also have the ability to output on the status of the monitoring equipment itself (self-test).

Functional Requirements: None.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO11

Data Flows:

Input Flows

cargo_data
container_status
intermodal_container_status_request

Output Flows

intermodal_container_monitoring_status
intermodal_container_status

User Service Requirements Mapped to Pspec:

USR = C5.5.1.1;
USR = C5.5.1.2;

2.7.3.3 Monitor Chassis Status

Description:

Overview: This process shall be responsible for determining the status of an intermodal chassis. The process shall monitor measures from the chassis systems, such as brakes, tires, and fasteners. The process shall monitor an overal operational readiness status of the chassis. The process shall provide the ability to output the individual system measures, or to output a general status operation readiness indication such as 'green'/'red'. The process shall have the ability to test its monitoring capability and provide an output on the status of its monitoring function.

Functional Requirements: None.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO11

Data Flows:

Input Flows

fic_chassis_data
fic_chassis_status
intermodal_chassis_status_request

Output Flows

intermodal_chassis_monitoring_status
intermodal_chassis_status

User Service Requirements Mapped to Pspec:

USR = C5.5.1.1;
USR = C5.5.1.2;

2.7.3.4 Manage On-Board Facility Access

Description:

Overview: This process shall be responsible for managing the electronic access to an Intermodal Terminal or Freight Consolidation Station. The process shall request and receive entry and exit permission to the facility.The process shall issue requests for guidance to the appropriate loading or unloading location within the Intermodal Terminal or Freight Consolidation Station.

Functional Requirements: None.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO12

Data Flows:

Input Flows

container_transfer_location
entry_permission
exit_permission
ffcs_entry_permission
ffcs_exit_permission
ffcs_freight_consolidation_location

Output Flows

container_transfer_location_information
container_transfer_location_request
cv_vehicle_identification
entry_request
exit_request
tfcs_commercial_vehicle_identification
tfcs_entry_request
tfcs_exit_request
tfcs_freight_consolidation_location_request

User Service Requirements Mapped to Pspec:

USR = C5.5.2.4;
USR = C5.5.2.5;

2.7.4.1 Monitor Intermodal Elements

Description:

Overview: This process shall be responsible for monitoring the status of the intermodal elements which are being transported by a vehicle of the fleet. The process shall receive as input from the on-board monitoring system(s) measures from the container systems as well as measures from the intermodal chassis. The process shall provide the ability to output the individual system measures, or to output a general status operation readiness indication such as 'green'/'red'. The process shall also have the ability to monitor the status of the monitoring equipment itself (self-test).

Functional Requirements: None.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO12

Data Flows:

Input Flows

intermodal_chassis_monitoring_status
intermodal_chassis_status
intermodal_container_monitoring_status
intermodal_container_status
intermodal_container_status_from_other_shipper

Output Flows

intermodal_chassis_status_request
intermodal_container_status_for_customer
intermodal_container_status_for_distribution
intermodal_container_status_for_fleet_manager
intermodal_container_status_for_other_shipper
intermodal_container_status_request

User Service Requirements Mapped to Pspec:

USR = C5.5.1.1;
USR = C5.5.1.2;

2.7.4.2 Provide Fleet Manager Interface for Intermodal

Description:

Overview: This process shall be responsible for providing the interface to the Commercial Vehicle Manager for Intermodal operations. The interface will handle the co-ordiantion of an intermodal container request, and the dispatch and routing of fleet assets to accomplish the shipment pickup and delivery and coordination with the Intermodal Terminal regarding delivery and pickup.

Functional Requirements: None.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO12

Data Flows:

Input Flows

container_delivery_confirmation
container_pickup_request
fcvm_intermodal_dispatch_assignment
intermodal_consignment_request
intermodal_container_status_for_fleet_manager

Output Flows

container_delivery_request
container_pickup_confirmation
intermodal_dispatch_assignment
tcvm_intermodal_consignment_request
tcvm_intermodal_container_status

User Service Requirements Mapped to Pspec:

USR = C5.5.2.4;

2.7.4.3 Manage Intermodal Customer Interface

Description:

Overview: This process shall be responsible for managing the interface to the Intermodal Customer. The interface shall handle the initial request and contracting for shipment. The process shall provide the ability for an Intermodal Customer to query the status of their shipment. The process shall respond with this status.

Functional Requirements: None.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO12

Data Flows:

Input Flows

fic_freight_consignment_request
fic_shipment_status_request
intermodal_container_shipment_status
intermodal_container_status_for_customer

Output Flows

intermodal_consignment_request
tic_consignment_contract
tic_container_delivery_confirmation
tic_shipment_status

User Service Requirements Mapped to Pspec:

USR = C5.5.1.3;
USR = C5.5.1.4;

2.7.4.4 Manage Other Intermodal FMS Interface

Description:

Overview: This process shall be responsible for managing the interface to the Other Intermodal FMS Systems. The process shall coordinate the pickup and delivery of an intermodal shipment with other shippers with which it must interface. The process shall provide the ability for an Other Intermodal FMS to query the status of their shipment. The process shall respond with this status.

Functional Requirements: None.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO12

Data Flows:

Input Flows

foifms_shipper_container_release
foifms_shipping_coordination
intermodal_container_status_for_other_shipper

Output Flows

intermodal_container_status_from_other_shipper
toifms_shipping_coordination

User Service Requirements Mapped to Pspec:

USR = C5.5.1.3;
USR = C5.5.1.4;

2.7.4.5 Manage Distribution and Logistics Management Provider Interface

Description:

Overview: This process shall be responsible for providing the interface to the Distribution and Logistics Management Provider (DLMP). The process shall accept requests from the DLMP for container status or location and provide the requested information. The process shall accept requests for shipments from the DLMP and shall provide confirmation of both pickup and delivery to the DLMP.

Functional Requirements: None.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO12

Data Flows:

Input Flows

fdlmp_container_location_query
fdlmp_container_pickup_request
fdlmp_container_status_query
intermodal_container_status_for_distribution
intermodal_container_status_from_manager

Output Flows

tdlmp_container_delivery_confirmation
tdlmp_container_location_response
tdlmp_container_pickup_confirmation
tdlmp_container_status_response

User Service Requirements Mapped to Pspec:

USR = C5.5.1.3;
USR = C5.5.1.4;

2.7.4.6 Manage Freight Consolidation Station Interface

Description:

Overview: This process shall be responsible for managing the interface to the Freight Consolidation Station. The process shall coordinate the pickup and delivery of an intermodal shipment to the station where part container shipments are made into full containers, or are unpacked.

Functional Requirements: None.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO12

Data Flows:

Input Flows

ffcs_container_delivery_confirmation
ffcs_container_pickup_request

Output Flows

container_pickup_and_delivery_status
tfcs_container_delivery_request
tfcs_container_pickup_confirmation

User Service Requirements Mapped to Pspec:

USR = C5.5.1.3;
USR = C5.5.1.4;

3.1.1 Produce Collision and Crash Avoidance Data

Description:

Overview: This process shall be responsible for sensing and evaluating the likelihood of a collision between two vehicles or a vehicle and a stationary object. The process shall base its detection on input from two other processes. One of these processes shall be that which continuously processes sensor inputs on-board the vehicle and the second shall be that which detects collision situations at intersections. When either event is detected this process shall output the appropriate messages to another process in the vehicle to warn the driver. If the vehicle is suitably equipped, the process shall initiate the deployment of crash restraint devices in advance of the collision and/or generate data to initiate direct operation of the vehicle to take evasive maneuvers.

In Subsystem:

Vehicle

In Market Package:

AVSS03

AVSS04

AVSS05

AVSS06

AVSS10

Data Flows:

Input Flows

collision_data
intersection_collision_avoidance_data

Output Flows

position_warnings
vehicle_action_requests

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.1;
USR = 6.1.0;
USR = 6.1.1;
USR = 6.1.1.2;
USR = 6.1.1.2.1;
USR = 6.1.2;
USR = 6.1.2.1;
USR = 6.1.2.2;
USR = 6.1.2.1.1;
USR = 6.1.2.2.1;
USR = 6.1.3;
USR = 6.1.3.1;
USR = 6.1.3.1.1;
USR = 6.1.3.2;
USR = 6.1.3.2.1;
USR = 6.1.3.3;
USR = 6.1.3.3.1;
USR = 6.2;
USR = 6.2.0;
USR = 6.2.1;
USR = 6.2.1.1;
USR = 6.2.1.1.1;
USR = 6.2.1.2;
USR = 6.2.1.2.2;
USR = 6.2.2;

USR = 6.2.2.1;
USR = 6.2.2.1.1;
USR = 6.3;
USR = 6.3.0;
USR = 6.3.1;
USR = 6.3.1.1;
USR = 6.3.2;
USR = 6.3.2.1;
USR = 6.5;
USR = 6.5.1;
USR = 6.5.1.1;
USR = 6.5.1.1.2;
USR = 6.6;
USR = 6.6.1;

3.1.2 Carry-out Safety Analysis

Description:

Overview: This process shall be responsible for producing safety warnings for display to the driver and output to the vehicle control processes. The process shall base its output on input from another process in the vehicle that is analyzing inputs to sensors. When data about a safety situation is received, the process shall output the appropriate messages to another process in the vehicle to warn the driver. If the vehicle is so equipped, the process shall send data to the process in the vehicle responsible for its control.

In Subsystem:

Vehicle

In Market Package:

AVSS01

AVSS02

Data Flows:

Input Flows

safety_data

Output Flows

safety_warnings
vehicle_and_driver_safety_status

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.5;
USR = 6.5.1;
USR = 6.5.1.1;
USR = 6.5.1.1.1;
USR = 6.5.1.1.2;
USR = 6.5.1.1.3;
USR = 6.1.2;
USR = 6.1.2.1;
USR = 6.5.2.1.1;
USR = 6.5.2.1.2;
USR = 6.0;
USR = 6.5;
USR = 6.5.3;
USR = 6.5.3.1;
USR = 6.5.3.1.1;
USR = 6.5.3.1.2;
USR = 6.7;
USR = 6.7.0;
USR = 6.7.1;
USR = 6.7.1.3;
USR = 6.7.1.3.1;

3.1.3 Process Vehicle On-board Data

Description:

Overview: This process shall be responsible for processing data received as input to sensors located on-board a vehicle. The process shall continuously analyze these inputs and produce data from which safety and/or position warnings and actions can be produced by another process. It shall also analyze the data to check for hazardous roadside conditions such as flooding, ice, snow, etc. and if detected shall output this data to processes in the Manage Traffic function.

In Subsystem:

Vehicle

In Market Package:

ATMS12

AVSS01

AVSS02

AVSS03

AVSS04

AVSS05

AVSS06

AVSS08

AVSS10

AVSS11

Data Flows:

Input Flows

fbv_diagnostics_data
fbv_driver_safety_status
fbv_vehicle_attitude_data
fbv_vehicle_motion_data
fbv_vehicle_proximity_data
fbv_vehicle_safety_status
fbv_vehicle_security_status
fre_roadside_data
From_Potential_Obstacles
From_Roadway

Output Flows

collision_data
safety_data
vehicle_smart_probe_data
vehicle_status_details_for_broadcast
vehicle_status_details_for_driver_security
vehicle_status_details_for_emergencies
vehicle_status_details_for_emissions

User Service Requirements Mapped to Pspec:

USR = 1.6;
USR = 1.6.4;
USR = 1.6.4(a);
USR = 6.0;
USR = 6.1;
USR = 6.1.0;
USR = 6.1.1;
USR = 6.1.1.1;
USR = 6.1.1.1.1;
USR = 6.1.1.1.1.1;
USR = 6.1.1.1.1.2;
USR = 6.1.1.1.2;
USR = 6.1.1.2;
USR = 6.1.1.2.1;
USR = 6.1.1.3;
USR = 6.1.1.3.1;
USR = 6.1.2;
USR = 6.1.2.1;
USR = 6.1.2.1.1;
USR = 6.1.2.2.1;
USR = 6.1.2.3.1;
USR = 6.2;
USR = 6.2.0;
USR = 6.2.1;
USR = 6.2.1.1;
USR = 6.2.1.1.1;
USR = 6.2.1.2;
USR = 6.2.1.2.2;
USR = 6.2.1.3;
USR = 6.2.1.3.1;
USR = 6.2.2;
USR = 6.2.2.1;
USR = 6.2.2.1.1;
USR = 6.2.2.2;
USR = 6.2.2.2.1;
USR = 6.2.2.3;
USR = 6.2.2.3.1;
USR = 6.3;
USR = 6.3.0;
USR = 6.3.1;
USR = 6.3.1.1;
USR = 6.3.2;
USR = 6.3.2.1;
USR = 6.3.3;
USR = 6.3.3.1;
USR = 6.5;
USR = 6.5.1;
USR = 6.5.1.1;
USR = 6.5.1.1.1;
USR = 6.5.1.1.2;
USR = 6.5.1.1.3;
USR = 6.5.2;
USR = 6.5.2.1;
USR = 6.5.2.1.1;
USR = 6.5.2.1.2;
USR = 6.5.3;
USR = 6.5.3.1;
USR = 6.5.3.1.1;
USR = 6.6;
USR = 6.6.1;
USR = 6.6.1.1;

USR = 6.6.1.2;
USR = 6.7;
USR = 6.7.2;

3.2.1 Provide Driver Interface

Description:

Overview: This process shall be responsible for providing an interface through which a vehicle driver can initiate, monitor and terminate automatic control of the vehicle. The output that any of these actions generates in terms of messages to the driver shall be sent by this process to another process that is in the Provide Driver and Traveler Services function and in the vehicle. The driver inputs shall be received by this process from another process that is also in the Provide Driver and Traveler Services function and in the vehicle.

In Subsystem:

Vehicle

In Market Package:

AVSS08

AVSS09

AVSS11

Data Flows:

Input Flows

ahs_status
control_status
vehicle_control_request

Output Flows

driver_ahs_input
driver_input
vehicle_control_status

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.6;
USR = 6.6.0;
USR = 6.6.1;
USR = 6.6.1.1;
USR = 6.6.1.2;

3.2.2 Provide AHS Control

Description:

Overview: This process shall be responsible for providing the facility that enables vehicles to operate in automatic highway system (ahs) lanes. This mode of operation shall only be initiated by the process when a request is received from the driver via other processes in the vehicle. The first action of the process must be to send data to the process that provides the ahs check-in facility. If a positive response is received from that process, i.e. the vehicle's check in is accepted, then the process shall enable ahs operation by sending the data to the vehicle control processes. Once the vehicle is in ahs operation, the process shall continuously monitor for an input from the driver that cancels ahs mode, and when this is received send mode canceling data to the vehicle control processes. Similarly the process shall also continuously monitor input from the process analyzing vehicle condition and the vehicle's presence on an ahs lane. The process shall send mode canceling data to the vehicle control processes, if the condition does not support ahs lane operation, or the vehicle is no longer on an ahs lane.

In Subsystem:

Vehicle

In Market Package:

AVSS11

Data Flows:

Input Flows

ahs_check_response
ahs_route
ahs_vehicle_data
driver_ahs_input
platoon_status

Output Flows

ahs_control_data_update
ahs_route_data
ahs_route_request
ahs_status
ahs_vehicle_condition
platoon_action

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.7;
USR = 6.7.1;
USR = 6.7.1.1;
USR = 6.7.1.1.1;
USR = 6.7.1.1.2;
USR = 6.7.1.1.3;
USR = 6.7.1.2;
USR = 6.7.1.2.1;
USR = 6.7.1.2.3;
USR = 6.7.1.3;
USR = 6.7.1.3.2;
USR = 6.7.2;
USR = 6.7.2.1;
USR = 6.7.2.2;

3.2.3.1 Provide Command Interface

Description:

Overview: This process shall be responsible for providing the interface through which all driver commands are passed to the correct processes in the vehicle for action. The process shall also pass all messages about vehicle control status on to another process in the vehicle for output to the driver. It shall also monitor the health of the other in-vehicle processes involved in automatic vehicle control. This process shall take the appropriate mode canceling action when any failures are detected in these processes.

In Subsystem:

Vehicle

In Market Package:

AVSS08

AVSS09

Data Flows:

Input Flows

driver_input
feedback_actuator_status
feedback_platoon_status
feedback_sensor_status
feedback_servo_status

Output Flows

control_status
driver_commands
driver_manual_input
driver_selection

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.7;
USR = 6.7.1;
USR = 6.7.1.2;
USR = 6.7.1.2.1;
USR = 6.7.1.2.2;
USR = 6.7.1.2.3;
USR = 6.7.2;
USR = 6.7.2.1;

3.2.3.2 Manage Platoon Following

Description:

Overview: This process shall be responsible for providing the facility for the automatic control of vehicles to be extended to cover the platooning of vehicles. The process shall enable vehicles to follow each other very closely (inches apart) in a platoon, responding to changes in speed and direction of the lead vehicle. The process shall monitor data from other vehicles in the platoon received via another process, and shall also send data about itself to the same process for communication to other platoon vehicles. If the data received from the process shows that the vehicle has been left on its own, i.e. there are no other vehicles in front or behind, the process shall send data to another process in the vehicle to increase speed and catch up with any platoon that may be ahead. The process shall only allow the vehicle to join or continue running in a platoon if it and/or the driver are considered to be in a safe condition, using data received from other processes in the vehicle.

In Subsystem:

Vehicle

In Market Package:

AVSS11

Data Flows:

Input Flows

data_from_front_vehicle
data_from_rear_vehicle
driver_selection
manual_input_received
platoon_action
vehicle_and_driver_safety_status
vehicle_control_data

Output Flows

data_to_front_vehicle
data_to_rear_vehicle
feedback_platoon_status
platoon_following_commands
platoon_status
platooning_selected

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.1;
USR = 6.1.1;
USR = 6.1.1.1;
USR = 6.1.1.1.1;
USR = 6.1.1.1.2.1;
USR = 6.1.1.1.2.2;
USR = 6.7;
USR = 6.7.1;
USR = 6.7.1.2;
USR = 6.7.1.2.2;
USR = 6.7.1.2.3;
USR = 6.7.2;
USR = 6.7.2.3;

3.2.3.3 Process data for Vehicle Actuators

Description:

Overview: This process shall be responsible for providing the interface between other automatic vehicle control process and the actuators which actually change the vehicle's controls. The process shall both implement commands and monitor the operation of the actuators to check that they only move when requested. If they move for any other reason, e.g. the driver has touched the vehicle controls, the process shall disable automatic operation. The process shall perform its own built-in self test (BIST) analysis. It shall report any errors that this shows to another process in the vehicle and shall cease to accept further requests to change the vehicle's actuators.

In Subsystem:

Vehicle

In Market Package:

AVSS08

AVSS09

AVSS11

Data Flows:

Input Flows

actuator_commands
driver_manual_input
fbv_brake_servo_response
fbv_steering_servo_response
fbv_throttle_servo_response
platooning_selected
vehicle_action_requests

Output Flows

feedback_actuator_response
feedback_actuator_status
manual_input_received
tbv_change_brake_setting
tbv_change_direction
tbv_change_throttle_setting
tbv_deploy_crash_restraints
tbv_steer_left
tbv_steer_right
tbv_steer_straight

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.1;
USR = 6.1.1;
USR = 6.1.1.1;
USR = 6.1.1.1.1;
USR = 6.1.1.1.2;
USR = 6.1.1.1.2.1;
USR = 6.1.1.1.2.2;
USR = 6.1.1.1.3;
USR = 6.1.1.1.3.1;
USR = 6.1.1.3;
USR = 6.1.1.3.1;
USR = 6.1.2;
USR = 6.1.2.3;
USR = 6.1.2.3.1;
USR = 6.1.3;
USR = 6.1.3.3;

USR = 6.1.3.3.1;
USR = 6.2;
USR = 6.2.1;
USR = 6.2.1.3;
USR = 6.2.1.3.1;
USR = 6.2.2;
USR = 6.2.2.3;
USR = 6.2.2.3.1;
USR = 6.3;
USR = 6.3.3;
USR = 6.3.3.1;
USR = 6.6;
USR = 6.6.1;
USR = 6.6.1.2;
USR = 6.7;
USR = 6.7.1;
USR = 6.7.1.2;
USR = 6.7.1.2.3;
USR = 6.7.2;
USR = 6.7.2.1;

3.2.3.4.1 Provide Speed Servo Control

Description:

Overview: This process shall be responsible for providing data which enables the vehicle's throttle to be regulated in such a way that a desired vehicle speed is maintained. The process shall enable the throttle to be overridden temporarily in order to maintain a desired headway between the vehicle and others in a platoon. The data that actually changes the throttle's position shall be sent to the process that provides data to in-vehicle actuators. The process shall perform its own built-in self test (BIST) analysis. It shall report any errors that this shows to another process in the vehicle and shall cease to accept further requests to change the vehicle's throttle position.

In Subsystem:

Vehicle

In Market Package:

AVSS08

Data Flows:

Input Flows

manual_throttle_input_detected
override_throttle
platoon_speed_servo_override
select_speed
speed
vehicle_speed_control_data

Output Flows

feedback_speed_servo_status
throttle_commands

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.1;
USR = 6.1.1;
USR = 6.1.1.1;
USR = 6.1.1.1.1;
USR = 6.1.1.1.2;
USR = 6.1.1.1.2.1;
USR = 6.1.1.1.2.2;
USR = 6.1.1.1.3;
USR = 6.1.1.1.3.1;
USR = 6.7;
USR = 6.7.1;
USR = 6.7.1.2;
USR = 6.7.1.2.3;
USR = 6.7.2;
USR = 6.7.2.3;

3.2.3.4.2 Provide Headway Servo Control

Description:

Overview: This process shall be responsible for providing data which enables the vehicle's brake and throttle to be regulated in such a way that its headway, i.e. the distance between it and the vehicle in front, is maintained. The process shall support the brake movements that either maintain the vehicle's headway for normal operation, or hold it at the value used in platoon following, whether on or off automated highway system (ahs) lanes. The process shall perform its own built-in self test (BIST) analysis. It shall report any errors that this shows to another process in the vehicle and shall cease to accept further requests to change the vehicle's brake setting.

In Subsystem:

Vehicle

In Market Package:

AVSS08

Data Flows:

Input Flows

headway
manual_brake_input_detected
platoon_headway_servo_override
select_headway
vehicle_headway_control_data

Output Flows

brake_commands
feedback_headway_servo_status
override_throttle

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.1;
USR = 6.1.1;
USR = 6.1.1.1;
USR = 6.1.1.1.1;
USR = 6.1.1.1.2;
USR = 6.1.1.1.2.1;
USR = 6.1.1.1.2.2;
USR = 6.1.1.1.3;
USR = 6.1.1.1.3.1;
USR = 6.7;
USR = 6.7.1;
USR = 6.7.1.2;
USR = 6.7.1.2.3;
USR = 6.7.2;
USR = 6.7.2.3;

3.2.3.4.3 Provide Lane Servo Control

Description:

Overview: This process shall be responsible for providing the data which enables the vehicle's steering to be adjusted so that it maintains a position that is in the middle of its current lane. The process shall enable this to be temporarily overridden as a result of action being taken by other processes to change lanes. The process shall perform its own built-in self test (BIST) analysis. It shall report any errors that this shows to another process in the vehicle and shall cease to accept further requests to change the vehicle's throttle position.

In Subsystem:

Vehicle

In Market Package:

AVSS09

Data Flows:

Input Flows

lane_deviation
manual_steering_input_detected
override_lane_hold
platoon_lane_servo_override
select_lane_hold

Output Flows

feedback_lane_servo_status
lane_steering_commands

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.7;
USR = 6.7.1;
USR = 6.7.1.2;
USR = 6.7.1.2.3;
USR = 6.7.2;
USR = 6.7.2.3;

3.2.3.4.4 Provide Change Lane Servo Control

Description:

Overview: This process shall be responsible for providing the data which enables the vehicle's steering to be adjusted so that it will move either left or right from one lane to another. The process shall enable this to temporarily override the lane center holding facility available from another process in the vehicle. The process shall perform its own built-in self test (BIST) analysis. It shall report any errors that this shows to another process in the vehicle and shall cease to accept further requests to change the vehicle's throttle position.

In Subsystem:

Vehicle

In Market Package:

AVSS09

Data Flows:

Input Flows

manual_steering_input_detected
platoon_change_lane_servo_override

Output Flows

feedback_change_lane_servo_status
override_lane_hold
steering_commands

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.7;
USR = 6.7.1;
USR = 6.7.1.2;
USR = 6.7.1.2.3;
USR = 6.7.2;
USR = 6.7.2.3;

3.2.3.4.5 Provide Vehicle Control Data Interface

Description:

Overview: This process shall be responsible for providing a communications and data processing interface between processes in the Provide Vehicle Control and Monitoring function. These processes shall comprise those responsible for controlling individual functions, e.g. throttle, brake, etc., and those that interface to actuators and those that monitor vehicle operation.

In Subsystem:

Vehicle

In Market Package:

AVSS08

AVSS09

Data Flows:

Input Flows

ahs_control_data_update
brake_commands
driver_commands
feedback_actuator_response
feedback_change_lane_servo_status
feedback_headway_servo_status
feedback_lane_servo_status
feedback_speed_servo_status
lane_steering_commands
platoon_following_commands
sensor_data
steering_commands
throttle_commands
vehicle_control_data_store

Output Flows

actuator_commands
feedback_servo_status
headway
lane_deviation
manual_brake_input_detected
manual_steering_input_detected
manual_throttle_input_detected
platoon_change_lane_servo_override
platoon_headway_servo_override
platoon_lane_servo_override
platoon_speed_servo_override
select_headway
select_lane_hold
select_speed
speed
vehicle_control_data
vehicle_headway_control_data
vehicle_speed_control_data

User Service Requirements Mapped to Pspec:

USR = 6.1.0;
USR = 6.1.1;
USR = 6.1.1.1;
USR = 6.1.1.1.1;
USR = 6.1.1.1.2;
USR = 6.7;
USR = 6.7.1;
USR = 6.7.1.2;
USR = 6.7.1.2.3;

3.2.3.5 Process Vehicle Sensor Data

Description:

Overview: This process shall be responsible for providing the facility to decode the input being sent to on-board vehicle sensors. The process shall support inputs to those sensors that monitor conditions both on-board the vehicle and in the way the vehicle relates to its surroundings. The data produced by the process shall be sent to another process which shall determine if any action is required.

In Subsystem:

Vehicle

In Market Package:

ATMS02

AVSS03

AVSS04

AVSS05

AVSS06

Data Flows:

Input Flows

fbv_vehicle_headway
fbv_vehicle_lane_position
fbv_vehicle_on_ahs_lane
fbv_vehicle_speed

Output Flows

feedback_sensor_status
sensor_data

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.7;
USR = 6.7.1;
USR = 6.7.1.1;
USR = 6.7.1.1.3;
USR = 6.7.1.2;
USR = 6.7.1.2.1;
USR = 6.7.1.2.3;
USR = 6.7.1.3;
USR = 6.7.1.3.2;
USR = 6.7.2;
USR = 6.7.2.1;
USR = 6.7.2.3;

3.2.3.6 Communicate with other Platoon Vehicles

Description:

Overview: This process shall be responsible for communicating with the other vehicles that are in a platoon. The process shall support communications with the platoon vehicles that are both immediately in front of and behind the vehicle in which it operates. The passing of data in both directions, i.e. both to and from the vehicles, shall be supported by the process.

In Subsystem:

Vehicle

In Market Package:

AVSS11

Data Flows:

Input Flows

data_to_front_vehicle
data_to_rear_vehicle
From_Other_Vehicle

Output Flows

data_from_front_vehicle
data_from_rear_vehicle
To_Other_Vehicle

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.7;
USR = 6.7.1;
USR = 6.7.1.1;
USR = 6.7.1.1.3;
USR = 6.7.1.2;
USR = 6.7.1.2.1;
USR = 6.7.1.2.3;
USR = 6.7.1.3;
USR = 6.7.1.3.2;
USR = 6.7.2;
USR = 6.7.2.1;
USR = 6.7.2.3;

3.2.4 Process Sensor Data for AHS input

Description:

Overview: This process shall be responsible for analyzing the input from the vehicle that provides information about its condition and that it is on an automatic highway system (ahs) lane. The process shall continuously analyze this data and provide output to the process that provides ahs control.

In Subsystem:

Vehicle

In Market Package:

AVSS11

Data Flows:

Input Flows

fbv_vehicle_condition
fbv_vehicle_on_ahs_lane

Output Flows

ahs_vehicle_data

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.7;
USR = 6.7.1;
USR = 6.7.1.1;
USR = 6.7.1.1.3;
USR = 6.7.1.2;
USR = 6.7.1.2.1;
USR = 6.7.1.2.3;
USR = 6.7.1.3;
USR = 6.7.1.3.2;
USR = 6.7.2;
USR = 6.7.2.1;
USR = 6.7.2.3;

3.2.5 Check Vehicle for AHS eligibility

Description:

Overview: This process shall be responsible for checking that vehicles are eligible for using the automated highway system (ahs) lanes on a highway. The process shall decide whether or not the vehicle is suitable for has operation by checking locally stored data that has been provided by a process in the Manage Traffic function, against data from the vehicle provided through the check request by a process on-board the vehicle. The process shall send the results of the check to the process on-board the vehicle that requested the ahs check-in. The vehicles that are successfully checked-in shall also be down loaded with ahs control data from this process.

In Subsystem:

Roadway Subsystem

In Market Package:

AVSS11

Data Flows:

Input Flows

ahs_control_information
ahs_vehicle_condition

Input/Output

ahs_vehicle_checking_parameters

Output Flows

ahs_check_response
ahs_checking_data

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.7;
USR = 6.7.0;
USR = 6.7.1;
USR = 6.7.1.1;
USR = 6.7.1.1.2;

3.2.6 Manage AHS Check-in and Check-out

Description:

Overview: This process shall be responsible for managing the checking in and checking out of suitably equipped vehicles requesting to use automated highway system (ahs) lanes. The process shall provide the special vehicle control parameters needed for ahs operation to the process that manages ahs check-in and collect data on vehicles that request check-in and check-out from that process. This process shall send a record of all check-in and check-out transactions regardless of whether they are successful or not, to the process responsible for managing ahs operational data.

In Subsystem:

Roadway Subsystem

In Market Package:

AVSS11

Data Flows:

Input Flows

ahs_checking_data
ahs_control_data_changes
ahs_route_data

Output Flows

ahs_checking_details
ahs_control_information

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.7;
USR = 6.7.0;
USR = 6.7.1;
USR = 6.7.1.1;
USR = 6.7.1.1.1;
USR = 6.7.1.1.2;
USR = 6.7.1.1.3;

3.2.7 Manage AHS Operations

Description:

Overview: This process shall be responsible for recording data about vehicles that have requested check-in and check-out for the use of the automated highway system (ahs) lanes, and for receiving ahs control parameters from a process in the Manage Traffic function. The process shall provide a process at the roadside with the vehicle control parameters needed for ahs operation. The process shall keep a log of all ahs check-in and check-out transactions received from the roadside process regardless of whether they are successful or not, and periodically pass this data on to the Plan System Deployment function.

In Subsystem:

Traffic Management

In Market Package:

AVSS11

Data Flows:

Input Flows

ahs_checking_details
ahs_control_data

Input/Output

ahs_usage_data

Output Flows

ahs_control_data_changes
ahs_operational_data

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.7;
USR = 6.7.0;
USR = 6.7.1;
USR = 6.7.1.1;
USR = 6.7.1.1.1;
USR = 6.7.1.1.2;
USR = 6.7.1.1.3;

3.3.1 Provide Cargo Data for Incident Notification

Description:

Overview: This process shall be responsible for providing data about a commercial vehicle's cargo in the event that the vehicle is involved in some type of emergency. The process shall produce the output on request from another process in the vehicle regardless of whether the cargo has itself suffered from any damage. The cargo data being provided by the process shall cover all types, regardless of whether or not they are classified as HAZMAT cargoes.

In Subsystem:

Commercial Vehicle Subsystem

In Market Package:

CVO02

CVO08

CVO10

Data Flows:

Input Flows

cargo_data_request
fcv_cargo_data
fcv_cargo_safety_status

Output Flows

processed_cargo_data

User Service Requirements Mapped to Pspec:

USR = 4.0;
USR = 4.5;
USR = 4.5.0;
USR = 4.5.1;
USR = 4.5.1;
USR = 4.5.1.1;
USR = 4.5.1.2;
USR = 4.5.1.2(a);
USR = 4.5.1.2(b);
USR = 4.5.1.2(c);
USR = 5.0;
USR = 5.1;
USR = 5.1.2;
USR = 5.1.2.1;
USR = 5.1.2.1.2;
USR = 5.1.2.2;
USR = 5.1.2.2(c);

3.3.2 Provide Communications Function

Description:

Overview: This process shall be responsible for sending messages it receives from other processes in this facility to the Manage Emergency Services function. It shall also be responsible for passing on the resulting response to the driver via processes in the Provide Driver and Traveler Services function. this process is also capable of receiving requests for additional data from the Manage Emergency Services function and transmitting follow-up details. this process can also receive commands related to the vehicle's security system from the Manage Emergency Services function and forward the commands to the vehicle's security system.

In Subsystem:

Vehicle

In Market Package:

CVO10

EM3

Data Flows:

Input Flows

emergency_data_request
emergency_request_vehicle_acknowledge
vehicle_emergency_request

Output Flows

emergency_message_auto_output
emergency_request_vehicle_details
tbv_vehicle_security_system_commands

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.1;
USR = 5.1.1;
USR = 5.1.1.3;

3.3.3 Build Automatic Collision Notification Message

Description:

Overview: This process shall be responsible for preparing and submitting data for transmission to the Manage Emergency Services function. The data shall be sent by this process when an emergency situation is detected by analyzing inputs from the vehicle. This process shall produce its outputs regardless of any action by the driver and shall be designed to be as the result of a crash which may have prevented the driver from initiating the emergency request personally.

In Subsystem:

Vehicle

In Market Package:

CVO10

EM3

Data Flows:

Input Flows

fbv_crash_sensor_data
processed_cargo_data
vehicle_identity_for_collision_notification_store
vehicle_location_for_incidents
vehicle_status_details_for_emergencies

Output Flows

cargo_data_request
vehicle_emergency_request

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.1;
USR = 5.1.1;
USR = 5.1.1.4;
USR = 5.1.2;
USR = 5.1.2.1;
USR = 5.1.2.1.1;
USR = 5.1.2.1.2;
USR = 5.1.2.2;
USR = 5.1.2.2(a);

3.4 Enhance Driver's Vision

Description:

Overview: This process shall be responsible for providing data from which a continuously updated display showing an enhanced version of the driver's vision. The process shall produce the data for this display using inputs to sensors mounted on the vehicle. It shall operate at all times and shall send its output to another process for integration with other messages for the driver.

In Subsystem:

Vehicle

In Market Package:

AVSS07

Data Flows:

Input Flows

From_Roadway

Output Flows

vision_data

User Service Requirements Mapped to Pspec:

USR = 6.0;
USR = 6.4;
USR = 6.4.0;
USR = 6.4.1;

4.1.1 Process Transit Vehicle Sensor Trip Data

Description:

Overview: This process shall collect and process data available to sensors on-board transit vehicles. This data shall be sent by this process to other processes on-board the transit vehicle and elsewhere in the Manage Traffic function for use in determining vehicle schedule deviations and for storage as operations data.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS1

APTS6

Data Flows:

Input Flows

ftv_vehicle_trip_data

Output Flows

transit_vehicle_arrival_times
transit_vehicle_collected_trip_data
transit_vehicle_on_board_data

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.1;
USR = 2.1.1.1;
USR = 2.1.3;
USR = 2.1.3.1;
USR = 2.1.3.1.1;

4.1.2.1 Determine Transit Vehicle Deviation and ETA

Description:

Overview: This process shall determine the schedule deviation and estimated times of arrival (ETA) at transit stops of a transit vehicle. The data shall be sent by this process to other processes in the Manage Transit function for use in calculating corrective instructions for output to the transit vehicle drivers, for use in calculation of a much wider return to schedule strategy where more than one vehicle and/or service is involved, and for storage as transit vehicle operational data. This process shall also send the data to the transit driver interface process, so that the driver is aware of the actual schedule deviation. This output shall be set to zero (no deviation) when that condition occurs, even when it has followed a period of deviation from schedule.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS2

Data Flows:

Input Flows

transit_services_for_eta
transit_vehicle_arrival_times
transit_vehicle_location_for_eta

Output Flows

transit_services_for_eta_request
transit_vehicle_deviation_data
transit_vehicle_deviations
transit_vehicle_deviations_from_schedule
transit_vehicle_eta
transit_vehicle_eta_for_advisory
transit_vehicle_schedule_deviation

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.1;
USR = 2.1.1.2;
USR = 2.1.1.2.1;
USR = 2.1.1.2.1.1;
USR = 2.1.1.2.4;

4.1.2.2 Determine Transit Vehicle Corrective Instructions

Description:

Overview: This process shall generate outputs that enable a transit vehicle schedule deviation to be corrected. The process shall derive its outputs from data received from another process in the Manage Traffic function. The outputs produced by the process shall consist of corrective instructions for output to the transit vehicle driver by a process on-board the vehicle, and preemption requests for traffic signal controllers at road and freeway intersections. The process shall only produce this output when another process has determined that deviation is small, or the transit vehicle is operating in an urban area. In all other conditions, the process shall provide an output that shows that there are no corrective instructions.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS2

Data Flows:

Input Flows

approved_corrective_plan
transit_services_for_corrections
transit_vehicle_deviations

Output Flows

transit_vehicle_arrival_conditions
transit_vehicle_corrective_instructions
transit_vehicle_preemption_request

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.1;
USR = 2.1.1.2;
USR = 2.1.1.2.1;
USR = 2.1.1.2.1.4;
USR = 2.1.1.2.2;
USR = 2.1.1.2.4;

4.1.2.3 Provide Transit Vehicle Driver Interface

Description:

Overview: This process shall provide a schedule correction interface for the transit driver in the transit vehicle. The interface shall provide data to the driver about how far the vehicle is from its schedule and what corrective action the driver must take. The data shall be received by the process from other processes in the Manage Traffic function. The output delivered by the process shall be available in audio or visual form in such way that while alerting the driver to the information it contains, it shall in no way impair the driver's ability to operate the vehicle in a manner that is both safe to its passengers and to other vehicles on the roads and freeways. The process shall maintain the output until new data is received from the other processes.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS2

Data Flows:

Input Flows

transit_vehicle_corrective_instructions
transit_vehicle_deviation_data

Output Flows

ttd_corrective_instructions
ttd_transit_vehicle_schedule_deviations

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.1;
USR = 2.1.1.2;
USR = 2.1.1.2.1;
USR = 2.1.1.2.1.3;
USR = 2.1.1.2.1.4;
USR = 2.1.1.2.1.4(a);
USR = 2.1.1.2.1.4(b);

4.1.2.4 Provide Transit Vehicle Correction Data Output Interface

Description:

Overview: This process shall provide the interface through which multimodal transportation service providers are informed of a transit vehicle schedule deviation. The output delivered by the process results from input received from another process in the Manage Transit function, and shall relate to the deviation of an individual transit vehicle. The process shall provide the output in a form that enables adjustments to be made to any connecting services being provided by the multimodal supplier so that transit users are not inconvenienced by the deviation of a transit vehicle on one service. A zero (or null) output shall be provided when no deviations are present.

In Subsystem:

Public Transport Management

In Market Package:

APTS7

Data Flows:

Input Flows

traffic_data_for_transit
transit_vehicle_arrival_conditions

Output Flows

tmtsp_transit_arrival_changes

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.2;
USR = 2.1.2.2.4;

4.1.2.5 Request Transit Vehicle Preemptions

Description:

Overview: This process shall provide the interface through which requests for preemption can be output from a transit

Public Transport Vehicle Subsystem

In Market Package:

APTS7

Data Flows:

Input Flows

transit_vehicle_preemption_request

Output Flows

transit_vehicle_roadway_preemptions

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.1;
USR = 2.1.1.2;
USR = 2.1.1.2.1;
USR = 2.1.1.2.1.4;
USR = 2.1.1.2.2;
USR = 2.1.1.2.3;

4.1.3 Provide Transit Vehicle Location Data

Description:

Overview: This process shall provide the transit vehicle's current location with a high degree of accuracy. The location shall be computed by this process from data sent by other processes that provides basic vehicle location and on-board vehicle conditions, such as proximity to transit stop, vehicle doors opened or closed, etc. The data shall be output continuously by the process and sent to other processes for their use and for storage.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS1

Data Flows:

Input Flows

transit_vehicle_on_board_data
vehicle_location_for_transit

Output Flows

transit_vehicle_location
transit_vehicle_location_for_deviation
transit_vehicle_location_for_eta
transit_vehicle_location_for_store

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.1;
USR = 2.1.1.1;
USR = 2.1.1.1(d);
USR = 2.1.1.1(f);
USR = 2.1.2;
USR = 2.1.2.2;
USR = 2.1.2.2.1;
USR = 2.1.2.2.1(a);
USR = 2.1.2.2.1(b);

4.1.4 Manage Transit Vehicle Deviations

Description:

Overview: This process shall manage large deviations of individual transit vehicles, deviations in rural areas, and deviations of large numbers of vehicles. The process shall generate the necessary corrective actions which may involve more than the vehicles concerned and more far reaching action, such as, the introduction of extra vehicles, wide area signal preemption by the Manage Traffic function, the premature termination of some services, etc. All corrective actions generated by this process shall be subject to the approval of the transit fleet manager before being implemented. Confirmation that the requested overall priority has been given by the Manage Traffic function shall be received by the process.

In Subsystem:

Public Transport Management

In Market Package:

APTS7

Data Flows:

Input Flows

ftfm_approved_corrections
prediction_data
transit_highway_priority_given
transit_ramp_priority_given
transit_road_priority_given
transit_services_for_scenarios
transit_vehicle_deviations_from_schedule
transit_vehicle_location_for_deviation

Output Flows

approved_corrective_plan
transit_highway_overall_priority
transit_ramp_overall_priority
transit_road_overall_priority
transit_vehicle_arrival_deviations
transit_vehicle_deviation_update
ttfm_proposed_corrections

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.1;
USR = 2.1.1.2;
USR = 2.1.1.2.1;
USR = 2.1.1.2.1.4;
USR = 2.1.1.2.2;
USR = 2.1.1.2.3;
USR = 2.1.1.2.4;

4.1.5 Provide Transit Vehicle Status Information

Description:

Overview: This process shall provide transit vehicle operational data to processes within the Manage Transit function, and on request to the transit fleet manager and the Manage Travel Demand facility in the Manage Traffic function. This process shall also provide transit probe and AVL information to the Manage Traffic function. Transit probe information can be provided by fixed route, flexibly routed, and paratransit services. The data shall be obtained by this process from another process that manages a store of transit vehicle operating data.

In Subsystem:

Public Transport Management

In Market Package:

APTS7

APTS8

Data Flows:

Input Flows

ftfm_request_transit_vehicle_data
transit_conditions_demand_request
transit_vehicle_information

Output Flows

transit_information_request
transit_probe_data
transit_running_data_for_demand
transit_vehicle_arrival_time
transit_vehicle_data
transit_vehicle_data_for_archive
transit_vehicle_status
ttfm_transit_vehicle_data

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.1;
USR = 2.1.1.1;
USR = 2.1.1.2;
USR = 2.1.1.2.1;
USR = 2.1.1.2.1.2;
USR = 2.1.1.2.1.3;

4.1.6 Manage Transit Vehicle Operations Data

Description:

Overview: This process shall manage the store of transit vehicle operating data. When any new data is received from another process, this process shall load it into the data store. This process shall also retrieve selected data on request from other processes in the Manage Transit function.

In Subsystem:

Public Transport Management

In Market Package:

APTS1

APTS2

APTS3

APTS8

Data Flows:

Input Flows

fm_transit_schedule_deviations_request
fws_current_weather
fws_predicted_weather
maintenance_status_for_transit
traffic_data_for_transit
transit_information_request
transit_vehicle_collected_maintenance_data
transit_vehicle_collected_trip_data
transit_vehicle_deviation_update
transit_vehicle_deviations_details_request
transit_vehicle_eta
transit_vehicle_location_for_store
transit_vehicle_schedule_deviation

Input/Output

transit_vehicle_operating_data

Output Flows

tm_transit_schedule_deviations_to_media
transit_deviation_data_received
transit_vehicle_advisory_eta
transit_vehicle_collected_maintenance_data_request
transit_vehicle_deviations_details
transit_vehicle_information
transit_vehicle_user_data

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1;
USR = 2.1.0;
USR = 2.1.1;
USR = 2.1.1.1;
USR = 2.1.1.1(a);
USR = 2.1.1.1(b);
USR = 2.1.1.1(e);
USR = 2.1.1.2;
USR = 2.1.1.2.1;
USR = 2.1.1.2.1.1;
USR = 2.1.1.2.4;
USR = 2.1.2;
USR = 2.1.2.2;
USR = 2.1.2.2.1;
USR = 2.1.2.2.1(c);
USR = 2.1.3;
USR = 2.1.3.2;

4.1.7.1 Manage Connections with External Systems

Description:

Overview: This process shall manage connections with Other Transit systems and with Multimodal Transportation Service Providers. The process shall provide to the other systems information on its transit services that interface with the other systems, in order to allow adjustments in the other systems to provide connection protection for passengers. The process shall receive inputs from the other systems with which it can cause adjustments to its own schedules to be made in order to provide connection protection. These adjustments may occur based upon the real time transporation asset schedules, or in more advance cases, based upon the needs of individual passengers.

Functional Requirements: none.

In Subsystem:

Public Transport Management

In Market Package:

APTS9

Data Flows:

Input Flows

connection_change_request
fmtsp_service_response
fotrm_service_response
other_TRM_real_time_data

Output Flows

current_connection_status
tmtsp_service_request
totrm_service_request
vehicle_correction_action

User Service Requirements Mapped to Pspec:

USR = C3.1.6.6;
USR = C3.1.6.7;

4.1.7.2 Provide Transit Vehicle Deviation Data Output Interface

Description:

Overview: This process shall provide the interface through which multimodal transportation service providers are informed of transit vehicle schedule deviations. The output delivered by the process shall result from input received from another process in the Manage Transit function, and shall relate to the deviation of a number of transit vehicles such that the disruption will affect several services, possibly on different routes. The process shall provide the output in a form that enables adjustments to be made to any connecting services being provided by the multimodal supplier so that transit users are not inconvenienced by the deviations. A zero (or null) output shall be provided when no deviations are present.

In Subsystem:

Public Transport Management

In Market Package:

APTS7

Data Flows:

Input Flows

transit_vehicle_arrival_deviations

Output Flows

tmtsp_transit_arrival_deviations

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.1;
USR = 2.1.2;
USR = 2.1.2.2;
USR = 2.1.2.2.4;
USR = 2.2.0;
USR = 2.2.1;
USR = 2.2.1.1;
USR = 2.2.1.1.2;
USR = 2.2.1.1.3;
USR = 2.2.1.1.4;

4.1.7.3 Manage Individual Service Requests

Description:

Overview: This process shall manage service requests for routing of an individual through the transit system. The process shall take requests from other processes (associated with the Information Service Provider or the Transit Passenger Information functions) and provide transit plans for both fixed and demand responsive transit. The process shall be able to coordinate with Other Transit Management systems or Multimodal Transportation Service Providers in order to provide a complete multimodal trip plan. In a more advanced implementation the process shall accept confirmations for the plans provided, track the passenger through the transit network, and coordinate with Other TRM and Multimodal Transportation Service Providers so that the passenger makes efficient connections between the transit system and other transit systems or other modes of transportation.

Functional Requirements: none.

In Subsystem:

Public Transport Management

In Market Package:

APTS9

Data Flows:

Input Flows

current_connection_status
fmtsp_individual_service_response
fotrm_individual_service_response
transit_trip_request
transit_user_information

Output Flows

connection_change_request
individual_transit_trip_plan
tmtsp_individual_service_request
totrm_individual_service_request

User Service Requirements Mapped to Pspec:

USR = C3.1.6.1;
USR = C3.1.6.2;
USR = C3.1.6.3;
USR = C3.1.6.4;
USR = C3.1.6.5;
USR = C3.1.6.6;
USR = C3.1.6.7;

4.1.8 Provide Transit Operations Data Distribution Interface

Description:

Overview: This process shall provide customized sets of transit vehicle schedule deviations to travelers, the traveler information data archive, and to the media. The process shall only provide data to the media and data archive when prompted by the arrival of new deviation data in the transit_vehicle_operational_data store, which is maintained by another process in the Manage Transit function. The outputs shall be made available following a direct request from the other ITS function, or as part of a subscription process relating to a traveler's transit profile. The process shall obtain the required data from the process that manages the store of transit vehicle operating data. The process shall send kiosk and personal transit deviation requests to the archival process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS1

ATIS2

ATIS4

Data Flows:

Input Flows

fm_transit_vehicle_deviations_request
transit_conditions_advisories_request
transit_conditions_guidance_request
transit_deviation_data_received
transit_deviation_kiosk_request
transit_deviations_personal_request
transit_vehicle_deviations_details
traveler_transit_profile

Output Flows

tm_transit_vehicle_deviations
transit_deviation_kiosk_request_for_archive
transit_deviations_for_broadcast_to_kiosks
transit_deviations_for_broadcast_to_personal_devices
transit_deviations_for_kiosks
transit_deviations_for_personal_devices
transit_deviations_personal_request_for_archive
transit_running_data_for_advisory_output
transit_running_data_for_guidance
transit_vehicle_deviations_details_request

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.1;
USR = 2.1.2;
USR = 2.1.2.2;
USR = 2.1.2.2.1;
USR = 2.1.2.2.5;
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;

4.1.9 Process Transit Vehicle Sensor Maintenance Data

Description:

Overview: This process shall collect and process vehicle maintenance data available to sensors on-board transit vehicles. When processed, the data shall be sent by this process on request to another process in the Manage Transit function for storage as transit vehicle operating data so that it can subsequently be used for work on future vehicle maintenance.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS6

Data Flows:

Input Flows

ftv_vehicle_maintenance_data
transit_vehicle_collected_maintenance_data_request

Output Flows

transit_vehicle_collected_maintenance_data

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.1;
USR = 2.1.1.1;
USR = 2.1.1.1(c);
USR = 2.1.3;
USR = 2.1.3.1;
USR = 2.1.3.1.1;

4.2.1.1 Process Demand Responsive Transit Trip Request

Description:

Overview: This process shall provide the interface through which processes in the Provide Driver and Traveler Service function can gain access to the Provide Demand Responsive Transit Service facility. The process shall enable the interface to support the receipt of trip requests, their transfer to another process for the actual demand responsive schedule generation, the output of the proposed schedule and their (possible) subsequent confirmation. The process shall store the input and schedule data relating to each request until such time as the request is confirmed or the data in the request is no longer valid, e.g. the time(s) used in the proposed schedule has(ve) passed. The confirmation of a particular schedule shall be sent by the process to another process that will enable the schedule to be implemented.

In Subsystem:

Public Transport Management

In Market Package:

APTS3

Data Flows:

Input Flows

paratransit_schedule
paratransit_service_confirmation
paratransit_trip_request

Input/Output

paratransit_service_data

Output Flows

paratransit_personal_schedule
paratransit_request
paratransit_requested_services
paratransit_service_data_for_archive

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.3.0;
USR = 2.3.1;
USR = 2.3.1.1;
USR = 2.3.1.2;
USR = 2.3.2;
USR = 2.3.2.7;

4.2.1.2 Compute Demand Responsive Transit Vehicle Availability

Description:

Overview: This process shall provide the facility for the calculation of the location and availability of transit vehicles for use in demand responsive transit operations. The process shall base its calculation on the vehicle's current location and on the output from a process that determines vehicle availability from data input to sensors. The output shall be loaded by the process into a store for use by another process.

In Subsystem:

Public Transport Management

In Market Package:

APTS3

Data Flows:

Input Flows

paratransit_transit_vehicle_availability
transit_vehicle_location

Output Flows

paratransit_available_vehicles

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.3.0;
USR = 2.3.2;
USR = 2.3.2.6;
USR = 2.3.2.7;

4.2.1.3 Generate Demand Responsive Transit Schedule and Routes

Description:

Overview: This process shall provide dynamic routing and scheduling of transit vehicles so that a demand responsive transit service can be provided. The generation of the specific route and schedule by the process shall be initiated by a request from the management process. The choice of route and schedule produced by the process shall depend on what other demand responsive transit schedules have been planned, the availability and location of vehicles, and the relevance of any regular transit routes and schedules. The process shall send its output to another process for output to the requesting process, and shall also load it into a data store for use if the schedule is later confirmed.

In Subsystem:

Public Transport Management

In Market Package:

APTS3

Data Flows:

Input Flows

paratransit_available_vehicles
paratransit_request
traffic_data_for_transit
transit_services_for_demand_response

Input/Output

paratransit_services

Output Flows

paratransit_schedule
transit_services_demand_response_request

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.3.0;
USR = 2.3.2;
USR = 2.3.2.1;
USR = 2.3.2.10;
USR = 2.3.2.2;
USR = 2.3.2.3;
USR = 2.3.2.4;
USR = 2.3.2.5;
USR = 2.3.2.6;
USR = 2.3.2.7;
USR = 2.3.2.8;
USR = 2.3.2.9;
USR = 2.3.4;
USR = 2.3.4.2;

4.2.1.4 Confirm Demand Responsive Transit Schedule and Route

Description:

Overview: This process shall provide output when a demand responsive transit schedule is confirmed. The outputs shall contain details of the schedule and shall be sent to the transit fleet manager and to processes that provide interfaces to the transit driver, a store of data used by the regular transit routes and schedule generation processes, and the transit driver schedule generation processes. The process shall obtain the data for the outputs from the store of data provided by the schedule generation process.

In Subsystem:

Public Transport Management

In Market Package:

APTS3

Data Flows:

Input Flows

paratransit_requested_services
paratransit_services

Output Flows

paratransit_service_output
paratransit_services_for_transit_drivers
paratransit_transit_driver_instructions
ttfm_paratransit_service

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.3.0;
USR = 2.3.1;
USR = 2.3.1.3;
USR = 2.3.2;
USR = 2.3.2.2;
USR = 2.3.2.3;
USR = 2.3.4;
USR = 2.3.4.3;

4.2.1.5 Process Demand Responsive Transit Vehicle Availability Data

Description:

Overview: This process shall manage data input to sensor(s) on board a transit vehicle. Data including the vehicle's availability for use in demand responsive transit services shall be provided by this process to other processes within the Manage Transit function.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS3

Data Flows:

Input Flows

ftv_availability

Output Flows

paratransit_transit_vehicle_availability

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.3.0;
USR = 2.3.3;
USR = 2.3.3.1;
USR = 2.3.3.1(a);
USR = 2.3.3.1(b);
USR = 2.3.3.1(c);
USR = 2.3.3.2;
USR = 2.3.3.2(a);
USR = 2.3.3.2(b);
USR = 2.3.3.3;

4.2.1.6 Provide Demand Responsive Transit Driver Interface

Description:

Overview: This process shall provide the interface through which a transit driver will be sent instructions about the demand responsive transit schedule that has been confirmed. The process shall send the data in a format that will enable the driver to implement the schedule. The output provided by the process shall be available in audio or visual form in such a way that while alerting the driver to the information it contains, it shall in no way impair the driver's ability to operate the vehicle in a manner that is both safe to its passengers, and to other vehicles on the roads and freeways. The input and output forms shall also include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS3

Data Flows:

Input Flows

paratransit_transit_driver_instructions

Output Flows

ttd_paratransit_information

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.3.0;
USR = 2.3.5;
USR = 2.3.5.1;
USR = 2.3.5.2;
USR = 2.3.5.2(a);
USR = 2.3.5.2(b);
USR = 2.3.5.3;
USR = 2.3.5.4;
USR = 2.3.5.4(a);
USR = 2.3.5.4(b);

4.2.2 Provide Transit Plans Store Interface

Description:

Overview: This process shall provide the interface to the store of current regular transit plans, i.e., routes and schedules and demand responsive transit schedules. The process shall enable the store to be used by the Demand Responsive Transit facility as a source of data about regular transit services when it is generating its schedules. The demand responsive transit schedule data shall be accessible as input to the regular transit route and schedule generation processes.

In Subsystem:

Public Transport Management

In Market Package:

APTS2

Data Flows:

Input Flows

paratransit_service_output
transit_routes_request
transit_routes_updates
transit_schedule_request
transit_schedule_updates
transit_services_demand_response_request

Input/Output

transit_plans

Output Flows

transit_routes_current_data
transit_schedule_current_data
transit_services_for_demand_response

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.1;
USR = 2.1.2.1.1;
USR = 2.2.0;
USR = 2.2.1;
USR = 2.2.1.1;
USR = 2.2.1.1.4;

4.2.3.1 Generate Transit Routes

Description:

Overview: This process shall generate new transit routes. The process shall use parameters set up by the transit fleet manager, operational data for the current routes and schedules, plus the current routes and digitized map data, as sources of input from which the new routes are generated. The process shall also use the requested input data containing the demand responsive transit routes and schedules. The generation of new routes by the process shall be initiated as a result of data received from the transit fleet manager interface process, with the output being sent to other processes for storage. The output data produced by the process shall include sufficient data for a specialist map data provider to generate maps showing transit routes and stops, either as separate data or as part of the general digitized map data provided to other ITS functions.

In Subsystem:

Public Transport Management

In Market Package:

APTS2

Data Flows:

Input Flows

map_data_for_transit
transit_operational_data
transit_routes_current_data
transit_service_planning_parameters
update_routes

Output Flows

transit_routes_data
transit_routes_request
transit_routes_updates

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.1;

4.2.3.2 Generate Schedules

Description:

Overview: This process shall generate new transit schedules for use by the regular transit operation. The process shall use parameters set up by the transit fleet manager, operational data for the current routes and schedules, plus the current routes and schedules themselves, as sources of input from which the new schedules are generated. The process shall also use the data containing the demand responsive transit routes and schedules to generate the new schedules. The generation of new schedules by the process shall be initiated as a result of data received from the transit fleet manager interface process or a request for services to a parking lot. The process shall send its output to another process for storage.

In Subsystem:

Public Transport Management

In Market Package:

APTS2

APTS7

Data Flows:

Input Flows

parking_lot_transit_request
transit_operational_data
transit_schedule_current_data
transit_service_planning_parameters
update_schedules

Output Flows

parking_lot_transit_response
transit_schedule_data
transit_schedule_request
transit_schedule_updates

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.1;
USR = 2.1.2.1.1;
USR = 2.1.2.2;
USR = 2.1.2.2.2;
USR = 2.1.2.2.3;
USR = 2.1.2.2.5;
USR = 2.1.2.2;
USR = 2.1.2.2.1;
USR = 2.1.2.2.2;

4.2.3.3 Produce Transit Service Data for External Use

Description:

Overview: This process shall obtain transit routes and services data and distribute it to ITS functions that are outside the transit center. The process shall run when a request for data is received from an external source, or when fresh data is received. In the latter case, the data shall only be sent by the process to the multimodal transportation service provider. For data requests that include an origin and a destination, the process shall only provide details of the transit service(s) that link the two points. The details shall only cover those portion(s) of the service(s) that are needed to complete the requested trip and not full details of the services.

In Subsystem:

Public Transport Management

In Market Package:

APTS8

Data Flows:

Input Flows

transit_service_external_data
transit_services_advisories_request
transit_services_demand_request
transit_services_guidance_request
transit_services_kiosk_request
transit_services_personal_request
transit_services_travelers_request

Output Flows

request_transit_service_external_data
tmtsp_transit_service_data
transit_services_for_advisory_data
transit_services_for_demand
transit_services_for_deployment
transit_services_for_guidance
transit_services_for_kiosks
transit_services_for_personal_devices
transit_services_for_travelers

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.3.0;
USR = 2.3.2;
USR = 2.3.2.2;
USR = 2.3.2.3;

4.2.3.4 Provide Transit Fleet Manager Interface for Services Generation

Description:

Overview: This process shall provide the interface through which the transit fleet manager controls the generation of new routes and schedules (transit services). The transit fleet manager shall be able to review and update the parameters used by the routes and schedules generation processes and to initiate these processes. This process shall also act as the interface through which the Manage Demand facility in the Manage Traffic function can request changes to the current routes and schedules in its efforts to adjust the modal split of travelers' trips in order to make the most efficient use of the road and highway network served by the local ITS functions. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Management

In Market Package:

APTS2

Data Flows:

Input Flows

ftfm_initiate_service_updates
ftfm_planning_parameters
ftfm_planning_parameters_update_request
ftfm_transit_display_update_request
ftfm_transit_services_output_request
map_data_for_transit
transit_services_changes_request
transit_services_data_for_output

Output Flows

request_transit_map_update
request_transit_services_data_for_output
transit_service_planning_parameters
transit_services_changes_response
ttfm_parameters
ttfm_transit_services_output
update_routes
update_schedules

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.1;
USR = 2.1.2.1.1;
USR = 2.1.2.1.2;
USR = 2.1.2.2;
USR = 2.1.2.2.2;
USR = 2.1.2.2.4;
USR = 2.1.2.2.4(a);
USR = 2.1.2.2.4(b);

4.2.3.5 Manage Transit Operational Data Store

Description:

Overview: This process shall collect transit operational data and load it into a data store for use by the routes and schedules generation processes. The data shall be provided to this process by other processes in the Manage Transit function and shall enable an accurate picture of how routes and schedules are currently operating in terms of the numbers of vehicles that are available, the numbers of passengers that they are carrying, and the numbers of passengers passing through each roadside facility (transit stop).

In Subsystem:

Public Transport Management

In Market Package:

APTS2

Data Flows:

Input Flows

ftfm_passenger_loading_updates
transit_roadside_passenger_data
transit_vehicle_availability
transit_vehicle_data
transit_vehicle_passenger_data

Input/Output

transit_operational_data

Output Flows

transit_operational_data_for_archive
ttfm_passenger_loading_error

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.1;
USR = 2.1.2.1.1;
USR = 2.2.0;
USR = 2.2.1;
USR = 2.2.1.1;
USR = 2.2.1.1.4;
USR = 2.3.0;
USR = 2.3.4;

4.2.3.6 Produce Transit Service Data for Manage Transit Use

Description:

Overview: This process shall obtain transit routes and services data and distribute it internally to other processes in the Manage Transit function. The process shall only provide its outputs when fresh data is received from another process. If this does not happen for a long period of time (days), then the process shall initiate its own request for fresh data.

In Subsystem:

Public Transport Management

In Market Package:

APTS2

Data Flows:

Input Flows

transit_service_internal_data
transit_services_for_eta_request

Output Flows

request_transit_service_internal_data
transit_services_for_advanced_fares
transit_services_for_corrections
transit_services_for_eta
transit_services_for_roadside_fares
transit_services_for_scenarios
transit_services_for_transit_drivers
transit_services_for_vehicle_fares

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.3.0;
USR = 2.3.2;
USR = 2.3.2.2;
USR = 2.3.2.3;

4.2.3.7 Provide Interface for Other TRM Data

Description:

Overview: This process shall provide the interface through which transit routes and schedules can be exchanged with other transit centers (Other TRM). This data shall be output when data is received from another (local) process and shall enable coordination between services provided by adjacent transit operations, particularly where they serve the same geographic areas. The process shall also provide routes and schedules to the local process when the data is received from other transit centers.

In Subsystem:

Public Transport Management

In Market Package:

APTS7

Data Flows:

Input Flows

fotrm_transit_services
transit_services_for_other_TRM

Output Flows

other_TRM_service_data
totrm_transit_services

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.8.0;
USR = 1.8.1;
USR = 1.8.1.2;
USR = 1.8.1.2(c);
USR = 1.8.1.3;
USR = 1.8.1.3(c);
USR = 1.8.1.4;
USR = 1.8.1.4(c);

4.2.3.8 Provide Interface for Transit Service Raw Data

Description:

Overview: This process shall provide and manage the interface to the store in which the raw transit service data is held. This data shall be sent to the process by the routes and schedules generation processes, which are the only other processes permitted to access the store, and then in read-only mode. The received data shall be loaded into the store and distributed by this process to the three processes that are responsible for distributing the data within the transit center (TRM), to other local ITS functions, and to other transit centers (Other TRM), respectively. The process shall read data from the store and return it to whichever of the other three processes has made a data request. Data shall also be received by the process from other transit centers (Other TRM) and from multimodal transportation service providers. The process shall load this data into the data store for use by the local route and schedule generation processes.

In Subsystem:

Public Transport Management

In Market Package:

APTS7

Data Flows:

Input Flows

fmtsp_transit_service_data
map_transit_data
other_TRM_service_data
request_transit_service_external_data
request_transit_service_internal_data
request_transit_services_data_for_output
transit_routes_data
transit_schedule_data

Input/Output

transit_service_raw_data

Output Flows

transit_service_external_data
transit_service_internal_data
transit_services_data_for_output
transit_services_for_other_TRM

User Service Requirements Mapped to Pspec:

USR = NA;

4.2.3.9 Update Transit Map Data

Description:

Overview: This process shall provide updates to the store of digitized map data used by the transit route generation process and as the background for displays of transit services requested by the transit fleet manager. The process shall obtain the new data from a specialist data supplier or some other appropriate data source, after receiving an update request from the transit fleet manager interface process within the function. The processes requiring data for use in transit route generation and as the background to displays will read the data from the store loaded by this process.

In Subsystem:

Public Transport Management

In Market Package:

APTS1

Data Flows:

Input Flows

fmup_transit_map_update
request_transit_map_update

Output Flows

map_data_for_transit
tmup_transit_map_update_request

User Service Requirements Mapped to Pspec:

USR = 1.8;
USR = 1.8.2;
USR = 1.8.2.4;
USR = 1.8.2.4(c);
USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.1;
USR = 2.1.2.1.2;
USR = 2.1.2.2;
USR = 2.1.2.2.2;

4.2.4 Manage Transit Archive Data

Description:

Overview: This process shall obtain transit passenger and deployment data, transit user payment transaction data, transit emergency data, transit security data, maintenance and personnel data, and distribute it to the Manage Archive Data function. The process shall run when a request for data is received from an external source, or when fresh data is received.

In Subsystem:

Public Transport Management

In Market Package:

AD1

AD2

Data Flows:

Input Flows

bad_transit_collected_fare_payment
bad_transit_roadside_fare_payment
bad_transit_vehicle_fare_payment
ftso_archive_commands
paratransit_service_data_for_archive
transit_archive_request
transit_archive_status
transit_driver_info_for_archive
transit_emergency_data_for_archive
transit_fare_transactions
transit_incident_info_for_archive
transit_operational_data_for_archive
transit_route_assign_for_archive
transit_services_for_deployment
transit_technician_info
transit_user_payments_transactions
transit_vehicle_data_for_archive
transit_vehicle_maintenance_info

Input/Output

transit_data_archive

Output Flows

transit_archive_data
ttso_archive_status

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.4;
USR = 7.1.3.1.4(a);
USR = 7.1.3.1.4(b);
USR = 7.1.3.1.4(d);
USR = 7.1.3.1.4(e);
USR = 7.1.3.1.4(f);
USR = 7.1.3.1.4(g);
USR = 7.1.3.1.9; USR = 7.1.3.1.9(b); USR = 7.1.3.1.9(c); USR = 7.1.3.1.9(d);

4.3.1 Monitor Transit Vehicle Condition

Description:

Overview: This process shall monitor the condition of a transit vehicle. It shall use the transit vehicle maintenance specification to analyze brake, drive train, sensors, fuel, steering, tire, processor, communications equipment, and transit vehicle mileage to identify mileage based maintenance, out-of-specification or imminent failure conditions. The data resulting from this analysis shall be loaded by the process into the store of transit vehicle operations data, through the output flow transit vehicle maintenance. This data is then sent to the process that generates transit vehicle maintenance schedules.

In Subsystem:

Public Transport Management

In Market Package:

APTS6

Data Flows:

Input Flows

transit_vehicle_maintenance_specs
transit_vehicle_status

Output Flows

transit_vehicle_maintenance
transit_vehicle_maintenance_information

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.1;
USR = 2.1.2.1.2;
USR = 2.1.3;
USR = 2.1.3.1;
USR = 2.1.3.1.2;

4.3.2 Generate Transit Vehicle Maintenance Schedules

Description:

Overview: This process shall generate transit vehicle maintenance schedules and includes what and when maintenance or repair is to be performed. Transit vehicle availability listings (current and forecast) shall also be generated by the process to support transit vehicle assignment planning. The maintenance and/or repair that is to be performed on the transit vehicle shall be scheduled by the process for a specific month, week, day(s), and hour(s). The availability of the transit vehicle that is also output by the process shall be based upon the transit vehicle maintenance schedule. The process shall load each transit vehicle maintenance schedule that it produces into the store of transit vehicle operations data, through the process that maintains this data store.

In Subsystem:

Public Transport Management

In Market Package:

APTS6

Data Flows:

Input Flows

transit_vehicle_maintenance_information

Output Flows

transit_vehicle_availability
transit_vehicle_maintenance_schedule
transit_vehicle_maintenance_schedule_data

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.1;
USR = 2.1.2.1.2;
USR = 2.1.3;
USR = 2.1.3.1;
USR = 2.1.3.1.2;
USR = 2.1.3.1.3;
USR = 2.1.3.1.4;

4.3.3 Generate Technician Work Assignments

Description:

Overview: This process shall assign transit maintenance personnel to a transit vehicle maintenance schedule. The maintenance schedule shall be received from another process and shall define what and when maintenance repair is to be performed to a specific transit vehicle. The process shall base the personnel assignment upon details about the personnel obtained from the transit fleet manager and held in a local data store. These details shall comprise personnel eligibility, work assignments, preferences and seniority. The process shall also provide these details to the transit fleet manager on request. When a work assignment has been generated, the process shall send it to the transit maintenance personnel and also to the process that monitors and verifies maintenance work activity. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Management

In Market Package:

APTS6

Data Flows:

Input Flows

ftfm_technician_information_request
ftfm_technician_information_updates
transit_vehicle_maintenance_schedule_data
transit_vehicle_maintenance_verification_results

Input/Output

transit_technician_data

Output Flows

transit_technician_info
transit_technician_work_assignment
ttfm_technician_information
ttmp_work_schedule

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.1;
USR = 2.1.2.1.2;
USR = 2.1.3;
USR = 2.1.3.1;
USR = 2.1.3.1.2;
USR = 2.1.3.1.3;
USR = 2.1.3.1.4;

4.3.4 Monitor And Verify Maintenance Activity

Description:

Overview: This process shall verify that the transit vehicle maintenance activities were performed correctly and that a time stamped maintenance log for record keeping was generated. The correctness of the maintenance activities shall be judged by the process against the transit vehicle's status, the maintenance personnel's work assignment, and the transit maintenance schedules produced by other processes. The process shall save a time stamped record of all the maintenance activities performed on the vehicle into the transit vehicle maintenance log.

In Subsystem:

Public Transport Management

In Market Package:

APTS6

Data Flows:

Input Flows

transit_technician_work_assignment
transit_vehicle_maintenance_specs
transit_vehicle_status

Output Flows

transit_vehicle_maintenance_log_data
transit_vehicle_maintenance_verification_results

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.1;
USR = 2.1.2.1.2;
USR = 2.1.3;
USR = 2.1.3.1;
USR = 2.1.3.1.2;
USR = 2.1.3.1.5;

4.3.5 Report Transit Vehicle Information

Description:

Overview: This process shall provide the transit fleet managers with the capability of requesting and receiving transit vehicle maintenance information. The process shall obtain the data for each request from the store of transit vehicle operations data, through the process that manages the data store, and shall produce the output to the transit fleet manager in an easily understood form. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Management

In Market Package:

APTS6

Data Flows:

Input Flows

ftfm_transit_vehicle_maintenance_information_request
ftfm_transit_vehicle_maintenance_specs
transit_vehicle_maintenance_data

Output Flows

transit_vehicle_maintenance_data_request
transit_vehicle_maintenance_specs_update
ttfm_transit_vehicle_maintenance_information

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.1;
USR = 2.1.2.1.2;
USR = 2.1.3;
USR = 2.1.3.1;
USR = 2.1.3.1.2;
USR = 2.1.3.1.5;

4.3.6 Update Transit Vehicle Information

Description:

Overview: This process shall provide the transit maintenance personnel with the capability to update transit vehicle maintenance information. The process shall send the data received from the transit maintenance personnel to the transit vehicle operations data store management process for use by other processes.

In Subsystem:

Public Transport Management

In Market Package:

APTS6

Data Flows:

Input Flows

ftmp_transit_vehicle_maintenance_updates

Output Flows

transit_vehicle_maintenance_data_update

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.1;
USR = 2.1.2.1.2;
USR = 2.1.3;
USR = 2.1.3.1;
USR = 2.1.3.1.2;
USR = 2.1.3.1.5;

4.3.7 Manage Transit Vehicle Operations Data Store

Description:

Overview: This process shall manage the store of transit vehicle operations data. It shall be able to load data it receives about vehicle maintenance into the store and provide that data on request to other processes.

In Subsystem:

Public Transport Management

In Market Package:

APTS6

Data Flows:

Input Flows

transit_vehicle_maintenance
transit_vehicle_maintenance_data_request
transit_vehicle_maintenance_data_update
transit_vehicle_maintenance_log_data
transit_vehicle_maintenance_schedule
transit_vehicle_maintenance_specs_update

Input/Output

transit_vehicle_operations_data

Output Flows

transit_vehicle_maintenance_data
transit_vehicle_maintenance_info
transit_vehicle_maintenance_specs

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.2;
USR = 2.1.2.1;
USR = 2.1.2.1.2;
USR = 2.1.3;
USR = 2.1.3.1;
USR = 2.1.3.1.2;
USR = 2.1.3.1.3;
USR = 2.1.3.1.4;
USR = 2.1.3.1.5;

4.4.1.1 Manage Transit Security

Description:

Overview: This process shall manage the security in the transit system by monitoring for potential incidents. Data shall be obtained by the process from a variety of sources and assessed for any security problems. Problems shall be passed by the process to the transit system operator for review and the required action. Information about incidents shall also be sent by this process to another process for output to the media, using interface parameters set up by the transit system operator.

In Subsystem:

Public Transport Management

In Market Package:

APTS5

Data Flows:

Input Flows

emergency_request_transit_details
secure_area_surveillance_information
transit_media_incident_interface_parameters
transit_operator_security_action

Output Flows

emergency_acknowledge_transit_details
secure_area_broadcast_message
secure_area_monitoring_control
transit_incident_details
transit_incident_info_for_archive
transit_incident_information
transit_media_incident_information
transit_operator_incident_information

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.4.0;
USR = 2.4.1;
USR = 2.4.1.1;
USR = 2.4.1.2;
USR = 2.4.2;
USR = 2.4.2.1;
USR = 2.4.2.2;
USR = 2.4.4;
USR = 2.4.4.1;
USR = 2.4.4.2;

4.4.1.2 Manage Transit Emergencies

Description:

Overview: This process shall support the management of emergencies that occur in the transit system by processing information received from transit vehicles. The process shall accept inputs from either the transit vehicle driver or a transit user, the latter through such interfaces as panic buttons, alarm switches, etc. The reported emergencies shall be sent to another process for action by the transit system operator and subsequently for output to the media. The process shall also send acknowledgment data to the process providing the interface to the transit driver.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS5

Data Flows:

Input Flows

ftu_emergency_request
transit_driver_emergency_request
transit_operator_request_acknowledge
transit_vehicle_location

Output Flows

transit_driver_emergency_acknowledge
transit_emergency_details
transit_emergency_information
transit_operator_emergency_request

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.4;
USR = 2.1.4.1;
USR = 2.1.4.4;
USR = 2.4.0;
USR = 2.4.1;
USR = 2.4.1.2;
USR = 2.4.1.3;

4.4.1.3 Provide Transit System Operator Security Interface

Description:

Overview: This process shall provide an interface for the transit system operator to identify and act upon potential information security problems and emergencies. This information shall be provided by other processes through input data flows. This process shall also provide the capability for the transit system operator to update parameters that control the output of data about the potential security problems to the media. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Management

In Market Package:

APTS5

Data Flows:

Input Flows

ftso_emergency_request_acknowledge
ftso_media_parameter_request
ftso_media_parameter_updates
ftso_security_action
ftso_video_camera_action_request
transit_operator_emergency_request
transit_operator_incident_information

Input/Output

transit_media_interface_parameters

Output Flows

transit_operator_request_acknowledge
transit_operator_security_action
ttso_emergency_request
ttso_media_parameters
ttso_potential_incidents_alarm
ttso_potential_security_problem
ttso_video_image_data

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.1;
USR = 2.1.1.2;
USR = 2.1.1.2.1;
USR = 2.1.2;
USR = 2.1.2.1;
USR = 2.1.2.1.2;
USR = 2.1.4;
USR = 2.1.4.1;
USR = 2.1.4.4;
USR = 2.4.0;
USR = 2.4.4;
USR = 2.4.4.1;
USR = 2.4.4.2;
USR = 2.4.4.3;

4.4.1.4 Provide Transit External Interface for Emergencies

Description:

Overview: This process shall provide the interface through which information about security problems and emergencies detected within the transit system are distributed directly to the media and other information systems. This process shall construct its output from the data supplied by other processes. This data shall contain parameters that define the way (format, content, etc.) in which the information is output by the process. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Management

In Market Package:

APTS5

Data Flows:

Input Flows

fm_transit_incident_information_request
transit_media_emergency_information
transit_media_incident_information

Output Flows

tm_transit_emergency_information
tm_transit_incident_information
transit_incident_data

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.4;
USR = 2.1.4.3; USR = 2.1.4.4; USR = 2.1.4.4(d);

4.4.1.5 Provide Transit Driver Interface for Emergencies

Description:

Overview: This process shall provide an interface to the transit vehicle through which the driver can both report an emergency situation and receive an acknowledgment. The process shall provide this interface in such a way that its operation for both inputs and outputs shall be transparent to transit users on board the vehicle and to anyone outside the vehicle, and shall not compromise the safe operation of the vehicle by the driver.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS5

Data Flows:

Input Flows

ftd_emergency_request
transit_driver_emergency_acknowledge

Output Flows

transit_driver_emergency_request
ttd_emergency_information

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.4;
USR = 2.1.4.2;
USR = 2.1.4.4;
USR = 2.1.4.4(a);
USR = 2.1.4.4(b);
USR = 2.1.4.4(c);
USR = 2.1.4.4(d);

4.4.1.6 Collect Transit Vehicle Emergency Information

Description:

Overview: This process shall collect data about emergencies that occur on-board transit vehicles for output to the media and the Manage Emergency Services function. These emergencies may be reported by either the transit driver or a transit user, the latter through such interfaces as panic buttons, alarm switches, etc. For output to the media interface process, the data shall be combined with the data in the media interface parameters data store.

In Subsystem:

Public Transport Management

In Market Package:

APTS5

Data Flows:

Input Flows

transit_emergency_details
transit_media_emergency_interface_parameters

Output Flows

transit_emergency_data
transit_emergency_data_for_archive
transit_media_emergency_information

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.4;
USR = 2.1.4.3;
USR = 2.1.4.4;
USR = 2.4.0;
USR = 2.4.1;
USR = 2.4.1.1;
USR = 2.4.1.1(a);
USR = 2.4.1.1(b);
USR = 2.4.1.1(c);
USR = 2.4.1.1(d);
USR = 2.4.1.1(e);
USR = 2.4.1.3;

4.4.1.7 Monitor Secure Area

Description:

Overview: This process shall monitor the secure area environment. Data shall be obtained by the process from a variety of sources and assessed for any security problems. Problems shall be passed by the process to other processes for review and the required action. Information about incidents shall also be sent by this process to another process for output to the media, using interface parameters set up by the transit system operator. The process shall also provide facilities for the control of video cameras and audio output in the secure area environment.

In Subsystem:

Remote Traveler Support

In Market Package:

APTS5

Data Flows:

Input Flows

fsa_area_image
secure_area_broadcast_message
secure_area_monitoring_control

Output Flows

secure_area_surveillance_information

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.4;
USR = 2.1.4.3;
USR = 2.1.4.4;
USR = 2.2.1.2;
USR = 2.2.1.2.1;
USR = 2.2.1.2.1.2;
USR = 2.2.1.2.1.2(a);
USR = 2.2.1.2.1.2(b);
USR = 2.2.1.2.1.2(c);
USR = 2.4.0;
USR = 2.4.1;
USR = 2.4.1.1;
USR = 2.4.1.1(a);
USR = 2.4.1.1(b);
USR = 2.4.1.1(c);
USR = 2.4.1.1(d);
USR = 2.4.1.1(e);
USR = 2.4.1.2;
USR = 2.4.2;
USR = 2.4.2.1;
USR = 2.4.2.2;
USR = 2.4.4;
USR = 2.4.4.1;
USR = 2.4.4.2;

4.4.1.8 Report Traveler Emergencies

Description:

Overview: This process shall provide an interface in the Provide Driver and Traveler Services function through which travelers can declare emergencies. The traveler may be at a kiosk or other device, transit stop, transit depot, etc. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Remote Traveler Support

In Market Package:

APTS5

EM3

Data Flows:

Input Flows

emergency_acknowledge_transit_details
emergency_request_traveler_acknowledge
ft_remote_emergency_request

Output Flows

emergency_request_transit_details
emergency_request_traveler_details
tt_emergency_response

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.1.0;
USR = 1.7.0;
USR = 1.7.1;
USR = 1.7.1.2;
USR = 1.7.1.2.1;
USR = 2.0;
USR = 2.1.0;
USR = 2.1.4;
USR = 2.1.4.3;
USR = 2.1.4.4;
USR = 2.4.1;
USR = 2.4.1.2;

4.4.2 Coordinate Multiple Agency Responses to Incidents

Description:

Overview: This process shall provide transit fleet managers with an interface through which they can control the coordination data sent to the Manage Emergency Services function following the detection of a security problem or emergency within the transit operations network by other processes. The process shall use data from the store of predefined responses to security problems and emergencies in the outputs that it sends to the Manage Emergency Services function. If no match can be found then the process shall send all the available data to the transit fleet manager for action. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Management

In Market Package:

APTS5

Data Flows:

Input Flows

ftfm_coordination_data
transit_emergency_information
transit_incident_coordination_data
transit_incident_information
transit_preplanned_responses_for_incidents

Output Flows

transit_coordination_data
ttfm_coordination_request

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.4.0;
USR = 2.4.4;
USR = 2.4.4.3;
USR = 2.4.4.5;

4.4.3 Generate Responses for Incidents

Description:

Overview: This process shall provide the interface through which the transit fleet manager can enter and review predefined responses to security problems and emergencies that have been detected by other processes within the Manage Transit function. This data shall be stored in a form which can be used by another process to provide coordination data to the Manage Emergency Services function. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Management

In Market Package:

APTS5

Data Flows:

Input Flows

ftfm_request_response_parameter_output
ftfm_response_parameters

Input/Output

transit_preplanned_responses_for_incidents

Output Flows

ttfm_response_parameter_output

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.4.0;
USR = 2.4.4;
USR = 2.4.4.4;

4.4.4 Coordinate Transit Disaster Response

Description:

Overview: This process is responsible for managing the interface between the Manage Transit function and the Disaster Command Authority terminator. This process shall respond to requests for availability data, situation reports on current disasters, shelter availability and locations, as well as inputs to request the local transit services function deploy transit vehicles to support evacuation activities. This process shall forward the requests onto the appropriate functions. Upon receipt of the required data about evacuation status this process shall forward the data to the Disaster Command Authority.

Functional Requirements: none.

In Subsystem:

Public Transport Management

In Market Package:

EM4

EM5

Data Flows:

Input Flows

fdca_disaster_transit_advisory
fdca_disaster_transit_shelter_availability
fdca_disaster_transit_shelter_location
fdca_transit_evacuation_request
transit_disaster_response

Output Flows

tdca_transit_evacuation_status
tdca_transit_shelter_request
transit_disaster_request

User Service Requirements Mapped to Pspec:

USR = C6.3.1.2;
USR = C6.3.1.3;
USR = C6.3.2.2;

4.5.1 Assess Transit Driver Performance

Description:

Overview: This process shall assess the transit driver's performance at previous work assignments. The process shall carry out this activity by 1) utilizing standardized performance evaluation criteria set forth by governmental regulations and transit operating company policies, 2) assessing the transit driver's driving history, and 3) assessing comments from the transit driver's supervisor(s). It shall also use the details of any moving violations or accidents, supervisor comments, government regulations, and company policies. The data shall be sent to this process by the process that provides the interface to a local data store, each time that the store is updated with driver performance data.

In Subsystem:

Public Transport Management

In Market Package:

APTS2

APTS3

Data Flows:

Input Flows

transit_driver_performance_considerations

Output Flows

transit_driver_performance
transit_driver_performance_data

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.3;
USR = 2.1.3.2;
USR = 2.1.3.2.4;

4.5.2 Assess Transit Driver Availability

Description:

Overview: This process shall assess the transit driver's availability based on previous work assignments plus health and vacation commitments. The process shall carry out this activity by 1) utilizing standardized transit driver work criteria set forth by governmental regulations and company policies, 2) monitoring the transit driver's health status and vacation status, and 3) monitoring the transit driver's accumulated work hours. The data shall be sent to this process by the process that provides the interface to a local data store, each time that the store is updated with driver availability data.

In Subsystem:

Public Transport Management

In Market Package:

APTS2

APTS3

Data Flows:

Input Flows

transit_driver_availability_considerations

Output Flows

transit_driver_availability
transit_driver_availability_data

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.3;
USR = 2.1.3.2;
USR = 2.1.3.2.1;
USR = 2.1.3.2.2;
USR = 2.1.3.2.3;
USR = 2.1.3.2.4;
USR = 2.3.4;
USR = 2.3.4.3;

4.5.3 Access Transit Driver Cost Effectiveness

Description:

Overview: This process shall assess the transit driver's cost effectiveness when carrying out previous work assignments. The process shall perform this activity by 1) utilizing standard transit driver cost criteria set forth by governmental regulations and company policies, and 2) monitoring the transit driver's hourly wage and accumulated work hours. The data shall be sent to this process by the process that provides the interface to a local data store, each time that the store is updated with driver cost effectiveness data.

In Subsystem:

Public Transport Management

In Market Package:

APTS2

APTS3

Data Flows:

Input Flows

transit_driver_cost_effectiveness_considerations

Output Flows

transit_driver_cost_effectiveness
transit_driver_cost_effectiveness_data

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.3;
USR = 2.1.3.2;
USR = 2.1.3.2.2;
USR = 2.1.3.2.4;

4.5.4 Assess Transit Driver Eligibility

Description:

Overview: This process shall assess the transit driver's eligibility for future work assignments. The process shall carry out this activity by 1) monitoring the transit driver's performance, availability and cost effectiveness, 2) utilizing standardized transit driver eligibility criteria set forth by governmental regulations and company policies, and 3) ensuring that the transit driver has the required experience, education and certifications. The data shall be sent to this process in one of two ways: 1) by the process that provides the interface to a local data store, each time that the store is updated with driver eligibility data, or 2) the data is produced as the result of analysis work carried out by other processes within the Manage Traffic function.

In Subsystem:

Public Transport Management

In Market Package:

APTS2

APTS3

Data Flows:

Input Flows

transit_driver_availability_data
transit_driver_cost_effectiveness_data
transit_driver_eligibility_considerations
transit_driver_performance_data

Output Flows

transit_driver_eligibility
transit_driver_eligibility_data

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.3;
USR = 3.1.3.2;
USR = 2.1.3.2.2;
USR = 2.1.3.2.4;

4.5.5 Generate Transit Driver Route Assignments

Description:

Overview: This process shall assign transit drivers to transit schedules. The transit driver's eligibility, route preferences, seniority, and transit vehicle availability shall be used by the process to determine the transit driver's route assignment. The output produced by the process shall be sent to the transit driver in the form of the next work assignment. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Management

In Market Package:

APTS2

APTS3

Data Flows:

Input Flows

paratransit_services_for_transit_drivers
transit_driver_eligibility_data
transit_driver_route_assignment_considerations
transit_services_for_transit_drivers
transit_vehicle_availability

Input/Output

transit_driver_route_data

Output Flows

transit_route_assign_for_archive
ttd_route_assignments

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.3;
USR = 2.1.3.2;
USR = 2.1.3.2.2;
USR = 2.1.3.2.3;
USR = 2.1.3.2.3(a);
USR = 2.1.3.2.3(c);
USR = 2.1.3.2.3(d);
USR = 2.3.4;
USR = 2.3.4.3;

4.5.6 Update Transit Driver Information

Description:

Overview: This process shall provide the interface through which the transit driver can input data to the store of transit driver information. The interface provided by this process shall enable the transit driver to update personal availability and route assignment information. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Management

In Market Package:

APTS2

APTS3

Data Flows:

Input Flows

ftd_information_updates

Output Flows

transit_driver_consideration_inputs

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.3;
USR = 2.1.3.2;
USR = 2.1.3.2.4;

4.5.7 Report Transit Driver Information

Description:

Overview: This process shall provide the interface between the transit fleet manager and the store of driver information. The interface provided by the process shall enable the fleet manager to review and update transit driver information. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Management

In Market Package:

APTS2

APTS3

Data Flows:

Input Flows

ftfm_transit_driver_information_request
ftfm_transit_driver_information_updates
ftfm_transit_driver_route_preferences
transit_driver_information_output

Output Flows

transit_driver_consideration_updates
transit_driver_information_output_request
ttfm_transit_driver_information

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.3;
USR = 2.1.3.2;
USR = 2.1.3.2.4;

4.5.8 Provide Transit Driver Information Store Interface

Description:

Overview: This process shall provide the read and write interface to the store of transit driver information. The interface enables the contents of the store to be updated with inputs received from the transit driver and transit fleet manager via other processes, as well as, inputs resulting from analysis of driver availability, cost effectiveness, eligibility, and performance carried out by other processes. The process shall also supply data to these processes when the store is updated with information from the transit driver and fleet manager. It shall also supply data to the process that generates driver route assignments when any of the analysis inputs is received.

In Subsystem:

Public Transport Management

In Market Package:

APTS2

APTS3

Data Flows:

Input Flows

transit_driver_availability
transit_driver_consideration_inputs
transit_driver_consideration_updates
transit_driver_cost_effectiveness
transit_driver_eligibility
transit_driver_information_output_request
transit_driver_performance

Input/Output

transit_driver_information

Output Flows

transit_driver_availability_considerations
transit_driver_cost_effectiveness_considerations
transit_driver_eligibility_considerations
transit_driver_info_for_archive
transit_driver_information_output
transit_driver_performance_considerations
transit_driver_route_assignment_considerations

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.1.0;
USR = 2.1.3;
USR = 2.1.3.2;
USR = 2.1.3.2.1;
USR = 2.1.3.2.2;
USR = 2.1.3.2.3;
USR = 2.1.3.2.4;

4.6.1 Detect Transit User on Vehicle

Description:

Overview: This process shall detect embarking transit users on-board a transit vehicle and read data from the payment instrument that they are carrying. The process shall provide an image of all transit users which shall be used for violation processing of those who do not have a payment instrument or whose transit fare transaction fails. It shall obtain an image of the required accuracy under all lighting conditions and over the range of speeds with which transit users will pass through the fare collection point on a transit vehicle.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS4

Data Flows:

Input Flows

ftu_transit_user_vehicle_image
request_transit_user_vehicle_image
transit_user_vehicle_tag_data

Output Flows

transit_user_vehicle_image
transit_user_vehicle_tag_identity

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1.0;
USR = 3.1.1;
USR = 3.1.2;
USR = 3.1.2.7;
USR = 3.1.2.8;
USR = 3.1.4;
USR = 3.1.4.3;

4.6.2 Determine Transit User Needs on Vehicle

Description:

Overview: This process shall determine the transit user's travel routing based on the transit vehicle's current location and the user's destination. The process shall support the transit user's routing, enabling it to include travel on the vehicle for all or part of its route and (possibly) transfer to another vehicle on another route. In order to achieve this capability, the process shall have access to the complete range of transit services (routes and schedules) that are available to the transit user. The transit vehicle's location shall be provided by other processes within the Manage Transit function. Details of all transactions with the transit user's payment details removed, shall be sent by this process to the interface process for loading into a data store.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS4

Data Flows:

Input Flows

transit_advisory_vehicle_information
transit_services_for_vehicle_fares
transit_user_vehicle_information
transit_user_vehicle_tag_identity
transit_vehicle_location

Output Flows

transit_user_vehicle_ride
transit_user_vehicle_ride_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR - 3.1.2.7;
USR = 3.1.4;
USR = 3.1.4.3;

4.6.3 Determine Transit Fare on Vehicle

Description:

Overview: This process shall calculate the transit user's fare based on the origin and destination provided by the user. The process shall calculate the fare using the transit routing, transit fare category, and transit user history components of the ride data, in addition to information provided by the interface process for the transit fares data store. The accumulated data shall be sent by this process to another process for the actual implementation of the fare payment transaction.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS4

Data Flows:

Input Flows

transit_fares_for_vehicle
transit_user_vehicle_ride

Output Flows

transit_user_vehicle_fare

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.2;

4.6.4 Manage Transit Fare Billing on Vehicle

Description:

Overview: This process shall manage the transit user fare payments on-board a transit vehicle. The process shall receive information about the fare that is to be paid and the method of payment adopted by the transit user. It shall always support two modes of operation to complete the back end financial processing: infrastructure interactive, or semi-autonomous batch processing. The interactive method shall be used for individual transactions, such as those in paratransit type operations where value/volume ratios are high. It shall send transit user fare payment data to processes in the Provide Electronic Payment Services function for financial authorization and transaction processing, plus the return of the result for display to the transit user. A failed transaction shall result in the transmission of an image of the transit user to another process. Batch processing shall be used by the process for routes where value/volume ratios are low. It shall be performed using all the same data flows and processes as in the interactive method, except that transaction records are queued in a transaction buffer store which shall be maintained by this process. The accumulated data for the fare transactions shall be sent to the Provide Electronic Payment Services function on command from the transit vehicle driver, or when the transit vehicle has reached a convenient point on its route. The transit vehicle driver shall be notified when batch processing has completed successfully. In either mode of operation, a record of the status of all transit fare processing shall be sent to an interface process for the fare collection storage database.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS4

Data Flows:

Input Flows

bad_tag_list_update
bad_transit_tag_store_data
confirm_vehicle_fare_payment
ftd_fare_transaction_mode_set_up
ftd_request_batch_mode_data_transfer
ftu_transit_user_vehicle_image
transit_user_transaction_queue
transit_user_vehicle_fare
transit_user_vehicle_tag_identity

Output Flows

bad_tag_list_request
bad_transit_tag_store_check_request
bad_transit_tag_store_updates
fare_collection_vehicle_violation_information
request_vehicle_fare_payment
transit_user_transaction_record
transit_user_vehicle_payment_response
transit_user_vehicle_processed_fare_data
ttd_batch_mode_data_transfer_status
ttd_request_fare_transaction_mode_set_up
ttu_vehicle_access_message

User Service Requirements Mapped to Pspec:

USR = 2.3.0;
USR = 2.3.3;
USR = 2.3.3.1;
USR = 2.3.3.1(c);
USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.1;
USR = 3.1.2.2;
USR = 3.1.2.3;
USR = 3.1.2.4;
USR = 3.1.2.5;
USR = 3.1.2.6;

4.6.5 Provide Transit User Fare Payment Interface on Vehicle

Description:

Overview: This process shall provide the fare payment interface for the transit user on-board a transit vehicle. The process shall prompt the transit user for information necessary that has not been provided for the transaction. The result of the transit service ride fare payment plus other services request and payment, shall be reported back to the transit user by the process. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS4

Data Flows:

Input Flows

transit_user_vehicle_credit_identity
transit_user_vehicle_payment_response
transit_vehicle_advanced_payment_response
transit_vehicle_location

Output Flows

transit_user_advanced_payment_on_vehicle
transit_user_vehicle_information
transit_vehicle_advanced_payment_request
ttu_vehicle_payment_confirmed

User Service Requirements Mapped to Pspec:

USR = 2.2.1.2.2;
USR = 2.2.1.2.2.1;
USR = 2.2.1.2.2.2;
USR = 2.2.1.2.2.3;
USR = 2.2.1.2.2.4;
USR = 2.3.0;
USR = 2.3.3;
USR = 2.3.3.1;
USR = 2.3.3.1(c);
USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.3;

4.6.6 Update Transit Vehicle Fare Data

Description:

Overview: This process shall provide a database on-board the transit vehicle for use in fare processing. The database shall contain transit fare information from which the fares for all possible trips within the transit operational network can be determined.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS4

Data Flows:

Input Flows

transit_vehicle_fare_data

Output Flows

transit_fares_for_vehicle

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.6;

4.6.7 Provide Transit Vehicle Passenger Data

Description:

Overview: This process shall provide passenger loading and fare statistics data to other ITS functions. The process shall send the data automatically at regular periodic intervals using data collected in the store of fare transaction data. This store receives data from the process that interfaces to the user on-board a transit vehicle.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS4

Data Flows:

Input Flows

transit_user_vehicle_processed_fare_data
transit_user_vehicle_ride_data

Input/Output

transit_vehicle_fare_collection_data

Output Flows

transit_vehicle_passenger_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.2;
USR = 3.1.2.7;

4.6.8 Manage Transit Vehicle Advanced Payments

Description:

Overview: This process shall act as the interface for advanced payment of tolls and parking lot charges from the transit user. Requests for these advanced payments shall be passed to other processes in the Provide Electronic Payment Services function for transaction processing. The process shall ensure that the response to these requests from transit users is returned to the transit vehicle from which it was made.

In Subsystem:

Public Transport Management

In Market Package:

APTS4

APTS8

Data Flows:

Input Flows

advanced_tolls_and_charges_vehicle_confirm
transit_vehicle_advanced_payment_request

Output Flows

advanced_tolls_and_charges_vehicle_request
transit_vehicle_advanced_payment_response

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.3;

4.7.1.1 Provide Transit User Roadside Data Interface

Description:

Overview: This process shall communicate with the Transit Management Center (TRMC) by providing public transit information at roadside locations. These locations may consist of transit vehicle stops or other locations that provide general public transit information. The process shall enable the roadside unit to obtain information about the transit services on request from the local transit user interface process and to receive data about late running services from other processes within the Manage Transit function. The received data shall be loaded into a local data store for future use. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Remote Traveler Support

In Market Package:

APTS8

Data Flows:

Input Flows

ftu_transit_information_request
transit_services_for_travelers
transit_vehicle_arrival_time

Input/Output

transit_services_roadside_data

Output Flows

transit_services_travelers_request
ttu_transit_information

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.2.0;
USR = 2.2.1;
USR = 2.2.1.2;
USR = 2.2.1.2.1;
USR = 2.2.1.2.1.1;
USR = 2.2.1.2.1.1.1;
USR = 2.2.1.2.1.1.2;
USR = 2.2.1.2.1.1.2(a);
USR = 2.2.1.2.1.1.2(b);
USR = 2.2.1.2.1.1.3;
USR = 2.2.1.2.1.2;
USR = 2.2.1.2.1.3;
USR = 2.2.1.2.2;
USR = 2.2.1.2.2.1;
USR = 2.2.1.2.2.2;
USR = 2.2.1.2.2.3;
USR = 2.2.1.2.2.4;

4.7.1.2 Provide Transit User Roadside Vehicle Data Interface

Description:

Overview: This process shall provide the roadside (transit stop) interface through which transit users receive information about an approaching transit vehicle or one that has already arrived. The process shall output the data to the transit user as soon as it is received and shall load the data into the local store for future use. Output of the data shall be maintained until the vehicle leaves the stop, when the process shall cease output of the data and delete it from the local store. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Remote Traveler Support

In Market Package:

APTS8

Data Flows:

Input Flows

transit_vehicle_user_data

Input/Output

transit_services_roadside_data

Output Flows

ttu_transit_vehicle_information

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.2.0;
USR = 2.2.1;
USR = 2.2.1.2;
USR = 2.2.1.2.1;
USR = 2.2.1.2.1.1;
USR = 2.2.1.2.1.1.1;
USR = 2.2.1.2.1.1.2;
USR = 2.2.1.2.1.1.3;
USR = 2.2.1.2.1.2;
USR = 2.2.1.2.1.3;
USR = 2.2.1.2.2;
USR = 2.2.1.2.2.1;
USR = 2.2.1.2.2.2;
USR = 2.2.1.2.2.3;
USR = 2.2.1.2.2.4;

4.7.2.1 Detect Transit User at Roadside

Description:

Overview: This process shall detect transit users embarking at a roadside transit stop and read data from the payment instrument that they are carrying. The process shall provide an image of all transit users which shall be used for violation processing of those who do not have a payment instrument or whose transit fare transaction fails. It shall obtain an image of the required accuracy under all lighting conditions and over the range of speeds with which transit users will pass through the fare collection point at the roadside, i.e., a transit stop.

In Subsystem:

Remote Traveler Support

In Market Package:

APTS4

Data Flows:

Input Flows

ftu_transit_user_roadside_image
request_transit_user_roadside_image
transit_user_roadside_tag_data

Output Flows

transit_user_roadside_image
transit_user_roadside_tag_identity

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1.0;
USR = 3.1.1;
USR = 3.1.2;
USR = 3.1.2.7;
USR = 3.1.2.8;
USR = 3.1.4;
USR = 3.1.4.3;

4.7.2.2 Determine Transit User Needs at Roadside

Description:

Overview: This process shall determine the transit user's travel routing based on the user's destination and the location of the roadside transit stop from which the route request is being made. The process shall support the transit user's routing enabling it to include travel on all or part of the route(s) operating from the stop and (possibly) transfer to another route. In order for this to be achieved, the process requires access to the complete range of transit services (routes and schedules) that are available to the transit user. Details of all transactions with the transit user's payment details removed, shall be sent by this process to the interface process for loading into the transit roadside fare collection data store.

In Subsystem:

Remote Traveler Support

In Market Package:

APTS4

Data Flows:

Input Flows

transit_services_for_roadside_fares
transit_user_roadside_information
transit_user_roadside_tag_identity

Output Flows

transit_user_roadside_ride
transit_user_roadside_ride_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR - 3.1.2.7;
USR = 3.1.4;
USR = 3.1.4.3;

4.7.2.3 Determine Transit Fare at Roadside

Description:

Overview: This process shall calculate the transit user's fare based on the origin and destination provided by the user. The process shall calculate the fare using the transit routing, transit fare category, and transit user history components of the ride data together with data provided by the interface process to the database of transit fares. The accumulated data shall be sent by the process to another process for the actual implementation of the fare payment transaction.

In Subsystem:

Remote Traveler Support

In Market Package:

APTS4

Data Flows:

Input Flows

transit_fares_for_roadside
transit_user_roadside_ride

Output Flows

transit_user_roadside_fare

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.2;

4.7.2.4 Manage Transit Fare Billing at Roadside

Description:

Overview: This process shall generate the data necessary to enable the financial transaction between the transit user and the transit provider to be completed at the roadside, i.e., at a transit stop. The process shall accept and process current transit passenger fare collection information. The process shall perform the front end transaction between the transit user and the transit system, and use the infrastructure interactive mode of operation to complete the back end processing. This means that the process shall send data about each transaction to processes in the Provide Electronic Payment Services function for the back end financial authorization and transaction processing. The process shall then await the return of the result for display to the transit user before accepting the next transaction. A failed transaction shall result in the transmission of an image of the transit user to another process. A record of the status of all transit fare processing shall be sent to another process for storage in a fare collection database.

In Subsystem:

Remote Traveler Support

In Market Package:

APTS4

Data Flows:

Input Flows

confirm_roadside_fare_payment
ftu_transit_user_roadside_image
transit_user_roadside_fare
transit_user_roadside_tag_identity

Output Flows

fare_collection_roadside_violation_information
request_roadside_fare_payment
transit_user_roadside_payment_response
transit_user_roadside_processed_fare_data
ttu_roadside_access_message

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.1;
USR = 3.1.2.2;
USR = 3.1.2.3;
USR = 3.1.2.4;
USR = 3.1.2.5;
USR = 3.1.2.6;
USR = 3.1.2.7;

4.7.2.5 Provide Transit User Roadside Fare Interface

Description:

Overview: This process shall provide the interface for the transit user at the roadside, i.e., at a transit stop. The interface shall enable the transit user to specify the required destination of a transit service ride and request other (yellow pages) services. The process shall prompt the transit user for information necessary for the transaction that has not been provided. The result of the transit service ride fare payment plus other services request and payment, shall be reported back to the transit user by the process. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Remote Traveler Support

In Market Package:

APTS4

Data Flows:

Input Flows

advanced_tolls_and_charges_roadside_confirm
ftu_destination_at_roadside
ftu_other_services_roadside_request
other_services_roadside_response
transit_user_roadside_credit_identity
transit_user_roadside_payment_response

Output Flows

advanced_tolls_and_charges_roadside_request
other_services_roadside_request
transit_user_advanced_payment_at_roadside
transit_user_roadside_information
ttu_other_services_roadside_confirmed
ttu_roadside_payment_confirmed

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.3;

4.7.2.6 Update Roadside Transit Fare Data

Description:

Overview: This process shall provide a database at the roadside, i.e., a transit stop, for use in fare processing. The database shall contain transit fare information from which the fares for all possible trips within the transit operational network can be determined.

In Subsystem:

Remote Traveler Support

In Market Package:

APTS4

Data Flows:

Input Flows

transit_roadside_fare_data

Output Flows

transit_fares_for_roadside

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.6;

4.7.2.7 Provide Transit Roadside Passenger Data

Description:

Overview: This process shall create passenger loading and fare statistics data based upon data collected at the roadside and send this data to the store of transit operations data. The process may send the data at regular periodic intervals, on-demand, or through some other trigger mechanism. The process shall create its outputs using information collected in the store of fare transaction data. This data is received from other processes at the roadside, i.e., at a transit stop.

In Subsystem:

Remote Traveler Support

In Market Package:

APTS4

Data Flows:

Input Flows

transit_user_roadside_processed_fare_data
transit_user_roadside_ride_data

Input/Output

transit_roadside_fare_collection_data

Output Flows

transit_roadside_passenger_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.2;
USR = 3.1.2.7;

5.1.1 Identify Emergencies from Inputs

Description:

Overview: This process shall enable existing emergency centers to receive the calls, determine response requirements (enough to determine what responding agencies to notify), and route distress calls to those predesignated responding agencies. This process shall provide the identified emergency information in a standard format as required. This process receives emergency requests from the general public, public safety agencies, and other service providers (e.g., a Mayday service provider). Every set of emergency data received shall be assigned a level of confidence by the process depending on its source, so that the subsequent processes can assess the level of response to be provided. This process shall include verification, in that it shall determine if a number of inputs might all be referring to the same incident, then designate that incident in its notifications to the most appropriate responding agencies. By reconciling numerous reports and other collaborative information from the field (e.g.,CCTV images, reports from field staff), the verification function confirms the existence, location, and nature of a reported emergency.

In Subsystem:

Emergency Management

In Market Package:

CVO10

EM1

EM3

Data Flows:

Input Flows

cf_hazmat_route_information
emergency_request_personal_traveler_details
emergency_request_traveler_details
fep_planned_event_data
fets_caller_information
fets_incident_information
incident_alert
mayday_emergency_data
transit_emergency_data
transit_incident_details

Output Flows

tep_planned_event_confirmation
verified_emergency

User Service Requirements Mapped to Pspec:

USR = 4.5;
USR = 4.5.0;
USR = 4.5.3;
USR = 4.5.3.1;
USR = 5.0;
USR = 5.1;
USR = 5.2;

5.1.2 Determine Coordinated Response Plan

Description:

Overview: This process shall determine the appropriate response for a verified emergency. This process shall classify, prioritize, and respond to verified emergencies accordingly. This process shall also determine the appropriate response plan and activate any remote controlled functions requested by a basic_vehicle terminator through the vehicle_security_ status data flow. A detailed description of the emergency, and any request for remote controlled emergency system activity, and any suggested response plan shall be sent to other processes for implementation. The same information shall also be forwarded to other emergency centers (other EM) for information and possible action.

In Subsystem:

Emergency Management

In Market Package:

ATMS08

CVO10

CVO10

EM1

EM3

Data Flows:

Input Flows

emergency_service_allocation_data
foec_emergency_center_identity
foec_incident_details
foec_incident_response_coordination
verified_emergency

Output Flows

emergency_response_data_for_communications
emergency_response_data_for_management
emergency_service_allocation_data_request
toec_emergency_center_identity
toec_incident_details
toec_incident_response_coordination

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.1;
USR = 5.2;

5.1.3 Communicate Emergency Status

Description:

Overview: This process shall receive the emergency service response plans and the status of their implementation for dissemination to other ITS functions. That dissemination shall be subject to sanitization according to pre-arranged rules, implemented in this process. The process shall also read data about emergency responses from the emergency services action log. All data shall be communicated by the process in standard formats to travelers, drivers, and other ITS functions. In the case of in-vehicle, personal traveler, and transit emergencies, after each emergency becomes a verified incident, the data shall be sent as soon as new status or plan data is received. Dissemination shall be controlled according to rules determined in this process to limit the information transmitted to that information useful to the receiver. Emergency information that is received from the emergency telephone system or E911 operators, shall be disseminated only when the response plan data is first received. That has the effect of only disseminating data on incidents that have been verified, since only verified incidents will have response plans. The process shall also extract data from the emergency service action log on request from processes in other ITS functions, and from the emergency services operator. Communication to in-vehicle processes may include requests for additional information or a set of commands to the vehicle security system.

In Subsystem:

Emergency Management

In Market Package:

ATMS08

CVO10

CVO10

EM1

EM1

EM3

Data Flows:

Input Flows

detailed_emergency_status
emergency_response_data_for_communications
emergency_service_action_log
emergency_service_log_output_request
fm_emergency_information_request
incident_details_request
incident_information_request

Output Flows

emergency_data_request
emergency_request_driver_acknowledge
emergency_request_personal_traveler_acknowledge
emergency_request_traveler_acknowledge
emergency_request_vehicle_acknowledge
emergency_service_log_for_archive
emergency_service_log_output
incident_details
incident_information
tets_incident_acknowledge
tm_emergency_information
transit_incident_coordination_data

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.1;
USR = 5.2;

5.1.4 Manage Emergency Response

Description:

Overview: This process shall enable existing emergency centers to receive emergency calls, determine response requirements to the extent necessary to route the information, and route distress calls and emergency information to predesignated responding agencies and vehicles. All identified emergency information shall be provided by the process in a standard format as required. The process shall also communicate with commercial fleet managers to obtain details of cargo and other vehicle data where this will affect the response of the emergency services, e.g., in the case of a vehicle carrying a HAZMAT load. The current status of all emergency service responses shall be stored by the process in an action log, for access by the communications process.

In Subsystem:

Emergency Management

In Market Package:

ATMS08

CVO10

EM1

Data Flows:

Input Flows

cf_hazmat_vehicle_information
emergency_response_data_for_management
emergency_service_allocation_override
emergency_vehicle_acknowledge
emergency_vehicle_dispatch_status
fws_current_weather
fws_predicted_weather
incident_command_request
incident_response_clear
incident_status_data
incident_video_for_emergency_services
maintenance_incidents_for_emergencies
resource_deployment_status
traffic_data_for_emergency_services
transit_coordination_data
wrong_way_vehicle_detection

Output Flows

cf_hazmat_request
detailed_emergency_status
emergency_service_action_log
emergency_service_allocations
emergency_vehicle_dispatch_failure
emergency_vehicle_incident_details
emergency_vehicle_response_request
incident_response_status
local_decision_support
remote_video_image_control
resource_request

User Service Requirements Mapped to Pspec:

USR = 4.5;
USR = 4.5.0;
USR = 4.5.2;
USR = 4.5.2.1;
USR = 4.5.2.2;
USR = 4.5.2.3;
USR = 4.5.2.3(a);
USR = 4.5.2.3(b);
USR = 4.5.2.3(c);
USR = 4.5.2.3(e);
USR = 4.5.2.3(f);
USR = 4.5.2.3(g);
USR = 4.5.2.3(h);
USR = 4.5.3;
USR = 4.5.3.3;
USR = 4.5.3.4;
USR = 5.0;
USR = 5.2;
USR = 5.2.1;
USR = 5.2.1.1;
USR = 5.2.1.2;
USR = 5.2.1.3;
USR = 5.2.2;
USR = 5.2.2.1;

5.1.5 Manage Emergency Service Allocation Store

Description:

Overview: This process shall manage the store of data that defines the way in which the emergency service resources shall be deployed in response to emergencies. Deployment shall vary by certain criteria, such as, type of emergency, source of information, time of day, location, etc. Parameters to define this allocation shall be loaded into the data store following receipt from the process that provides the emergency services operator interface.

In Subsystem:

Emergency Management

In Market Package:

ATMS08

CVO10

EM1

Data Flows:

Input Flows

emergency_service_allocation_data_output_request
emergency_service_allocation_data_request
emergency_service_allocation_data_updates

Input/Output

emergency_service_allocation_criteria

Output Flows

archive_provide_emergency_service_allocation_data
emergency_service_allocation_data
emergency_service_allocation_data_output

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.2;
USR = 2.2.1;
USR = 2.2.1.1;
USR = 2.2.1.1.4;
USR = 2.3;
USR = 2.3.4;

5.1.6 Process Mayday Messages

Description:

Overview: This process shall receive mayday messages from vehicles and drivers, determine whether the mayday message indicates an emergency that requires the attention of public safety agencies, and forward mayday emergency data to the appropriate agency when assistance is required. The content of the data flow 'mayday emergency data' shall include all the key data from the incoming data flow 'emergency request details' and an agency ID indicating the mayday provider that received and processed the mayday message. While not depicted in the logical architecture, the process will also be heavily dependent on voice communications to better ascertain the nature and severity of the emergency and to report this information to the appropriate local agency. This process shall also receive and keep a historical log of signals sent in the tracking_vehicle data flow.

In Subsystem:

Emergency Management

In Market Package:

CVO10

EM3

Data Flows:

Input Flows

emergency_request_driver_details
emergency_request_vehicle_details
foec_mayday_emergency_data

Input/Output

mayday_vehicle_tracking

Output Flows

mayday_emergency_data
toec_mayday_emergency_data

User Service Requirements Mapped to Pspec:

USR = 4.5;
USR = 4.5.0;
USR = 4.5.3;
USR = 4.5.3.1;
USR = 5.0;
USR = 5.1;
USR = 5.2;

5.2 Provide Operator Interface for Emergency Data

Description:

Overview: This process shall provide the emergency services operator with an interface to the other processes in the Manage Emergency Services function. The process shall enable the operator to review and update the data used to allocate emergency services to incidents, applying temporary overrides to current emergency service allocations to suit the special needs of a current incident, and requesting output of the log of emergency service actions. It shall also enable the output of a message showing the failure of an emergency vehicle dispatched in response to an incident. This output shall override all other outputs. The process shall support inputs from the emergency services operator in both manual and audio form, and shall provide its outputs in audible and visual forms. The visual output may appear in either hardcopy or as a display, or both, and an audible output shall accompany the emergency vehicle dispatch failure message.

In Subsystem:

Emergency Management

In Market Package:

ATMS08

CVO10

CVO10

EM1

EM1

EM2

EM3

Data Flows:

Input Flows

emergency_service_allocation_data_output
emergency_service_allocations
emergency_service_log_output
emergency_vehicle_dispatch_failure
feso_emergency_action_log_request
feso_emergency_allocation_override
feso_emergency_data_input
feso_emergency_data_output_request
feso_emergency_display_update_request
map_data_for_emergency_display

Output Flows

emergency_service_allocation_data_output_request
emergency_service_allocation_data_updates
emergency_service_allocation_override
emergency_service_log_output_request
request_emergency_display_update
teso_emergency_action_log_output
teso_emergency_data_output
teso_emergency_vehicle_dispatch_failure

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.1;
USR = 5.2;

5.3.1 Select Response Mode

Description:

Overview: This process shall select the appropriate emergency services and their vehicle(s) to respond to incidents. The process shall determine the type and number of vehicles to be dispatched, and provide the vehicle(s) with information on the type and location of the incident. It shall request data about the vehicles that are available from the interface process to the data store of emergency vehicle status. Once the vehicle determination has been made, the status data shall be changed by the process, and incident data sent to the process responsible for the actual dispatch of the vehicle(s).

In Subsystem:

Emergency Management

In Market Package:

ATMS08

CVO10

EM1

Data Flows:

Input Flows

emergency_vehicle_incident_details
emergency_vehicle_response_request
emergency_vehicle_status_data_for_responses

Output Flows

emergency_vehicle_dispatch_data
emergency_vehicle_dispatch_status
emergency_vehicle_status_data_change
emergency_vehicle_status_data_request

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.2;
USR = 5.2.1;
USR = 5.2.1.2;
USR = 5.2.1.3;

5.3.2 Dispatch Vehicle

Description:

Overview: This process shall direct selected emergency vehicles and drivers to respond to an incident, receive acknowledgment that they will in fact respond, and provide them with the location and details of the incident that was pre-calculated and sent to this process. If called for, the process shall send details to the Manage Traffic function to request a traffic control preemption be provided for the vehicle(s) if that mode of priority is available and chosen. The data for the emergency vehicle driver shall be sent to the driver interface process.

In Subsystem:

Emergency Management

In Market Package:

EM2

Data Flows:

Input Flows

emergency_traffic_control_response
emergency_vehicle_dispatch_data
emergency_vehicle_dispatch_response
emergency_vehicle_route
emergency_vehicle_status_data_for_dispatch

Output Flows

emergency_traffic_control_request
emergency_vehicle_dispatch_request
emergency_vehicle_route_request
emergency_vehicle_suggested_route

User Service Requirements Mapped to Pspec:

USR = 1.6;
USR = 1.6.3;
USR = 1.6.3.2;
USR = 1.6.3.2.2;
USR = 1.6.3.2.2(c);
USR = 4.5;
USR = 4.5.0;
USR = 4.5.3;
USR = 4.5.3.2;
USR = 5.0;
USR = 5.2;
USR = 5.2.1;
USR = 5.2.1.2;
USR = 5.2.1.3;
USR = 5.2.2;

5.3.3 Track Vehicle

Description:

Overview: This process shall manage information about the location of all emergency vehicles available for dispatch and that have been dispatched, and the ETA for vehicles en route. The process shall send this data to the store of emergency vehicle status data. If the vehicle is on its way to an emergency, as indicated by the received vehicle status, the process shall also send data to processes in the Manage Traffic function that will enable the vehicle to have whatever level and mode of priority is available and granted at traffic signals.

In Subsystem:

Emergency Vehicle Subsystem

In Market Package:

EM2

Data Flows:

Input Flows

vehicle_location_for_emergency_services

Output Flows

emergency_vehicle_preemptions
emergency_vehicle_tracking_data

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.2;
USR = 5.2.1;
USR = 5.2.1.1;

5.3.4 Assess Response Status

Description:

Overview: This process shall assess the status of emergency vehicles that are responding to an incident. In making its assessment, the process shall use data from the process managing a store of vehicle status, plus data from the emergency vehicle driver interface process. The process shall send the results of the assessment to the process responsible for managing emergency and emergency response information and update the store of vehicle status.

In Subsystem:

Emergency Management

In Market Package:

ATMS08

CVO10

EM1

Data Flows:

Input Flows

emergency_vehicle_route_assignment
emergency_vehicle_status_data_for_assessment
incident_status_update

Output Flows

emergency_vehicle_acknowledge
emergency_vehicle_status_data_needed
emergency_vehicle_status_data_update

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.2;
USR = 5.2.1;

5.3.5 Provide Emergency Personnel Interface

Description:

Overview: This process shall provide an interface for emergency personnel, through which data can be exchanged with other processes in the Manage Emergency Services function. It shall support the exchange of incident data to which responses are being made by emergency personnel. The process shall support inputs from emergency personnel in both audible and manual forms, with outputs being available in both audio or visual forms. The visual form may include display and hardcopy formats. Both inputs and outputs shall be provided in such a way that while alerting the driver to the information they contain, they shall in no way impair the driver's ability to operate the vehicle in a safe manner.

In Subsystem:

Emergency Vehicle Subsystem

In Market Package:

EM1

EM2

Data Flows:

Input Flows

emergency_vehicle_dispatch_request
emergency_vehicle_suggested_route
fep_emergency_dispatch_acknowledge
fep_incident_command_request
fep_incident_status
local_decision_support

Output Flows

emergency_vehicle_dispatch_response
incident_command_request
incident_status_data
incident_status_update
tep_decision_support
tep_emergency_dispatch_order

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.2;
USR = 5.2.2;
USR = 5.2.2.2;

5.3.6 Maintain Vehicle Status

Description:

Overview: This process shall maintain a data store of the current status of all emergency vehicles available for dispatch and that have been dispatched. It shall provide data from the store on request from other processes and shall update the contents of the store with new data received from other processes. The process shall output the status of a vehicle to the process responsible for vehicle tracking for as long as it is on its way to an incident, to update ETA estimates and enable local vehicle priority to be given at intersections, if that mode of priority is chosen and granted.

In Subsystem:

Emergency Management

In Market Package:

EM2

Data Flows:

Input Flows

emergency_vehicle_status_data_change
emergency_vehicle_status_data_needed
emergency_vehicle_status_data_request
emergency_vehicle_status_data_update
emergency_vehicle_tracking_data

Input/Output

emergency_vehicle_status_data

Output Flows

archive_manage_emergency_vehicle_data
emergency_vehicle_status_data_for_assessment
emergency_vehicle_status_data_for_dispatch
emergency_vehicle_status_data_for_responses

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.2;
USR = 5.2.1;
USR = 5.2.1.1;

5.3.7 Provide Emergency Vehicle Route

Description:

Overview: This process shall calculate and assign emergency vehicle routes for incident assistance upon request.

In Subsystem:

Emergency Management

In Market Package:

EM2

Data Flows:

Input Flows

emergency_vehicle_route_request
fmup_emergency_route_map_update

Output Flows

emergency_vehicle_route
emergency_vehicle_route_assignment
tmup_emergency_route_map_request

User Service Requirements Mapped to Pspec:

USR = 1.6;
USR = 1.6.3;
USR = 1.6.3.2;
USR = 1.6.3.2.2;
USR = 1.6.3.2.2(c);
USR = 4.5;
USR = 4.5.0;
USR = 4.5.3;
USR = 4.5.3.2;
USR = 5.0;
USR = 5.2;
USR = 5.2.1;
USR = 5.2.1.2;
USR = 5.2.1.3;
USR = 5.2.2;

5.4.1 Process TM Detected Violations

Description:

Overview: This process shall manage the details of high occupancy vehicle (HOV) lane use, wrong-way vehicle detection in reversible lanes, and pollution violations reported by the Manage Traffic function. The process shall use the parameters in the store of traffic management (TM) violation (enforcement) data to obtain the vehicle registration data from the appropriate State Department of Motor Vehicles (DMV) office, before sending all of the received information to the correct law enforcement agency. This process shall also maintain the TM enforcement data store, entering all information received from other processes.

In Subsystem:

Traffic Management

In Market Package:

ATMS05

ATMS17

Data Flows:

Input Flows

fdmv_traffic_violation_state_identity
fdmv_traffic_violation_vehicle_registration
hov_lane_violation
vehicle_pollution_alert

Input/Output

enforcement_data_for_TM

Output Flows

tdmv_traffic_violation_identity_code
tdmv_traffic_violation_vehicle_license
tea_traffic_violation_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.6;
USR = 1.6.0;
USR = 1.6.2;
USR = 1.6.2.4;
USR = 1.6.2.4.1;
USR = 1.6.3;
USR = 1.6.3.2;
USR = 1.6.3.2.2;
USR = 1.6.3.2.2(b);

5.4.2 Process Violations for Tolls

Description:

Overview: This process shall manage the details of toll payment violations reported by the Provide Electronic Payments Services function. The process shall use the parameters in the store of toll payment violation (enforcement) data to obtain the vehicle registration data from the appropriate State Department of Motor Vehicles (DMV) office (or alternate source) for vehicles that are not equipped with a tag, before sending all of the received information to the correct law enforcement agency. This process shall also maintain the toll payment enforcement datastore, entering all information received from other processes.

In Subsystem:

Toll Administration

In Market Package:

ATMS10

Data Flows:

Input Flows

fdmv_toll_violation_state_identity
fdmv_toll_violation_vehicle_registration
toll_violation_information

Input/Output

enforcement_data_for_tolls

Output Flows

tdmv_toll_violation_identity_code
tdmv_toll_violation_vehicle_license
tea_toll_violation_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.4;

5.4.3 Process Parking Lot Violations

Description:

Overview: This process shall manage the details of parking lot payment violations reported by the Provide Electronic Payment Services function. The process shall use the parameters in the store of parking lot violation (enforcement) data to obtain the vehicle registration data from the appropriate State Department of Motor Vehicles (DMV) office (or alternate source) for vehicles that are not equipped with a tag, before sending all of the received information to the correct law enforcement agency. This process shall also maintain the store of parking lot violation (enforcement) data, entering all information received from other processes.

In Subsystem:

Parking Management

In Market Package:

ATMS16

Data Flows:

Input Flows

fdmv_parking_lot_violation_state_identity
fdmv_parking_lot_violation_vehicle_registration
parking_lot_violation_information

Input/Output

enforcement_data_for_parking

Output Flows

tdmv_parking_lot_violation_identity_code
tdmv_parking_lot_violation_vehicle_license
tea_parking_violation_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.4;

5.4.4 Process Fare Payment Violations

Description:

Overview: This process shall manage the details of fare payment violations reported by the Provide Electronic Payments function. The process shall use the parameters in the store of fare payment violation (enforcement) data to process and send the data to the correct law enforcement agency. This process shall also maintain the fare payment enforcement data store, entering all information received from other processes.

In Subsystem:

Public Transport Management

In Market Package:

APTS4

Data Flows:

Input Flows

fare_violation_information

Input/Output

enforcement_data_for_fare_payment

Output Flows

bad_transit_collected_fare_payment
tea_fare_payment_violation_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.4;

5.4.5 Process Vehicle Fare Collection Violations

Description:

Overview: This process shall manage the details of fare collection violations reported by the Manage Transit function that have taken place on-board a transit vehicle. The process shall use the parameters in the store of vehicle fare collection violation (enforcement) data to process and send the information to the correct law enforcement agency. This process shall also maintain the vehicle fare collection enforcement data store, entering all information received from other processes.

In Subsystem:

Public Transport Management

In Market Package:

APTS4

Data Flows:

Input Flows

fare_collection_vehicle_violation_information

Input/Output

enforcement_data_for_vehicle_fare_collection

Output Flows

bad_transit_vehicle_fare_payment
tea_fare_collection_vehicle_violation_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.4;

5.4.6 Process CV Violations

Description:

Overview: This process shall manage the details of violations committed by commercial vehicles, their drivers and/or operators, reported by the Manage Commercial Vehicles function. The process shall use the parameters in the store of commercial vehicle violation (enforcement) data to obtain the vehicle registration data from the appropriate State Department of Motor Vehicles (DMV) office, before sending all of the received data to the correct law enforcement agency. This process shall also maintain the commercial vehicle violation (enforcement) data store.

In Subsystem:

Commercial Vehicle Administration

In Market Package:

CVO04

Data Flows:

Input Flows

cv_violation_data
fdmv_cv_violation_state_identity
fdmv_cv_violation_vehicle_registration

Input/Output

enforcement_data_for_cv

Output Flows

tdmv_cv_violation_identity_code
tdmv_cv_violation_vehicle_license
tea_cv_violation_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.4;

5.4.7 Process Roadside Fare Collection Violations

Description:

Overview: This process shall manage the details of fare collection violations reported by the Manage Transit function that have taken place at the roadside, i.e., at a transit stop. The process shall use the parameters in the store of roadside fare collection violation (enforcement) data to process and send the information to the correct law enforcement agency. This process shall also maintain the roadside fare collection enforcement data store, entering all information received from other processes.

In Subsystem:

Public Transport Management

In Market Package:

APTS4

Data Flows:

Input Flows

fare_collection_roadside_violation_information

Input/Output

enforcement_data_for_roadside_fare_collection

Output Flows

bad_transit_roadside_fare_payment
tea_fare_collection_roadside_violation_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.4;

5.5 Update Emergency Display Map Data

Description:

Overview: This process shall provide updates to the store of digitized map data used as the background for displays of incidents and emergencies produced by processes in the Manage Emergency Services function. The process shall obtain the new data from a specialist data supplier or some other appropriate data source, on receiving an update request from the emergency system operator interface process within the Manage Emergency Services function.

In Subsystem:

Emergency Management

In Market Package:

ATMS08

CVO10

EM1

EM2

Data Flows:

Input Flows

fmup_emergency_display_update
request_emergency_display_update

Output Flows

map_data_for_emergency_display
tmup_request_emergency_display_update

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.1;
USR = 5.2;

5.6 Manage Emergency Services Data

Description:

Overview: This process shall collect emergency service data, emergency vehicle management data, emergency vehicle data, and incident data. It shall distribute this data to the Manage Archive Data Request where it can be archived and accessed upon request or upon receipt of fresh data.

In Subsystem:

Emergency Management

In Market Package:

AD1

AD2

Data Flows:

Input Flows

archive_manage_emergency_vehicle_data
archive_provide_emergency_service_allocation_data
em_archive_request
em_archive_status
emergency_service_log_for_archive
feso_archive_commands

Input/Output

emergency_data_archive

Output Flows

em_archive_data
teso_archive_status

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.5;
USR = 7.1.3.1.5(a);
USR = 7.1.3.1.5(b);
USR = 7.1.3.1.5(c);
USR = 7.1.3.1.5(d);
USR = 7.1.3.1.5(f);
USR = 7.1.3.1.5(h);

5.7.1 Collect Disaster Response Data

Description:

Overview: This process is responsible for gathering data to support the Coordinate Disaster Response function. The data gathered by this process includes incident information, medical facility information, and other emergency data. This process receives request for information concerning disasters to coordinate the response to disaster-type emergencies with the Disaster Command Authority. This data is collected and forwarded to other processes that archive the information or disseminate it to other agencies. This process shall send dispatch information to other processes within the Manage Emergency Services function based on inputs from the Disaster Command Authority.

Functional Requirements: none.

In Subsystem:

Emergency Management

In Market Package:

EM4

Data Flows:

Input Flows

disaster_data
disaster_emergency_response_request
emergency_input_for_disaster
medical_facility_data

Output Flows

disaster_archive_data
disaster_data
disaster_emergency_response_dispatch
emergency_disaster_incident_data

User Service Requirements Mapped to Pspec:

USR = C6.3.1.1;
USR = C6.3.1.2;
USR = C6.3.1.5;

5.7.2 Provide Medical Facility Interface

Description:

Overview: This process is responsible for managing the interface between the Manage Emergency Services function and the Medical Facility terminator. This process shall respond to requests for availability data from the Provide Disaster Coordination Authority Interface process and forward the request onto the Medical Facility terminator. Upon receipt of availability data from the medical facility this process shall pass the information back to the Provide Disaster Coordination Authority Interface function and the Collect Disaster Response Data. This process supports the coordination of transportation to medical facilities based on the level of service supported by the facility and their availability. In places where medical facilities are also used as emergency shelters the information provided through this process shall also support the coordination of evacuation plans. The data provided by this process shall support the dissemination to travellers as well as other emergency services within this function. This process shall support both the receipt data on a continual basis or upon demand based upon requests from other functions.

Functional Requirements: none.

In Subsystem:

Emergency Management

In Market Package:

EM4

EM5

Data Flows:

Input Flows

fmf_medical_facility_availability
medical_facility_disaster_request

Output Flows

medical_facility_data
medical_facility_disaster_response
tmf_medical_facility_request

User Service Requirements Mapped to Pspec:

USR = C6.3.1.2;
USR = C6.3.1.3;
USR = C6.3.2.2;

5.7.3 Provide Disaster Coordination Authority Interface

Description:

Overview: This process is responsible for managing the interface between the Manage Emergency Services function and the Disaster Command Authority terminator. This process shall respond to requests for availability data, situation reports on current disasters, shelter availability and locations, as well as inputs to manage how the local emergency services function should address the situation. This process shall forward the requests for information onto the appropriate functions. Upon receipt of the required data about medical facilities or emergency services or traffic data this process shall forward the data to the Disaster Command Authority.

Functional Requirements: none.

In Subsystem:

Emergency Management

In Market Package:

EM4

Data Flows:

Input Flows

emergency_disaster_incident_data
fdca_disaster_response_input
fdca_disaster_shelter_availability
fdca_disaster_shelter_location
fdca_medical_facility_request
fdca_situation_report
medical_facility_disaster_response

Output Flows

disaster_emergency_response_request
medical_facility_disaster_request
tdca_disaster_emergency_plan
tdca_disaster_incident_data
tdca_medical_facility_availability

User Service Requirements Mapped to Pspec:

USR = C6.3.1.1;
USR = C6.3.1.2;
USR = C6.3.1.3;

6.1.1 Provide Trip Planning Information to Traveler

Description:

Overview: This process shall obtain all the information needed to fulfill the traveler's request for a trip. The process shall support the request for trips that require the use of one or more modes of transport, and shall use the preferences and constraints specified by the traveler in the trip request, plus data from the store of trip planning parameters, to select the most appropriate modes. It shall send details of the trip requirements to the specialized processes that provide route information for the different modes of transport. When route data is received back from these processes, this process shall ensure that the whole trip is covered by one coherent route for which all the data such as costs, arrival times, and
modal change points are known. The information provided to the traveler by the process shall be sufficient to enable the traveler to understand the routing, modes and cost of the trip. The trip information shall be stored for possible use in subsequent trip confirmation. The process also includes parking lot data. This data is used in transactions requiring electronic payment of parking lot services, as well as for a traveler making a parking lot reservation. This process shall exchange all input and output data from and to the traveler with the appropriate traveler interface process. The traveler shall send parking lot data, traveler trip requests, and traveler current condition requests to the archival process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS6

ATIS7

ATIS8

ATIS1

ATIS2

ATIS4

ATIS5

ATIS8

Data Flows:

Input Flows

current_conditions
multimodal_service_data_response
paratransit_personal_schedule
paratransit_route_request
parking_lot_availability
prices
rideshare_response
supplied_route
traveler_current_condition_request
traveler_personal_current_condition_request
traveler_personal_trip_request
traveler_trip_request
trip_planning_parameters

Output Flows

multimodal_service_data_request
paratransit_route_response
paratransit_trip_request
parking_lot_data_for_archive
parking_lot_data_request
request_prices
traveler_personal_trip_information
traveler_rideshare_request
traveler_trip_and_cond_requests_for_archive

traveler_trip_information
trip_information
trip_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.1.0;
USR = 1.1.1;
USR = 1.1.1.1;
USR = 1.1.1.1.1;
USR = 1.1.1.1.2;
USR = 1.1.1.1.3;
USR = 1.1.1.1.4;
USR = 1.1.1.1.5;
USR = 1.1.1.1.6;
USR = 1.1.2;
USR = 1.1.2.1;
USR = 1.1.2.1.1;
USR = 1.1.2.1.2;
USR = 1.1.2.1.3;
USR = 1.1.2.1.4;
USR = 1.1.2.1.5;
USR = 1.1.2.1.6;
USR = 1.1.2.1.7;
USR = 1.1.2.1.8;
USR = 1.1.3;
USR = 1.1.3.1;
USR = 1.1.3.1.1;
USR = 1.1.3.1.2;
USR = 1.1.3.1.3;
USR = 1.1.3.1.4;
USR = 1.1.3.2;
USR = 1.1.3.3;
USR = 1.1.3.3.1;
USR = 1.1.3.3.2;
USR = 1.1.4;
USR = 1.1.4.1;
USR = 1.1.4.1.1;
USR = 1.1.4.1.2;
USR = 1.1.4.1.3;
USR = 1.4.0;
USR = 1.4.1;
USR = 1.4.1.3;
USR = 1.4.3;
USR = 1.4.3.1;
USR = 1.4.3.3;
USR = 1.4.3.3(a);
USR = 1.4.3.3(b);
USR = 1.4.3.3(c);
USR = 1.4.4.3(d);
USR = 1.4.4.3(e);
USR = 1.4.4.3(f);
USR = 1.5.0;
USR = 1.5.2;
USR = 1.5.2.2;
USR = 1.5.2.2(d);
USR = 1.5.2.2(e);
USR = 1.5.2.2(f);
USR = 1.5.2.2(g);
USR = 1.5.2.2(h);
USR = 1.6;
USR = 1.6.4;

USR = 1.6.4(b);
USR = 1.6.4(c);
USR = 1.7;
USR = 1.7.1;
USR = 1.7.1.1;
USR = 1.7.1.1.1;
USR = 1.7.1.1.1(f);
USR = 1.7.1.1.1(g);
USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.2;
USR = 1.8.1.2(d);
USR = 1.8.1.2(e);
USR = 1.8.1.3;
USR = 1.8.1.3(a);
USR = 1.8.1.3(e);
USR = 1.8.1.3(f);
USR = 1.8.1.6;
USR = 1.8.1.6(a);
USR = 1.8.2;
USR = 1.8.2.3;
USR = 1.8.2.3(c);
USR = 1.8.2.4;
USR = 1.8.2.4(a);
USR = 1.8.2.4(f);
USR = 2.0;
USR = 2.3.0;
USR = 2.3.1;
USR = 2.3.1.2;
USR = 2.4.0;
USR = 2.4.3;
USR = 2.4.3.1;
USR = 2.4.3.2;
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;
USR = 7.1.3.1.8(e);
USR = 7.1.3.1.8(g);

6.1.2 Confirm Traveler's Trip Plan

Description:

Overview: This process shall confirm a trip previously requested by a traveler and any financial transactions that this may require. The process shall base the trip confirmation upon information created by the process responsible for trip planning and stored locally. Confirmation details shall be sent to specialized processes (such as those responsible for demand responsive transit, ridesharing, etc.) to make reservations for their services. The response to these reservation requests and any necessary payment transactions shall be sent to the traveler. This process shall exchange all input and output data to and from the traveler via the appropriate traveler interface process. The trip confirmation shall be sent to the archival process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS6

ATIS7

ATIS8

ATIS8

ATIS2

ATIS5

Data Flows:

Input Flows

multimodal_service_confirmation
paratransit_route_confirm
parking_lot_reservation_confirm
rideshare_confirmation
traveler_payment_information
traveler_payment_response
traveler_personal_payment_information
traveler_personal_trip_confirmation
traveler_trip_confirmation
trip_information
trip_planning_parameters

Output Flows

multimodal_service_confirm
paratransit_service_confirmation
parking_lot_reservation_request
traveler_confirm_for_archive
traveler_payment_confirmation
traveler_payment_request
traveler_personal_payment_confirmation
traveler_personal_transaction_confirmation
traveler_rideshare_confirmation
traveler_transaction_confirmation

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.1.0; USR= 1.1.3;
USR = 1.1.3.1;
USR = 1.1.3.2;
USR = 1.1.3.3;
USR = 1.1.4;
USR = 1.1.4.1;
USR = 1.1.4.1.1;
USR = 1.1.4.1.2;
USR = 1.1.4.1.3;
USR = 1.4.0;
USR = 1.4.1;
USR = 1.4.1.1;
USR = 1.4.1.2;
USR = 1.4.1.2(a);
USR = 1.4.1.3;
USR = 1.4.1.4;
USR = 1.4.2;
USR = 1.4.2.1;
USR = 1.4.2.4;
USR = 1.4.3;
USR = 1.4.3.2;
USR = 1.5.0;
USR = 1.5.1;
USR = 1.5.1.4;
USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.6;
USR = 1.8.1.6(a);
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;

6.1.3 Manage Multimodal Service Provider Interface

Description:

Overview: This process shall collect data about services that are available to travelers from multimodal service providers. These suppliers shall be those that provide travel services that are not part of regular transit or demand responsive transit operations, e.g. heavy rail, and may not involve surface transportation, e.g. ferry and airline operations. The process shall provide data formatted for use as part of a traveler's multimodal trip, and shall support subsequent confirmation of any portion provided by the Multimodal Service Provider.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

ATIS2

Data Flows:

Input Flows

fmtsp_air_services
fmtsp_ferry_services
fmtsp_multimodal_service_confirmation
fmtsp_rail_services
multimodal_service_confirm
multimodal_service_data_request

Input/Output

multimodal_service_details_data

Output Flows

multimodal_service_confirmation
multimodal_service_data_response
tmtsp_air_services_request
tmtsp_confirm_multimodal_service
tmtsp_ferry_services_request
tmtsp_rail_services_request

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.2.0;
USR = 2.2.1;
USR = 2.2.1.1.3;
USR = 2.2.1.1.4;

6.1.4 Provide ISP Operator Interface for Trip Planning Parameters

Description:

Overview: This process shall manage the data store containing parameters used by the trip planning processes. These parameters shall govern the way in which multimodal trips are planned by other processes within Provide Trip Planning Services. This process shall accept inputs from the ISP Operator to define or update trip planning parameters. This process shall output these trip planning parameters to the ISP Operator.

In Subsystem:

Information Service Provider

In Market Package:

ATIS8

ATIS2

Data Flows:

Input Flows

fispo_trip_planning_parameters_request
fispo_trip_planning_parameters_update

Input/Output

trip_planning_parameters

Output Flows

tispo_trip_planning_parameters

User Service Requirements Mapped to Pspec:

USR = 1.6;
USR = 1.6.4;
USR = 1.6.4(b);
USR = 1.6.4(c);

6.1.5 Collect Service Requests and Confirmation for Archive

Description:

Overview: This process shall receive all traveler requests, such as requests for traffic and transit information, requests for current conditions such as weather, trip requests, guidance route requests, advisory requests, yellow page information requests, and service confirmations. These requests shall be stored in the service_req_and_confirm_data data store and output to the traveler information data archive. The process shall run when a new request or confirmation is received from an external source.

In Subsystem:

Information Service Provider

In Market Package:

AD1

AD2

Data Flows:

Input Flows

advisory_data_request_for_archive
traffic_data_kiosk_request_for_archive
traffic_data_personal_request_for_archive
transit_deviation_kiosk_request_for_archive
transit_deviations_personal_request_for_archive
traveler_confirm_for_archive
traveler_route_accepted_for_archive
traveler_route_request_for_archive
traveler_trip_and_cond_requests_for_archive
traveler_yellow_pages_requests_for_archive
vehicle_guidance_route_accepted_for_archive
vehicle_route_request_for_archive
yellow_pages_advisory_requests_for_archive

Input/Output

service_req_and_confirm_data

Output Flows

service_req_and_confirm_for_archive

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;
USR = 7.1.3.1.8(g);
USR = 7.1.3.1.8(h);

6.1.6 Manage Traveler Info Archive Data

Description:

Overview: This process shall accept traveler information service requests and confirmations, parking management information, payment transaction data, rideshare requests, commercial and non-commercial probe data, route guidance data, and origin/destination data, and store it in its local traveler info data archive data store, together with a catalog to describe the data. When requested by the Manage Archive Data function, this information will be sent to that function. The process shall also provide a control interface to the ISP Operator, responding with the status received from the requester of the archive. The process shall run when a request for data or a catalog is received from an external source, when a command is received from the ISP Operator, or when fresh data is received.

In Subsystem:

Information Service Provider

In Market Package:

AD1

AD2

Data Flows:

Input Flows

current_other_routes_use_for_archive
current_road_network_use_for_archive
fispo_archive_commands
parking_lot_data_for_archive
rideshare_data_for_archive
route_guidance_data_for_archive
service_req_and_confirm_for_archive
traveler_archive_request
traveler_archive_status
traveler_info_payments_transactions
traveler_rideshare_request_for_archive
trip_request_for_archive
vehicle_guidance_probe_data_for_archive

Input/Output

traveler_info_data_archive

Output Flows

tispo_archive_status
traveler_archive_data

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.4;
USR = 7.1.3.1.4(c);
USR = 7.1.3.1.8;
USR = 7.1.3.1.8(a);
USR = 7.1.3.1.8(c);
USR = 7.1.3.1.8(d);
USR = 7.1.3.1.8(e);
USR = 7.1.3.1.8(f);
USR = 7.1.3.1.8(g);
USR = 7.1.3.1.8(h);

6.2.1.1 Collect Traffic Data for Advisory Messages

Description:

Overview: This process shall collect and fuse traffic data that will be used to create broadcast or advisory messages to travelers. The input data for this process shall consist of historical, current, and predicted traffic and planned event data. The process shall extract from the data those elements appropriate for advisory or broadcast messages and load it into the store of traveler_traffic_information_data store. The data can be provided to the process either via direct request from the process or as a result of periodic (unrequested) updates.

In Subsystem:

Information Service Provider

In Market Package:

AD1

AD2

ATIS1

ATIS4

Data Flows:

Input Flows

foisp_traffic_data
foisp_traffic_information_request
planned_events
prediction_data
traffic_data_for_advisory_output
traveler_traffic_information_data_output

Output Flows

toisp_traffic_data_request
toisp_traffic_information
traffic_data_advisory_request
traveler_traffic_information_data_update

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2.0;
USR = 1.2.2;
USR = 1.2.2.1;
USR = 1.2.2.1.1;
USR = 1.2.2.1.2;
USR = 1.2.2.1.2.1;
USR = 1.2.2.1.3;
USR = 1.5;
USR = 1.5.2;
USR = 1.5.2.5;
USR = 1.5.2.5(d);
USR = 1.5.2.5(g);
USR = 2.0;
USR = 2.2.0;
USR = 2.2.1;
USR = 2.2.1.1;
USR = 2.2.1.1.1;
USR = 2.2.3;
USR = 2.2.3.1;
USR = 2.2.3.1.2;
USR = 2.2.3.1.2(a);
USR = 2.2.3.1.2(b);
USR = 2.2.3.2;
USR = 2.2.3.2.1;

6.2.1.2 Provide Traffic and Transit Advisory Messages

Description:

Overview: This process shall provide advisory data to users in vehicles (drivers or transit users) as a result of a request from the driver or transit user. (e.g. This process supports a request/response type of exchange with the user.) The advisory information is extracted from the data stores of traveler traffic and transit information. The process shall have the capability to filter the advisory data, read from the data stores, store so that the output only contains data that is relevant to the current location of the vehicle from which the request was made. When the user requests location specific data, the vehicle's location shall be provided to the process in the request message. Advisory data requests shall be sent to the data archival process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

ATIS2

Data Flows:

Input Flows

advisory_data_request
traveler_profile_from_vehicle
traveler_traffic_information_data
traveler_transit_information_data

Output Flows

advisory_data
advisory_data_request_for_archive

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2.0;
USR = 1.2.2;
USR = 1.2.2.1;
USR = 1.2.2.1.1;
USR = 1.2.2.1.2;
USR = 1.2.2.1.2.1;
USR = 1.2.2.1.3;
USR = 1.5.0;
USR = 1.5.1;
USR = 1.5.1.1;
USR = 1.5.1.2;
USR = 1.5.1.2.1;
USR = 1.5.1.2.2;
USR = 1.5.1.2.3;
USR = 1.5.1.2.4;
USR = 1.5.1.2.5;
USR = 1.5.1.3;
USR = 1.5.1.4;
USR = 1.5.1.5;
USR = 1.5.2;
USR = 1.5.2.1;
USR = 1.5.2.2;
USR = 1.5.2.3;
USR = 1.5.2.4;
USR = 2.0;
USR = 2.2.0;
USR = 2.2.1;
USR = 2.2.1.1;

USR = 2.2.1.1.1;
USR = 2.2.2;
USR = 2.2.2.1;
USR = 2.2.2.2;
USR = 2.2.2.3;
USR = 2.2.2.3(a);
USR = 2.2.2.3(b);
USR = 2.2.2.3(c);
USR = 2.2.3;
USR = 2.2.3.1;
USR = 2.2.3.1.1;
USR = 2.2.3.1.1(a);
USR = 2.2.3.1.1(b);
USR = 2.2.3.1.1(c);
USR = 2.2.3.1.1(d);
USR = 2.2.3.1.2;
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;

6.2.1.3 Collect Transit Data for Advisory Messages

Description:

Overview: This process shall collect and fuse transit advisory data that will be used to create broadcast or advisory messages to travelers. The process shall extract from the data those elements appropriate for advisory or broadcast messages and load it into the traveler_transit_information_data store. The data can be provided to the process either via direct request from the process or as a result of periodic (unrequested) updates. The process shall fuse all the received data into a coherent set, which is loaded into a traveler_transit_information_data store for access by other processes.

In Subsystem:

Information Service Provider

In Market Package:

ATIS1

ATIS4

Data Flows:

Input Flows

foisp_transit_data
foisp_transit_information_request
transit_incident_data
transit_running_data_for_advisory_output
transit_services_for_advisory_data
traveler_transit_information_data_output

Output Flows

toisp_transit_data_request
toisp_transit_information
transit_conditions_advisories_request
transit_services_advisories_request
traveler_transit_information_data_update

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2.0;
USR = 1.2.2;
USR = 1.2.2.1;
USR = 1.2.2.1.1;
USR = 1.2.2.1.2;
USR = 1.2.2.1.2.1;
USR = 1.2.2.1.3;
USR = 2.0;
USR = 2.2.0;
USR = 2.2.1;
USR = 2.2.1.1;
USR = 2.2.1.1.1;
USR = 2.2.3;
USR = 2.2.3.1;
USR = 2.2.3.1.1;
USR = 2.2.3.1.2;
USR = 2.2.3.2;
USR = 2.2.3.2.1;
USR = 2.2.3.2.2;

6.2.1.4 Provide Traffic and Transit Broadcast Messages

Description:

Overview: This process shall extract advisory data from stores of traveler traffic and transit information at locally determined intervals and send it out to drivers or transit users in vehicles as wide area broadcast messages. The content and rate of these messages shall be based upon parameters from the broadcast_parameters_data store, which is managed by the ISP operator.

In Subsystem:

Information Service Provider

In Market Package:

ATIS1

ATIS4

Data Flows:

Input Flows

broadcast_parameters_data
traveler_traffic_information_data
traveler_transit_information_data

Output Flows

broadcast_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2.0;
USR = 1.2.2;
USR = 1.2.2.1;
USR = 1.2.2.1.1;
USR = 1.2.2.1.2;
USR = 1.2.2.1.2.1;
USR = 1.2.2.1.3;
USR = 1.5.0;
USR = 1.5.1;
USR = 1.5.1.1;
USR = 1.5.1.2;
USR = 1.5.1.2.1;
USR = 1.5.1.2.2;
USR = 1.5.1.2.3;
USR = 1.5.1.2.4;
USR = 1.5.1.2.5;
USR = 1.5.1.3;
USR = 1.5.1.4;
USR = 1.5.1.5;
USR = 1.5.2;
USR = 1.5.2.1;
USR = 1.5.2.2;
USR = 1.5.2.3;
USR = 2.0;
USR = 2.2.0;
USR = 2.2.1;
USR = 2.2.1.1;
USR = 2.2.1.1.1;
USR = 2.2.2;
USR = 2.2.2.1;
USR = 2.2.2.2;
USR = 2.2.2.3;
USR = 2.2.3;

USR = 2.2.3.1;
USR = 2.2.3.1.1;

6.2.1.5 Provide ISP Operator Broadcast Parameters Interface

Description:

Overview: This process shall provide the interface through which the ISP operator can manipulate data in the broadcast_parameters_data store. The data, in this store, shall be used by another process to define the scope and rate of wide area broadcast messages to vehicles. The process shall provide the ISP operator with the ability to request parameter data output and/or update the data store with new parameter values.

In Subsystem:

Information Service Provider

In Market Package:

ATIS1

ATIS4

Data Flows:

Input Flows

broadcast_parameters_data_output
fispo_broadcast_data_parameters_request
fispo_broadcast_data_parameters_update

Output Flows

broadcast_parameters_data_update
tispo_broadcast_data_parameters_output

User Service Requirements Mapped to Pspec:

USR = NA;

6.2.1.6 Provide Transit Advisory Data On Vehicle

Description:

Overview: This process shall gather transit advisory data and provide it via another process to the transit user on-board a transit vehicle. The interface shall receive requests from the transit user specifying the required destination of a transit service ride and other (yellow pages) type services. The transit user may also request and receive information about the state of traffic on the roadway, as well as transit route and stop data (i.e., traffic and transit advisory data). This process extracts data from the store of traveler transit information upon request for advisory data from the driver or transit user in a vehicle. The process shall filter the data read from the store so that output only contains that which is relevant to the current location of the vehicle from which the request was made. The vehicle's location shall be provided to the process in the request data. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS8

Data Flows:

Input Flows

ftu_destination_on_vehicle
ftu_other_services_vehicle_request
other_services_vehicle_response
transit_advisory_data_request
transit_vehicle_location
traveler_information_request
traveler_transit_information_data

Output Flows

other_services_vehicle_request
transit_advisory_data
transit_advisory_vehicle_information
traveler_information
ttu_other_services_vehicle_confirmed

User Service Requirements Mapped to Pspec:

USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.6;
USR = 1.8.1.6(b);
USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.3;
USR = 2.2.1.2.2;
USR = 2.2.1.2.2.1;
USR = 2.2.1.2.2.2;
USR = 2.2.1.2.2.3;
USR = 2.2.1.2.2.4;

6.2.2 Prepare and Output In-vehicle Displays

Description:

Overview: This process shall provide in-vehicle advisory and broadcast data for output to drivers and transit users. The process shall format requests from users for advisory data and output the requests to other processes. The request for advisory data shall allow the user to request only information relevant to the location of the vehicle. The request may be repeated, periodically, or when the vehicle changes location by a distance determined by the implementation. Data broadcast to the driver shall include traffic related data(incidents, link data and in-vehicle signage), as well as data from the vehicle itself. This vehicle data includes vehicle conditions, smart probe data, safety and position warnings, and enhanced vision images. Safety and warning messages shall be prioritized by the process to supersede advisory and broadcast messages. The process shall also support the transfer of reservation requests from the users in vehicles for other services such as yellow pages.

In Subsystem:

Vehicle

In Market Package:

ATIS7

ATIS8

ATIS1

ATIS2

ATIS9

ATMS12

Data Flows:

Input Flows

advisory_data
broadcast_data
driver_advisory_information_request
emergency_message_auto_output
emergency_message_driver_output
position_warnings
safety_warnings
transit_user_advisory_information_request
vehicle_control_status
vehicle_location_for_advisories
vehicle_roadway_warning_data_output
vehicle_signage_data
vehicle_smart_probe_data_output
vehicle_status_details_for_broadcast
vision_data
yellow_pages_advisory_data

Output Flows

advisory_data_request
driver_advisory_information
driver_broadcast_information
transit_user_advisory_information
traveler_profile_from_vehicle
yellow_pages_advisory_requests

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2.0;
USR = 1.2.1;
USR = 1.2.1.1;
USR = 1.2.1.3;
USR = 1.2.1.5;
USR = 1.2.3;
USR = 1.2.3.1;
USR = 1.2.3.1.1;
USR = 1.2.3.1.2;
USR = 1.2.3.1.3;
USR = 1.2.3.1.4;
USR = 1.2.3.1.4.1;
USR = 1.2.3.1.4.2;
USR = 1.2.3.1.5;
USR = 1.2.3.2;
USR = 1.2.3.2.2;
USR = 1.2.3.2.2.1;
USR = 1.2.3.2.4;
USR = 1.2.3.2.5;
USR = 1.3;
USR = 1.3.4;
USR = 1.3.4.2;
USR = 1.3.4.2.2;
USR = 1.3.4.2.2(b);
USR = 2.0;
USR = 2.2.0;
USR = 2.2.1;
USR = 2.2.1.1;
USR = 2.2.1.1.1;
USR = 2.2.1.2.2;
USR = 2.2.1.2.2.1;
USR = 2.2.1.2.2.2;
USR = 2.2.1.2.2.3;
USR = 2.2.1.2.2.4;

6.2.3 Provide Transit User Advisory Interface

Description:

Overview: This process shall provide a data input and output interface for a transit user on-board a transit vehicle. The process shall enable traffic and travel advisory information, plus yellow pages information to be requested and output to the transit user. When constructing the outputs the process shall use the data in the store of vehicle display definitions data. In addition to the traveler's request/ response for information, broadcast advisories about the imminent arrival of the transit vehicle at the next stop are also displayed for the transit user. The process shall handle all inputs and outputs in such a way that they do not impair the vehicle driver's ability to control the transit vehicle in a manner that is safe to both its occupants, to other road and freeway users, and to pedestrians. The input and output forms shall also include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS8

Data Flows:

Input Flows

ftu_request_advisory_information
transit_advisory_data
transit_user_advisory_information
transit_vehicle_advisory_eta
transit_vehicle_eta_for_advisory
traveler_information

Output Flows

transit_advisory_data_request
transit_user_advisory_information_request
traveler_information_request
ttu_advisory_information
ttu_traveler_information

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2.0;
USR = 1.2.1;
USR = 1.2.1.1;
USR = 1.2.1.2;
USR = 1.2.1.3;
USR = 1.2.1.5;
USR = 1.2.3;
USR = 1.2.3.1;
USR = 1.2.3.1.1;
USR = 1.2.3.1.2;
USR = 1.2.3.1.3;
USR = 1.2.3.1.4;
USR = 1.2.3.1.4.1;
USR = 1.2.3.1.4.2;
USR = 1.2.3;
USR = 1.2.3.1;
USR = 1.2.3.1.5;
USR = 1.2.3.2;
USR = 1.2.3.2.2;
USR = 1.2.3.2.2.1;
USR = 1.2.3.2.4;
USR = 1.2.3.2.5;
USR = 1.5.0;
USR = 1.5.2;
USR = 1.5.2.1;
USR = 1.5.2.2;
USR = 1.5.2.3;
USR = 2.2.1.2.2;
USR = 2.2.1.2.2.1;
USR = 2.2.1.2.2.2;
USR = 2.2.1.2.2.3;

6.2.4 Collect Yellow Pages Data

Description:

Overview: This process shall collect and fuse data about (yellow pages) services in order to provide information to users in vehicles. The process shall fuse all the received yellow pages data into a coherent set and loaded into the yellow_pages_information_data store for access by processes in response to requests from users in vehicles.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

ATIS1

ATIS2

ATIS4

ATIS8

Data Flows:

Input Flows

yellow_pages_data
yellow_pages_output

Output Flows

yellow_pages_data_request
yellow_pages_update

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2.0;
USR = 1.2.2;
USR = 1.2.2.1;
USR = 1.2.2.1.1;
USR = 1.2.2.1.2;
USR = 1.2.2.1.2.1;
USR = 1.2.2.1.3;
USR = 1.5.0;
USR = 1.5.2;
USR = 1.5.2.2;
USR = 1.5.2.2(b);
USR = 1.5.2.2(g);
USR = 2.0;
USR = 2.2.0;
USR = 2.2.1;
USR = 2.2.1.1;
USR = 2.2.1.1.1;
USR = 2.2.3;
USR = 2.2.3.1;
USR = 2.2.3.1.1;
USR = 2.2.3.2;
USR = 2.2.3.2.1;
USR = 2.2.3.2.2;
USR = 2.2.3.2.2(a);
USR = 2.2.3.2.2(b);
USR = 2.2.3.2.2(c);

6.2.5 Provide Driver Interface

Description:

Overview: This process shall provide a user interface for a driver through which traffic and travel advisory information can be obtained. The process shall enable traffic and travel advisory information to be requested and output to the driver, and shall also support the automatic output of wide area broadcast information (including in vehicle signage) to the driver. The process shall support output of safety and vision enhancement information to the user. When constructing all outputs the process shall use the vehicle_display_definitions_data store parameters. One purpose of the vehicle_display_definitions_data store is to provide a translation table for road sign and message templates used for in-vehicle display. Part of the input interface provided by the process shall enable the driver to invoke and cancel automatic control of the vehicle including the use of automated highway system (ahs) lanes. The process shall support inputs from the driver in manual or audio form, and shall provide its outputs in audible or visual forms. Visual output may be either in hardcopy, or as a display. Both types of output shall not impair the driver's ability to control the vehicle in a safe manner.

In Subsystem:

Vehicle

In Market Package:

ATIS7

ATIS8

ATIS1

ATIS2

ATIS9

ATMS12

AVSS01

AVSS02

AVSS03

AVSS04

AVSS05

AVSS07

AVSS08

AVSS10

CVO10

EM3

Data Flows:

Input Flows

driver_advisory_information
driver_broadcast_information
fd_activate_vehicle_control
fd_request_advisory_information

Input/Output

vehicle_display_definitions_data

Output Flows

driver_advisory_information_request
td_advisory_information

td_broadcast_information
vehicle_control_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2.0;
USR = 1.2.1.5;
USR = 1.2.3;
USR = 1.2.3.2;
USR = 1.2.3.2.2;
USR = 1.2.3.2.2.1;
USR = 1.2.3.2.5;
USR = 1.3.0;
USR = 1.3.3;
USR = 1.3.3.1;
USR = 1.5;
USR = 1.5.2;
USR = 1.5.2.5;
USR = 1.5.2.5(e);
USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.6;
USR = 1.8.1.6(c);
USR = 6.5;
USR = 6.5.0;
USR = 6.5.3;
USR = 6.5.3.1;
USR = 6.5.3.1.2;

6.2.6 Provide Yellow Pages Data and Reservations

Description:

Overview: This process shall extract data from the yellow_pages_information_data store upon request for data from the driver or a transit user in a vehicle. The data read from the store may be filtered, by the process, so that output only contains that which is relevant to the current location of the vehicle. The process shall also enable the user to make reservations for yellow pages services from a vehicle. Yellow pages advisory requests shall be sent to the data archival process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

Data Flows:

Input Flows

yellow_pages_advisory_requests
yellow_pages_information_data
yellow_pages_reservation_confirmation

Output Flows

yellow_pages_advisory_data
yellow_pages_advisory_requests_for_archive
yellow_pages_reservation_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2.0;
USR = 1.2.2;
USR = 1.2.2.1;
USR = 1.2.2.1.1;
USR = 1.2.2.1.2;
USR = 1.2.2.1.2.1;
USR = 1.2.2.1.3;
USR = 1.5.0;
USR = 1.5.1;
USR = 1.5.1.1;
USR = 1.5.1.2;
USR = 1.5.1.2.1;
USR = 1.5.1.2.2;
USR = 1.5.1.2.3;
USR = 1.5.1.2.4;
USR = 1.5.1.2.5;
USR = 1.5.1.3;
USR = 1.5.1.4;
USR = 1.5.1.5;
USR = 1.5.2;
USR = 1.5.2.1;
USR = 1.5.2.2;
USR = 1.5.2.2(a);
USR = 1.5.2.2(b);
USR = 1.5.2.2(h);
USR = 1.5.2.3;
USR = 1.5.2.3(a);
USR = 2.0;
USR = 2.2.0;
USR = 2.2.1;
USR = 2.2.1.1;

USR = 2.2.1.1.1;
USR = 2.2.2;
USR = 2.2.2.3;
USR = 2.2.3;
USR = 2.2.3.1;
USR = 2.2.3.1.1;
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;

6.3.1 Get Traveler Request

Description:

Overview: This process shall receive input data from a traveler located at a kiosk and send requests to the appropriate processes within the Provide Driver and Traveler Services function for further processing. The process shall provide support for trip planning, traffic, transit, yellow pages services information requests, trip confirmation, yellow pages confirmation, and payment requests. The actual interface to the traveler is provided through a separate process, which creates the input flow to this process.

In Subsystem:

Remote Traveler Support

In Market Package:

ATIS7

ATIS8

ATIS2

ATIS5

Data Flows:

Input Flows

traveler_trip_planning_requests

Output Flows

traveler_current_condition_request
traveler_payment_information
traveler_traffic_condition_request
traveler_transaction_request
traveler_transit_condition_request
traveler_trip_confirmation
traveler_trip_request
traveler_yellow_pages_information_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.1.0;
USR = 1.1.3;
USR = 1.1.3.2;
USR = 1.1.3.2.1;
USR = 1.1.3.2.2;
USR = 1.1.3.2.3;
USR = 1.1.3.2.4;
USR = 1.1.3.2.5;
USR = 1.1.3.2.6;
USR = 1.1.3.2.7;
USR = 1.1.3.2.8;
USR = 1.1.3.2.9;
USR = 1.1.3.2.10;
USR = 1.4.0;
USR = 1.4.1;
USR = 1.4.1.1;
USR = 1.4.1.1(a);
USR = 1.4.1.1(b);
USR = 1.4.1.2;
USR = 1.4.1.2(b);
USR = 1.4.1.2(c);
USR = 1.4.1.2(d);
USR = 1.4.1.2(e);

USR = 1.4.1.3;
USR = 1.5.0;
USR = 1.5.2;
USR = 1.5.2.6;

6.3.2 Inform Traveler

Description:

Overview: This process provides the traveler (located at a kiosk) with data about all requested trip, traffic, transit, yellow pages services information, confirmation of any requested reservations, and payments made as part of confirmed trip plans. The data is sent by the process to an interface process that is responsible for its actual output to the traveler. This data may include digitized map data to act as the background to the output when the data is to be shown in a suitable format. This process shall receive data from other ITS functions by requesting it or be sent data as a result of requests from another process.

In Subsystem:

Remote Traveler Support

In Market Package:

ATIS7

ATIS8

ATIS1

ATIS2

ATIS5

Data Flows:

Input Flows

map_data_for_traveler_displays
traffic_data_for_broadcast_to_kiosks
traffic_data_for_kiosks
transit_deviations_for_broadcast_to_kiosks
transit_deviations_for_kiosks
transit_services_for_kiosks
traveler_payment_confirmation
traveler_traffic_condition_request
traveler_transaction_confirmation
traveler_transit_condition_request
traveler_trip_information
traveler_yellow_pages_data

Output Flows

traffic_data_kiosk_request
transit_deviation_kiosk_request
transit_services_kiosk_request
traveler_trip_planning_responses

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.1.0;
USR = 1.1.1;
USR = 1.1.1.1;
USR = 1.1.1.1.1;
USR = 1.1.1.1.2;
USR = 1.1.1.1.3;
USR = 1.1.1.1.4;
USR = 1.1.1.1.5;
USR = 1.1.1.1.6;
USR = 1.1.2;
USR = 1.1.2.1;
USR = 1.1.2.1.1;
USR = 1.1.2.1.2;
USR = 1.1.2.1.3;
USR = 1.1.2.1.4;

USR = 1.1.2.1.5;
USR = 1.1.2.1.6;
USR = 1.1.2.1.8;
USR = 1.4.0;
USR = 1.4.1;
USR = 1.4.1.3;
USR = 1.5.0;
USR = 1.5.2;
USR = 1.5.2.1;
USR = 1.5.2.2;
USR = 1.5.2.6;
USR = 1.5.2.6(a);
USR = 1.5.2.6(b);
USR = 1.5.2.6(e);
USR = 1.8;
USR = 1.8.2;
USR = 1.8.2.3;
USR = 1.8.2.3(a);
USR = 2.0;
USR = 2.3.0;
USR = 2.3.1;
USR = 2.3.1.3;

6.3.3 Provide Traveler Kiosk Interface

Description:

Overview: This process shall provide an interface at a kiosk through which travelers can input data and can receive data. The functions that the traveler can perform include plan and confirm trips, obtain current traffic and transit information, and declare emergencies. The process shall support the inclusion of yellow pages services such as lodging, restaurants, theaters, and other tourist activities as a part of trip planning and confirmation. The process shall be able to store frequently used data, such as the kiosk location, to reduce the amount of input needed by the traveler for each request. The process shall also carry out input data verification and require input confirmation before passing any of the traveler data to other processes (except when an emergency is being declared). The traveler's payment information shall be obtained by this process from another process specially designed for that purpose. The process shall support traveler inputs in manual or audio form, and shall provide its outputs in audible or visual forms consistent with a kiosk. These forms shall include those that are suitable for travelers with hearing or vision physical disabilities. The process shall enable viewing of data that has been previously output. Where it is appropriate, the process shall use the kiosk's location to filter data being displayed to only show information relevant to the kiosk's location, or to a specific location requested by the user.

In Subsystem:

Remote Traveler Support

In Market Package:

ATIS7

ATIS8

APTS5

ATIS1

ATIS2

ATIS5

EM3

Data Flows:

Input Flows

ft_extra_trip_data
ft_trip_planning_requests
traveler_roadside_credit_identity
traveler_trip_planning_responses

Input/Output

traveler_regular_data

Output Flows

traveler_roadside_trip_costs
traveler_trip_planning_requests
tt_extra_trip_data_request
tt_trip_planning_responses

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.1.0;
USR = 1.1.3;
USR = 1.1.3.2;
USR = 1.1.3.2.1;
USR = 1.1.3.2.2;
USR = 1.1.3.2.3;
USR = 1.1.3.2.4;
USR = 1.1.3.2.5;
USR = 1.1.3.2.6;
USR = 1.1.3.2.7;
USR = 1.1.3.2.8;
USR = 1.1.3.2.9;
USR = 1.1.3.2.10;
USR = 1.4.0;
USR = 1.4.1;
USR = 1.4.1.1;
USR = 1.4.1.1(a);
USR = 1.4.1.1(b);
USR = 1.4.1.2;
USR = 1.4.1.3;
USR = 1.4.1.4;
USR = 1.5;
USR = 1.5.2;
USR = 1.5.2.5;
USR = 1.5.2.5(f);
USR = 1.5.2.6;
USR = 1.5.2.6(b);
USR = 1.5.2.6(c);
USR = 1.5.2.6(d);
USR = 1.5.2.6(e);
USR = 2.0;
USR = 2.3.0;

6.3.4 Update Traveler Display Map Data at Kiosk

Description:

Overview: This process shall provide updates to the digitized map data used as the background for displays of trip, traffic and transit information. This data shall be suitable for use in kiosk displays. The process shall obtain the new data from map data suppliers or some other appropriate data source.

In Subsystem:

Remote Traveler Support

In Market Package:

ATIS7

ATIS8

ATIS2

ATIS5

Data Flows:

Input Flows

fmup_traveler_display_update

Output Flows

map_data_for_traveler_displays
tmup_request_traveler_display_update

User Service Requirements Mapped to Pspec:

USR = 1.1;
USR = 1.1.3;
USR = 1.1.3.1;
USR = 1.1.3.1.1;
USR = 1.1.3.2.8;
USR = 1.1.4;
USR = 1.1.4.1;
USR = 1.1.4.1.3;
USR = 1.5;
USR = 1.5.2;
USR = 1.5.2.5;
USR = 1.5.2.5(f);

6.4.1 Screen Rider Requests

Description:

Overview: This process shall accept and screen traveler requests for ride-sharing. These requests shall be sent to the process as a result of trip plan requests received from travelers by other processes. This process shall use eligibility data from a local rideshare_data store, to screen travelers before they are matched with other travelers and to enable ridesharing for all or part of their proposed trips. Traveler rideshare requests and rideshare data from the rideshare_data data store shall be sent to the data archival process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS8

Data Flows:

Input Flows

rideshare_eligibility_data
traveler_rideshare_request

Output Flows

rideshare_data_for_archive
rideshare_ineligible_status_notification
rideshare_request_from_eligible_traveler
traveler_rideshare_request_for_archive

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.4.0;
USR = 1.4.1;
USR = 1.4.1.4;
USR = 1.4.2;
USR = 1.4.2.1;
USR = 1.4.2.2;
USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.2;
USR = 1.8.1.2(d);
USR = 1.8.1.2(g);
USR = 1.8.1.3;
USR = 1.8.1.3(d);
USR = 1.8.1.3(g);
USR = 1.8.2;
USR = 1.8.2.1;
USR = 1.8.2.1(d);
USR = 1.8.2.4;
USR = 1.8.2.4(d);
USR = 1.8.2.4(g);
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.4;
USR = 7.1.3.1.4(c);

6.4.2 Match Rider and Provider

Description:

Overview: This process shall match travelers for ridesharing trips. The process shall attempt to achieve a match by considering some or all of the following: the origin and destination of the traveler's proposed trip, any routing constraints, preferences specified by the traveler, compatibility of this rideshare with rideshares confirmed by other travelers, the requesting traveler's eligibility data, and traffic data obtained on request from the Manage Traffic function. The process shall consider the possible disbenefits to other travelers who will be part of the same rideshare when finding the rideshare best suited to the traveler's requirements. The process shall store data about selected rideshares in the rideshare_data store, and shall update the data when confirmation of the rideshare acceptance is received from another process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS8

Data Flows:

Input Flows

rideshare_confirmation_data
rideshare_request_from_eligible_traveler
traffic_data_for_ridesharing

Input/Output

rideshare_data

Output Flows

rideshare_selection
traffic_data_ridesharing_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.4.0;
USR = 1.4.1;
USR = 1.4.1.2;
USR = 1.4.1.3;
USR = 1.4.1.4;
USR = 1.4.3;
USR = 1.4.3.4;
USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.2;
USR = 1.8.1.2(d);
USR = 1.8.1.2(g);
USR = 1.8.1.3;
USR = 1.8.1.3(d);
USR = 1.8.1.3(g);
USR = 1.8.2;
USR = 1.8.2.1;
USR = 1.8.2.1(d);
USR = 1.8.2.4;
USR = 1.8.2.4(d);
USR = 1.8.2.4(g);

6.4.3 Report Ride Match Results to Requestor

Description:

Overview: This process shall report ridesharing match results to requesters. The data for the results shall be provided to this process by other processes responsible for assessing traveler eligibility, and the actual match with travelers in other rideshares. The process shall output data indicating a failure when either the data from the eligibility process shows a failure, or no ridesharing match can be found. The process shall also determine that no ridesharing match can be found if no match is found between the traveler's rideshare request and the rideshare data provided as input to it by another process. When a successful match is found, the process shall output the rideshare data to the process from which the traveler's request was received.

In Subsystem:

Information Service Provider

In Market Package:

ATIS8

Data Flows:

Input Flows

rideshare_ineligible_status_notification
rideshare_selection
traveler_rideshare_request

Output Flows

rideshare_response

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.4.1;
USR = 1.4.1.3;
USR = 1.4.1.4;

6.4.4 Confirm Traveler Rideshare Request

Description:

Overview: This process shall confirm the traveler's rideshare match and initiate a payment transaction where appropriate. The process shall send the payment transaction data for action by a process in the Provide Electronic Payment Services function. The results of this transaction shall be sent by this process to the process providing the overall trip confirmation. Once a rideshare match is confirmed, this data is sent to the rideshare match process where it can be factored in to subsequent matches.

In Subsystem:

Information Service Provider

In Market Package:

ATIS8

Data Flows:

Input Flows

rideshare_payment_confirmation
traveler_rideshare_confirmation

Output Flows

rideshare_confirmation
rideshare_confirmation_data
rideshare_payment_request

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.3.0;
USR = 2.3.1;
USR = 2.3.1.3;
USR = 2.3.1.4;

6.5.1 Collect and Update Traveler Information

Description:

Overview: This process shall collect and update data about incidents, road construction, weather, events and yellow pages data. This data shall be obtained by the process from other ITS functions and from outside sources such as the weather service, yellow pages service providers and the media. The process shall load the data into a local store for use by the process that provides yellow pages information and reservations.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

ATIS1

ATIS2

ATIS4

Data Flows:

Input Flows

fm_traveler_information
fws_current_weather
fws_predicted_weather
fypsp_yellow_pages_data
incident_information
yellow_pages_new_data_request
yellow_pages_service_provider_data
yellow_pages_update_request

Input/Output

tourist_information

Output Flows

current_conditions
incident_information_request
tm_traveler_information_request
typsp_yellow_pages_info_request
yellow_pages_update_response

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.1.0;
USR = 1.1.2;
USR = 1.1.2.1;
USR = 1.1.2.1.1;
USR = 1.1.2.1.2;
USR = 1.1.2.1.3;
USR = 1.1.2.1.4;
USR = 1.1.2.1.5;
USR = 1.1.2.1.6;
USR = 1.1.2.1.8;
USR = 1.5.0;
USR = 1.5.1;
USR = 1.5.1.1;
USR = 1.5.1.2;
USR = 1.5.1.3;
USR = 1.7;
USR = 1.7.1;
USR = 1.7.1.1;
USR = 1.7.1.1.1;

6.5.2 Provide Traveler Yellow Pages Information and Reservations

Description:

Overview: This process shall provide information and reservation services obtained from yellow pages service providers. The process shall provide the information and reservation data so that it can easily form part of a traveler's information request or trip planning activities. The process shall be able to request additional yellow pages information if the process cannot find the required data in the tourist_information data store. The process shall send requests for payment to a process in the Provide Electronic Payment Services function for action, and shall send the response back to the process from which the payment request was received. The traveler's yellow pages requests shall be sent to the data archival process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

Data Flows:

Input Flows

fypsp_transaction_confirmation
traveler_other_services_payment_result
traveler_payment_information
traveler_personal_payment_information
traveler_personal_transaction_request
traveler_personal_yellow_pages_information_request
traveler_transaction_request
traveler_yellow_pages_information_request
yellow_pages_data_request
yellow_pages_reservation_request
yellow_pages_update_response

Output Flows

traveler_other_services_payment_request
traveler_personal_yellow_pages_data
traveler_yellow_pages_data
traveler_yellow_pages_requests_for_archive
typsp_transaction_request
yellow_pages_data
yellow_pages_reservation_confirmation
yellow_pages_update_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.5.0;
USR = 1.5.1;
USR = 1.5.1.3;
USR = 1.5.1.5;
USR = 1.5.2;
USR = 1.5.2.2;
USR = 1.5.2.2(a);
USR = 1.5.2.2(b);
USR = 1.5.2.2(h);
USR = 1.5.2.3;
USR = 1.5.2.3(a);
USR = 1.5.2.3(b);
USR = 1.5.2.4;
USR = 7.0;
USR = 7.1;
USR = 7.1.0;

USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;

6.5.3 Register Yellow Pages Service Providers

Description:

Overview: This process shall register yellow pages service providers. The process shall accept requests for registration from the providers and shall pass the data to a process in the Provide Electronic Payment Services function for action. The process shall send the result of this action to the provider, and if successful, shall send a request for the process that manages the contents of the store of tourist information to request data from the provider. The details of the provider shall also be loaded into the store used by that process, so that data from the provider can readily be obtained in the future. This process shall perform updating of the yellow pages service provider details.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

Data Flows:

Input Flows

fypsp_provider_profile_update
fypsp_request_provider_registration
yellow_pages_service_provider_registration_response

Output Flows

typsp_provider_update_confirm
yellow_pages_new_data_request
yellow_pages_service_provider_data
yellow_pages_service_provider_registration_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.7.0;
USR = 1.7.4;

6.6.1 Provide Multimodal Route Selection

Description:

Overview: This process shall manage the creation of multimodal routes (those with one or more modes in them) in response to traveler's trip or route requests. It shall support on-line route guidance for travelers using personal devices, route guidance for vehicles, selection of specialized vehicle based routes for other ITS functions, (such as Manage Emergency Services and Manage Commercial Vehicles), and selection of multimodal routes in response to trip planning requests from travelers. The multimodal routes provided by the process shall take account of the traveler's preferences and constraints. Constraints can include the access needs of those with disabilities. Preferences can include minimizing waiting time at modal interchange points, level of traveler security, or minimum cost. Trip requests, traveler route requests, and traveler route acceptances shall be sent to the data archival process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS6

ATIS8

ATIS2

ATIS5

Data Flows:

Input Flows

cf_route_request
cv_route_request
fws_current_weather
fws_predicted_weather
other_route
paratransit_route_response
planned_events
transit_route
traveler_route_accepted
traveler_route_request
trip_request
vehicle_route

Output Flows

cf_route
cv_route
get_other_route
get_transit_route
get_vehicle_route
paratransit_route_confirm
paratransit_route_request
supplied_route
traveler_guidance_route
traveler_route_accepted_for_archive
traveler_route_request_for_archive
trip_request_for_archive

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.2;
USR = 1.3.2;
USR = 1.3.2.1;
USR = 1.3.2.2;
USR = 1.3.2.2.2;
USR = 1.3.3;
USR = 1.3.3.1;
USR = 1.3.3.1(b);
USR = 1.3.4;
USR = 1.3.4.2;
USR = 1.3.4.2.1;
USR = 1.3.4.3;
USR = 1.4.0;
USR = 1.4.3;
USR = 1.4.3.3;
USR = 1.4.4.3(f);
USR = 1.4.4.3(g);
USR = 5.0;
USR = 5.2.0;
USR = 5.2.2;
USR = 5.2.2.1;
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;
USR = 7.1.3.1.8(f);

6.6.2.1 Calculate Vehicle Route

Description:

Overview: This process shall calculate trip planning and real-time dynamic guidance routes for all types of vehicles. The route data provided by the process in response to requests from vehicles using infrastructure based in-vehicle guidance shall only contain data necessary for the vehicle to provide guidance (since the data is intended for use by an in-vehicle navigation unit). The route provided for trip planning purposes shall contain data in a form which can be presented to a user via display (or alternatively in audio form). The process shall select the route according to the data included in the route request. Data provided by the requesting process includes preferences and constraints. The process shall have the capability of using current and/or predicted conditions of the road network in route calculation. The process shall have the capability of including additional factors such as current or predicted weather in the calculation of route. If the process cannot find the data it needs in the route_segment_details_data store, it shall request the process responsible for providing route calculation data to obtain it from the appropriate source. The process shall have the capability of outputting routes for special priority vehicles to the Manage Traffic function so that signal preemption could be provided for the special priority vehicle. The process shall send details of routes for commercial vehicles with hazardous or unusual loads to the Manage Incidents function for monitoring (as a potential, or a planned event). Route guidance data and vehicle guidance route requests and acceptances shall be sent to the data archival process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS5

ATIS6

ATIS8

ATIS8

Data Flows:

Input Flows

fws_current_weather
fws_predicted_weather
get_vehicle_route
map_data_for_route_selection
route_segment_details
route_segment_details_updated
route_selection_parameters
vehicle_guidance_route_accepted
vehicle_route_request

Input/Output

routes_for_vehicles_data

Output Flows

logged_special_vehicle_route
request_route_segment_data
route_guidance_data_for_archive
special_vehicle_priority_routing
vehicle_guidance_route
vehicle_guidance_route_accepted_for_archive
vehicle_route
vehicle_route_request_for_archive

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2.0;
USR = 1.2.1;
USR = 1.2.1.4;
USR = 1.2.1.4.1;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.2.1(a);
USR = 1.3.1.2.1(b);
USR = 1.3.1.3;
USR = 1.3.1.3(a);
USR = 1.3.1.3(b);
USR = 1.3.2;
USR = 1.3.2.2;
USR = 1.3.2.2.1;
USR = 1.3.3;
USR = 1.3.3.1;
USR = 1.3.3.2;
USR = 1.3.3.2(a);
USR = 1.3.3.2(b);
USR = 1.3.3.2.1;
USR = 1.3.3.2.2;
USR = 1.3.3.3;
USR = 1.3.4;
USR = 1.3.4.2;
USR = 1.3.4.3;
USR = 1.3.4.3.1;
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;
USR = 7.1.3.1.8(d);

6.6.2.2 Provide Vehicle Route Calculation Data

Description:

Overview: This process shall update the data stores containing information which is used by the another process to calculate vehicle routes. This process shall also provide data about links (speed or travel times), and queues to be broadcast to vehicles (to support autonomous guidance with dynamic link updates). The process shall fuse link and queue data received from Manage Traffic sources with probe data received from vehicles under infrastructure based route guidance, or with probe data obtained from other sources (such as from an electronic toll collection system). The process shall obtain route segment data as requested data or as data received at periodic intervals from other ITS functions. The process shall have the capability to request data about route segments outside its own area by sending a data request to the process that provides the interface with other ISP's. Link addresses, mapped to other ISPs, will be maintained by this process and stored in the link_data_store. Usage of current road networks shall be sent to the data archival process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS6

ATIS6

ATIS8

ATIS5

ATMS02

Data Flows:

Input Flows

current_highway_network_state
current_road_network_state
link_data_for_guidance
other_route_segment_data
planned_events
prediction_data
request_route_segment_data
routes_for_vehicles_data
traffic_data_for_guidance
vehicle_probe_data_amalgamation

Input/Output

route_segment_details

Output Flows

current_road_network_use
current_road_network_use_for_archive
link_and_queue_data
link_data_store
request_other_route_segment_data
route_segment_details_updated
traffic_data_guidance_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.3;
USR = 1.3.3;
USR = 1.3.3.1;
USR = 1.3.3.2;
USR = 1.3.3.2.1;
USR = 1.3.3.2.2;
USR = 1.3.3.3;
USR = 1.3.4;
USR = 1.3.4.2;
USR = 1.3.4.3;
USR = 1.3.4.3.1;
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;

6.6.2.3 Provide Route Segment Data for Other Areas

Description:

Overview: This process shall obtain from another ISP current or predicted data for road links that are outside the area served by the local supplier. This area, which may be defined on a geographic or jurisdictional basis, is the portion of the transportation network on which the ISP maintains real time information. Identification of which ISP to contact is based upon a store that maps a link to the ISP which maintains real time information about this link. If there is no map to another ISP in the data store, then the process will return default or static data for the link(s). This process shall also respond to similar requests from other ISPs for real time data on links within the local database.

In Subsystem:

Information Service Provider

In Market Package:

ATIS6

ATIS7

ATIS8

ATIS2

ATIS5

ATIS8

Data Flows:

Input Flows

foisp_data_supply
foisp_request_data
link_data_store
request_other_route_segment_data
route_segment_details

Output Flows

other_route_segment_data
toisp_data_supply
toisp_request_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.2.1(c);
USR = 1.7.0;
USR = 1.7.4;

6.6.2.4 Update Vehicle Route Selection Map Data

Description:

Overview: This process shall provide the interface to map update providers, or other appropriate data sources, through which updates of the digitized map data can be obtained. The process shall request new data from the provider on request from the ISP operator interface process. The data received from the supplier shall be loaded into a the map_data_for_route_selection data store by the process in such a way that it can be easily used by the route selection process in determining vehicle routes, trip plannning, and on-line vehicle guidance.

In Subsystem:

Information Service Provider

In Market Package:

ATIS6

ATIS7

ATIS8

ATIS8

ATIS2

ATIS5

Data Flows:

Input Flows

fmup_route_selection_map_data
request_route_selection_map_update

Output Flows

map_data_for_route_selection
tmup_request_route_selection_map_update

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.2;
USR = 1.3.2.1;
USR = 1.3.2.1(a);

6.6.2.5 Provide ISP Operator Route Parameters Interface

Description:

Overview: This process shall provide the interface through which the ISP operator can input and update route calculation parameters used by the Provide Driver and Traveler Services function. The process shall provide an interface through which the ISP operator can review and request update of map data. The operator shall be able to use the process to request digitized map updates from suppliers, request output of trip planning and route selection control parameters, or to update the control parameters in the route_selection_parameters data store. The process shall support inputs from the ISP operator in manual or audio form, and shall provide its outputs in audible or visual forms. It shall enable the visual output to be in hardcopy, and/or display.

In Subsystem:

Information Service Provider

In Market Package:

ATIS5

ATIS6

ATIS8

Data Flows:

Input Flows

fispo_request_other_routes_selection_map_data_update
fispo_request_route_selection_map_data_update
fispo_route_selection_parameters_request
fispo_route_selection_parameters_update

Input/Output

route_selection_parameters

Output Flows

request_other_routes_map_update
request_route_selection_map_update
tispo_route_selection_parameters

User Service Requirements Mapped to Pspec:

USR = 1.3;
USR = 1.3.3;
USR = 1.3.3.1;
USR = 1.3.3.1(a);
USR = 1.3.4;
USR = 1.3.4.1;
USR = 1.3.4.1(a);

6.6.2.6 Calculate Vehicle Probe Data for Guidance

Description:

Overview: This process shall calculate route segment travel times from vehicle probe data. The probe data shall be accepted by the process from a variety of sources including toll collection points and vehicles receiving on-line infrastructure based guidance. The process shall be responsible for combining the data obtained from these sources and producing one set of route segment travel times or route segment speeds. The process shall indicate route segments for which no data, or insufficient data, is available (this indication could be by setting the link time or speed to zero). Vehicle guidance probe data shall be sent to the data archival process.

In Subsystem:

Information Service Provider

In Market Package:

ATMS02

Data Flows:

Input Flows

vehicle_guidance_probe_data
vehicle_toll_probe_data

Output Flows

vehicle_guidance_probe_data_for_archive
vehicle_probe_data_amalgamation

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.3;
USR = 1.3.2;
USR = 1.3.2.1;
USR = 1.3.2.1(b);
USR = 1.3.3;
USR = 1.3.3.1;
USR = 1.3.3.2;
USR = 1.3.3.2.1;
USR = 1.3.3.2.2;
USR = 1.3.3.3;
USR = 1.3.4;
USR = 1.3.4.2;
USR = 1.3.4.3;
USR = 1.3.4.3.1;
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;
USR = 7.1.3.1.8(a);
USR = 7.1.3.1.8(c);

6.6.3 Update Other Routes Selection Map Data

Description:

Overview: This process shall provide the interface to a map update providers through which to obtain fresh updates of digitized map data used in identification of non-vehicle portions of routes. The process shall request new data from the provider on request from the ISP operator interface process. The data received from the supplier shall be loaded into the map_data_for_general_use data store by the process in such a way that it can be easily used by the route selection process in determining non-vehicle routes for use in on-line traveler guidance and trip planning.

In Subsystem:

Information Service Provider

In Market Package:

ATIS5

ATIS6

ATIS8

ATIS8

Data Flows:

Input Flows

fmup_other_routes_map_data
request_other_routes_map_update

Output Flows

map_data_for_general_use
tmup_request_other_routes_map_update

User Service Requirements Mapped to Pspec:

USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.2.1(b);

6.6.4 Select Transit Route

Description:

Overview: This process shall determine routes that are based on regular transit services. Routes shall be provided by the process to travelers in response to trip planning and on-line personal guidance requests. The data provided by the process shall be different for the two types of requests since trip planning information will not need the detail that guidance data requires. The process shall base routes on the current state of the regular transit services using data obtained from processes in the Manage Transit function. It shall also respond to any preferences and constraints, such as those for travelers with special needs, that are specified in the route request. Data on the current use of transit routes in on-line guidance shall be provided by the process to the Manage Demand function to aid in demand management. This data on current use of the transit routes in on-line guidance is stored in the transit_mode_routes data store.

In Subsystem:

Information Service Provider

In Market Package:

ATIS6

ATIS8

ATIS2

ATIS5

Data Flows:

Input Flows

get_transit_route
transit_running_data_for_guidance
transit_services_for_guidance

Input/Output

transit_mode_routes
transit_route_details

Output Flows

current_transit_routes_use
transit_conditions_guidance_request
transit_route
transit_services_guidance_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.2.1(b);
USR = 1.3.1.2.1(c);
USR = 1.3.1.3;
USR = 1.3.3;
USR = 1.3.3.2;
USR = 1.3.3.2.1;
USR = 1.3.3.2.2;
USR = 1.3.4;
USR = 1.3.4.3;
USR = 1.4.0;
USR = 1.4.3;
USR = 1.4.3.3;

6.6.5 Select Other Routes

Description:

Overview: This process shall determine routes, or portions of routes, not based on use of vehicles or regular transit services. Routes shall be provided by the process for travelers in response to trip planning, on-line personal guidance requests, and for data archival. Data provided by the process will be different for the two types of requests since the data for trip planning will not need the level of detail that guidance data requires. The process shall calculate its routes using digitized map data obtained and updated by another process. It shall make use of the alternative modes, (such as ferries, walking, cycling, etc.) that have been specified in the route request, and shall also take account of any preferences and constraints, (such as those for travelers with special needs). Data on current use of routes in on-line guidance shall be provided by the process to the Manage Demand function.

In Subsystem:

Information Service Provider

In Market Package:

ATIS6

ATIS8

ATIS2

ATIS5

Data Flows:

Input Flows

get_other_route
map_data_for_general_use
planned_events

Input/Output

other_modes_routes

Output Flows

current_other_routes_use
current_other_routes_use_for_archive
other_route

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.2.1(d).2;
USR = 1.3.1.2.1(d).3;
USR = 1.3.1.3;
USR = 1.3.1.3(c);
USR = 1.3.1.3(d);
USR = 1.3.3;
USR = 1.3.3.2;
USR = 1.3.3.2.1;
USR = 1.3.3.2.2;
USR = 1.3.4;
USR = 1.3.4.3;
USR = 1.3.4.3.1;
USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.8;

6.7.1.1 Build Driver Personal Security Message

Description:

Overview: This process shall respond to the input of a request from a driver for action by the emergency services. Input of the request shall be received by the process from the driver via a panic button or some other functionally similar form of input device provided as part of the in-vehicle equipment. When the input is received, the process shall send a message to the communications process, containing the vehicle's current location, its identity and basic vehicle data relevant to its current condition, as well as any other data, such as personal medical history, vehicle orientation, etc., that may be developed in-vehicle by other systems.

In Subsystem:

Vehicle

In Market Package:

CVO10

EM3

Data Flows:

Input Flows

fd_emergency_request
vehicle_identity_for_driver_security_store
vehicle_location_for_emergencies
vehicle_status_details_for_driver_security

Output Flows

driver_personal_emergency_request

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.1.0;
USR = 5.1.1;
USR = 5.1.1.1;
USR = 5.1.1.1(a);
USR = 5.1.1.1(b);
USR = 5.1.1.1(c);
USR = 5.1.1.1(d);
USR = 5.1.1.1(e);
USR = 5.1.1.2;

6.7.1.2 Provide Driver In-vehicle Communications Function

Description:

Overview: This process shall prepare and send an emergency message from a driver to the Manage Emergency Services function. The message shall only be sent by the process in response to data received from another process that monitors driver inputs. Once an emergency message has been sent, the process shall send a message to that effect to another process in the Provide Vehicle Monitoring and Control function for output to the driver. The process shall then await a response from the Manage Emergency Services function, and then send a detailed message to the other process for output to the driver. Output of the emergency message to the Manage Emergency Services function shall be repeated by the process at regular intervals until a response is received.

In Subsystem:

Vehicle

In Market Package:

CVO10

EM3

Data Flows:

Input Flows

driver_personal_emergency_request
emergency_request_driver_acknowledge

Output Flows

emergency_message_driver_output
emergency_request_driver_details

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.1.0;
USR = 5.1.1;
USR = 5.1.1.3;

6.7.2.1.1 Determine In-vehicle Guidance Method

Description:

Overview: This process shall act as the interface for guidance requests received from drivers in vehicles. The process shall select the best method for in-vehicle guidance based on data in the driver's request. Three general methods of route guidance are supported: 1) dynamic (infrastructure based guidance is provided to the vehicle unit), 2) dynamic autonomous (link and queue speed or travel times are obtained from the infrastructure and used by the autonomous in vehicle unit), and autonomous (the in vehicle unit uses only locally available data- there is no information provided by the infrastructure). When dynamic guidance is selected, the vehicle's travel time for each link shall be provided by the process back to a central source of data. If the communications link to the central source fails in either of the modes that use it, the process shall automatically revert to the use of local data only. When the original mode was centralized guidance, the process shall use the last set of guidance data that was received, and if this is not sufficient for the vehicle to reach the requested destination, automatically revert to autonomous guidance using local data only.

In Subsystem:

Vehicle

In Market Package:

ATIS6

ATIS5

Data Flows:

Input Flows

autonomous_vehicle_guidance_data
driver_guidance_accepted
driver_guidance_data
driver_guidance_request
dynamic_vehicle_guidance_data

Input/Output

retained_vehicle_guidance_data

Output Flows

autonomous_vehicle_guidance_accepted
autonomous_vehicle_guidance_data_request
driver_input_request
driving_guidance_instructions
dynamic_vehicle_guidance_data_request
vehicle_guidance_route_accepted

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2.0;
USR = 1.2.1;
USR = 1.2.1.4;
USR = 1.2.1.4.2;
USR = 1.2.2;
USR = 1.2.2.1;
USR = 1.2.2.1.4;
USR = 1.2.2.2;
USR = 1.2.3;
USR = 1.2.3.2;
USR = 1.2.3.2.1;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.3;

USR = 1.3.2;
USR = 1.3.2.2;
USR = 1.3.2.3;
USR = 1.3.2.3.1;
USR = 1.3.3.1;
USR = 1.3.3.2;
USR = 1.3.3.2.2;
USR = 1.3.3.3;
USR = 1.3.4;
USR = 1.3.4.2;
USR = 1.3.4.2.1;
USR = 1.3.4.2.2;

6.7.2.1.2 Provide Dynamic In-vehicle Guidance

Description:

Overview: This process shall enable dynamic vehicle route guidance data to be calculated. The process shall perform the same dynamic vehicle route guidance services for vehicles that are under automatic control using automatic highway system (ahs)lanes. When providing dynamic guidance, the process provides vehicle travel times as probe data to another process in the Provide Driver and Traveler Services function. The process shall base its guidance request on data input by the driver through another process, and on the vehicles current location as provided by another process.

In Subsystem:

Vehicle

In Market Package:

ATIS6

ATIS5

ATMS02

Data Flows:

Input Flows

ahs_route_request
dynamic_vehicle_guidance_data_request
vehicle_guidance_route
vehicle_location_for_dynamic_guidance

Output Flows

ahs_route
dynamic_vehicle_guidance_data
vehicle_guidance_probe_data
vehicle_route_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.3;
USR = 1.3.2;
USR = 1.3.2.2;
USR = 1.3.2.3;
USR = 1.3.2.3.1;
USR = 1.3.3.1;
USR = 1.3.3.2;
USR = 1.3.3.2.2;
USR = 1.3.3.3;
USR = 1.3.4;
USR = 1.3.4.2;
USR = 1.3.4.2.1;
USR = 1.3.4.2.2;
USR = 1.3.4.3;

6.7.2.1.3 Provide Autonomous In-vehicle Guidance

Description:

Overview: This process shall provide autonomous in-vehicle guidance. It shall calculate the route using data obtained from an in-vehicle navigable map database which can be supplemented with link queue and travel time data obtained from a central source, if specified by the driver and available. The process shall provide guidance in the form of actual driving instructions, e.g. turn left at the next intersection, take the right lane, etc. When link queue and travel time data are being used, the process shall provide guidance for the best route for current traffic conditions, within the preferences and constraints specified by the driver in the guidance request.

In Subsystem:

Vehicle

In Market Package:

ATIS3

ATIS4

Data Flows:

Input Flows

autonomous_vehicle_guidance_accepted
autonomous_vehicle_guidance_data_request
link_and_queue_data
vehicle_location_for_autonomous_guidance
vehicle_map_database

Output Flows

autonomous_vehicle_guidance_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.3;
USR = 1.3.2;
USR = 1.3.2.2;
USR = 1.3.2.3;
USR = 1.3.2.3.1;
USR = 1.3.3.1;
USR = 1.3.3.2;
USR = 1.3.3.2.2;
USR = 1.3.3.3;
USR = 1.3.4;
USR = 1.3.4.2;
USR = 1.3.4.2.1;
USR = 1.3.4.2.2;
USR = 1.3.4.2.2(a);
USR = 1.3.4.3;

6.7.2.2 Process Vehicle Location Data

Description:

Overview: This process shall provide the vehicle's current location. It shall calculate the location from one or more sources of position data such as GPS, DGPS, odometer and differential odometers, and shall refine its calculations using techniques such as map matching, etc. Location data (intended for use by in-vehicle navigation, guidance systems, and any emergency notification systems) should be provided by the process in a manner that is as precise as is practical within cost and technology constraints. Location data intended for transit vehicles and driver advisories may be less precise.

In Subsystem:

Vehicle

In Market Package:

ATIS6

APTS1

ATIS3

ATIS4

ATIS5

ATMS02

CVO01

EM2

EM3

Data Flows:

Input Flows

From_Location_Data_Source
vehicle_map_database

Output Flows

vehicle_location_for_advisories
vehicle_location_for_autonomous_guidance
vehicle_location_for_cv
vehicle_location_for_dynamic_guidance
vehicle_location_for_emergencies
vehicle_location_for_emergency_services
vehicle_location_for_incidents
vehicle_location_for_maintenance
vehicle_location_for_transit

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.2.0;
USR = 1.2.1;
USR = 1.2.1.1;
USR = 1.2.1.2;
USR = 1.2.1.3;
USR = 1.2.1.5;
USR = 1.2.3;
USR = 1.2.3.1;
USR = 1.2.3.1.1;
USR = 1.2.3.1.2;
USR = 1.2.3.1.3;
USR = 1.2.3.1.4;
USR = 1.2.3.1.4.1;
USR = 1.2.3.1.4.2;
USR = 1.2.3.1.5;
USR = 1.2.3.2;
USR = 1.2.3.2.2.1;
USR = 1.2.3.2.4;
USR = 1.2.3.2.5;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.2;
USR = 1.3.2.1;
USR = 1.3.2.2;
USR = 1.3.3;
USR = 1.3.3.1;
USR = 1.3.4;
USR = 1.3.4.2;
USR = 1.3.4.2.1;
USR = 1.3.4.2.2;
USR = 1.3.4.3;
USR = 2.2.1.2.2;
USR = 2.2.1.2.2.1;
USR = 2.2.1.2.2.2;
USR = 2.2.1.2.2.3;
USR = 2.2.1.2.2.4;
USR = 5.0;
USR = 5.1.0;
USR = 5.1.1;
USR = 5.1.1.1;
USR = 5.1.1.1(d);
USR = 5.1.1.2;
USR = 5.1.1.4;
USR = 5.1.2;
USR = 5.1.2.1;
USR = 5.1.2.1.1;
USR = 5.1.2.1.2;
USR = 5.1.2.2;
USR = 5.1.2.2(b);
USR = 6.0;
USR = 6.5.0;
USR = 6.5.1;
USR = 6.5.1.1;
USR = 6.5.1.1.1;
USR = 6.5.1.1.2;
USR = 6.5.1.1.3;
USR = 6.5.2;
USR = 6.5.2.1;

USR = 6.5.2.1.1;
USR = 6.5.2.1.2;
USR = 6.5.3;

6.7.2.3 Provide Driver Guidance Interface

Description:

Overview: This process shall provide a user interface for the vehicle's driver through which route guidance is provided. Three types of route guidance provided by other processes shall be supported by this process(dynamic infrastructure based, autonomous with infrastructure data update, and autonomous). The process shall enable input by the driver of the type of guidance required, the data from which the route is to be determined and output of the resulting route. The process shall not provide on-line guidance until the route has been accepted by the driver. For those forms of guidance that require an on-board map database, the process shall provide an interface through which the driver may obtain and pay for an initial copy of the database plus updates when needed. The process shall support inputs from the driver in either manual or audio form, and shall provide its outputs in audible or visual forms. It shall enable the visual output to be either in hardcopy, and/or display. Both types of output shall not impair the driver's ability to control the vehicle in a safe manner.

In Subsystem:

Vehicle

In Market Package:

ATIS6

ATIS3

ATIS4

ATIS5

Data Flows:

Input Flows

driver_credit_identity
driver_input_request
driver_map_update_response
driving_guidance_instructions
fd_guidance_data
fd_guidance_map_update_request
fd_guidance_request
fd_guidance_route_accepted

Output Flows

driver_advanced_payment_for_map
driver_guidance_accepted
driver_guidance_data
driver_guidance_request
driver_map_update_request
td_driving_guidance
td_guidance_input_request
td_guidance_map_update_response
td_guidance_route_details

ser Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.2.1(d);
USR = 1.3.1.2.1(d).2;
USR = 1.3.1.2.1(d).3;
USR = 1.3.2;
USR = 1.3.2.1;
USR = 1.3.2.2;
USR = 1.3.3;
USR = 1.3.3.1;
USR = 1.3.4;
USR = 1.3.4.1;
USR = 1.3.4.1(d);
USR = 1.3.4.1(e);
USR = 1.3.4.2;
USR = 1.3.4.2.1;
USR = 1.3.4.2.2;

6.7.2.4 Update Vehicle Navigable Map Database

Description:

Overview: This process shall update the vehicle's navigable database based on digitized data obtained from a map provider, or other appropriate data source. The update shall be initiated by the driver through another process .The process shall have the capability to allow a financial transaction (to pay for the update) to be successfully completed using processes in the Provide Electronic Payment Services function. When the new map data is received, it shall be loaded by the process into the vehicle_map_database data store for use by other processes. The result of the update request (successful or not) shall be sent back to the driver interface process for output to the driver.

In Subsystem:

Vehicle

In Market Package:

ATIS6

ATIS3

ATIS4

ATIS5

Data Flows:

Input Flows

driver_map_update_payment_response
driver_map_update_request
fmup_vehicle_map_update
fmup_vehicle_map_update_cost

Output Flows

driver_map_update_payment_request
driver_map_update_response
tmup_vehicle_map_update_cost_request
tmup_vehicle_map_update_request
vehicle_map_database

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.2;
USR = 1.3.2.1;

6.8.1.1.1 Determine Personal Portable Device Guidance Method

Description:

Overview: This process shall act as the interface for personal guidance requests received from travelers with personal portable devices. The process shall select the best method for personal guidance based on data in the traveler's request Two methods shall be available to the process, comprising dynamic infrastructure based guidance is provided to the personal portable device), and autonomous (the personal portable device uses only locally available data- there is no information provided by the infrastructure) If the communications link to the central source fails, the process shall use the last set of guidance data that was received, and if this is not sufficient for the traveler to reach the requested destination, automatically revert to the use of autonomous guidance using local data only.

In Subsystem:

Personal Information Access

In Market Package:

ATIS6

ATIS5

Data Flows:

Input Flows

autonomous_traveler_guidance_data
dynamic_traveler_guidance_data
traveler_guidance_accepted
traveler_guidance_data
traveler_guidance_request

Input/Output

retained_traveler_guidance_data

Output Flows

autonomous_traveler_guidance_accepted
autonomous_traveler_guidance_data_request
dynamic_traveler_guidance_data_request
traveler_guidance_instructions
traveler_input_request
traveler_route_accepted

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.3;
USR = 1.3.2;
USR = 1.3.2.2;
USR = 1.3.2.3;
USR = 1.3.2.3.1;
USR = 1.3.3;
USR = 1.3.3.2;
USR = 1.3.3.2.2;
USR = 1.3.3.3;
USR = 1.3.4;
USR = 1.3.4.1;
USR = 1.3.4.1(d);
USR = 1.3.4.1(e);
USR = 1.3.4.2;
USR = 1.3.4.2.1;
USR = 1.3.4.2.2;

6.8.1.1.2 Provide Personal Portable Device Dynamic Guidance

Description:

Overview: This process shall enable dynamic traveler guidance data to be calculated. The process shall base its guidance request on the data input by the traveler from a personal portable device through other processes, and on the traveler's current location as provided by another process.

In Subsystem:

Personal Information Access

In Market Package:

ATIS6

ATIS5

Data Flows:

Input Flows

dynamic_traveler_guidance_data_request
traveler_guidance_route
traveler_location_for_dynamic_guidance

Output Flows

dynamic_traveler_guidance_data
traveler_route_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.3;
USR = 1.3.2;
USR = 1.3.2.2;
USR = 1.3.2.3;
USR = 1.3.2.3.1;
USR = 1.3.3;
USR = 1.3.3.2;
USR = 1.3.3.2.2;
USR = 1.3.3.3;
USR = 1.3.4;
USR = 1.3.4.2;
USR = 1.3.4.2.1;
USR = 1.3.4.2.2;
USR = 1.3.4.3;

6.8.1.1.3 Provide Personal Portable Device Autonomous Guidance

Description:

Overview: This process shall provide autonomous on-line guidance when requested by the traveler from a personal portable device. It shall calculate the route using data obtained from a navigable map database stored in the traveler's personal portable device. Guidance shall be provided by the process in the form of actual instructions to the traveler, e.g. cross the road here, take the subway to a specific station. The process shall provide guidance for the shortest route, within the preferences and constraints specified by the traveler in the guidance request.

In Subsystem:

Personal Information Access

In Market Package:

ATIS3

ATIS4

Data Flows:

Input Flows

autonomous_traveler_guidance_accepted
autonomous_traveler_guidance_data_request
traveler_location_for_autonomous_guidance
traveler_map_database

Output Flows

autonomous_traveler_guidance_data

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.3;
USR = 1.3.2;
USR = 1.3.2.2;
USR = 1.3.2.3;
USR = 1.3.2.3.1;
USR = 1.3.3;
USR = 1.3.3.2;
USR = 1.3.3.2.2;
USR = 1.3.3.3;
USR = 1.3.4;
USR = 1.3.4.2;
USR = 1.3.4.2.1;
USR = 1.3.4.2.2;
USR = 1.3.4.3;
USR = 1.5;
USR = 1.5.2;
USR = 1.5.2.5;
USR = 1.5.2.5(d);
USR = 1.5.2.5(g);

6.8.1.2 Provide Personal Portable Device Guidance Interface

Description:

Overview: This process shall be responsible for providing a user interface for the traveler through which personal guidance can be delivered. The process shall enable the traveler to input data to request a a suitable route. This process shall be capable of supporting two types of route guidance: dynamic (infrastructure based guidance is provided to the personal protable device), and autonomous (the personal portable device uses only locally available data- there is no information provided by the infrastructure). The process shall also act as the interface for output of on-line guidance to the traveler. Mutlimodal routes shall be supported by the process. The process shall not provide on-line guidance until the route has been accepted by the traveler. For those forms of guidance that require an on-board map database, the process shall provide an interface through which the traveler may obtain and pay for an initial copy of the database plus updates when needed. The process shall support inputs from the traveler in either manual or audio form, and shall provide outputs in audible or visual forms. It shall enable the visual output to be either in hardcopy, or display. Both types of output shall be produced in such a way that in using them the traveler does not become a hazard to other travelers.

In Subsystem:

Personal Information Access

In Market Package:

ATIS6

ATIS3

ATIS4

ATIS5

Data Flows:

Input Flows

ft_guidance_data
ft_guidance_map_update_request
ft_guidance_request
ft_guidance_route_accepted
traveler_guidance_instructions
traveler_input_request
traveler_map_update_response
traveler_personal_credit_identity

Output Flows

traveler_guidance_accepted
traveler_guidance_data
traveler_guidance_request
traveler_map_update_request
traveler_personal_map_update_cost
tt_guidance
tt_guidance_input_request
tt_guidance_map_update_response
tt_guidance_route_details

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3.0;
USR = 1.3.1;
USR = 1.3.1.1;
USR = 1.3.1.2;
USR = 1.3.1.2.1;
USR = 1.3.1.3;
USR = 1.3.2;
USR = 1.3.2.2;
USR = 1.3.2.3;
USR = 1.3.2.3.1;
USR = 1.3.3;
USR = 1.3.3.2;
USR = 1.3.3.2.2;
USR = 1.3.3.3;
USR = 1.3.4;
USR = 1.3.4.2;
USR = 1.3.4.2.1;
USR = 1.3.4.2.2;

6.8.1.3 Process Personal Portable Device Location Data

Description:

Overview: This process shall provide the traveler's current location. It shall calculate the location from one or more sources of position data such as GPS or DGPS, and shall refine its calculations using techniques such as map matching, dead reckoning, etc. The process shall provide the location to the to other processes for use in autonomous and dynamic guidance. This location should be precise as is practical within cost and technology constraints. It is intended for use by traveler personal navigation and guidance systems, as well as emergency notification systems.

In Subsystem:

Personal Information Access

In Market Package:

ATIS6

ATIS3

ATIS4

ATIS5

EM3

Data Flows:

Input Flows

From_Location_Data_Source
traveler_map_database

Output Flows

traveler_location_for_autonomous_guidance
traveler_location_for_dynamic_guidance
traveler_location_for_emergencies
traveler_location_for_information
traveler_location_for_planning

User Service Requirements Mapped to Pspec:

USR = NA;

6.8.1.4 Update Traveler Navigable Map Database

Description:

Overview: This process shall update the traveler's navigable database based on digitized data obtained from a map provider, or other appropriate data source. The update shall be initiated by the traveler through another process. The process shall have the capability to allow a financial transaction (to pay for the update) to be completed using processes in the Provide Electronic Payment Services function. When the new map data is received, it shall be loaded by the process into the traveler_map_database data store for use by other processes. The result of the update request (successful or not) shall be sent back to the traveler interface process for output to the traveler.

In Subsystem:

Personal Information Access

In Market Package:

ATIS6

ATIS3

ATIS4

ATIS5

Data Flows:

Input Flows

fmup_traveler_map_update
fmup_traveler_map_update_cost
traveler_map_update_payment_response
traveler_map_update_request

Output Flows

tmup_traveler_map_update_cost_request
tmup_traveler_map_update_request
traveler_map_database
traveler_map_update_payment_request
traveler_map_update_response

User Service Requirements Mapped to Pspec:

USR = NA;

6.8.1.5 Provide Traveler Emergency Message Interface

Description:

Overview: This process shall provide an emergency notification interface for a traveler using a personal portable device. The emergency notification interface shall enable the output of messages generated by a traveler's emergency request to another process.

In Subsystem:

Personal Information Access

In Market Package:

EM3

Data Flows:

Input Flows

emergency_message_traveler_output
traveler_location_for_information

Output Flows

tt_emergency_message

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.5.0;
USR = 1.5.2;
USR = 1.5.2.1;
USR = 1.5.2.2;
USR = 1.5.2.3;

6.8.2.1 Build Traveler Personal Security Message

Description:

Overview: This process shall respond to the input of a request from a traveler for action by the emergency services. Input of the request shall be received by the process from the traveler via a panic button or some other functionally similar form of input device provided as part of the traveler's personal portable device. When the input is received, the process shall send a message to the communications process, containing the traveler's current location and identity.

In Subsystem:

Personal Information Access

In Market Package:

EM3

Data Flows:

Input Flows

ft_personal_emergency_request
traveler_identity_store
traveler_location_for_emergencies

Output Flows

traveler_personal_emergency_request

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.1.0;
USR = 5.1.1;
USR = 5.1.1.1;
USR = 5.1.1.1(d);
USR = 5.1.1.1(e);
USR = 5.1.1.2;

6.8.2.2 Provide Traveler Emergency Communications Function

Description:

Overview: This process shall prepare and send an emergency message from a traveler's personal portable device to the Manage Emergency Services function. The message shall only be sent by the process in response to data received from another process that monitors traveler inputs. Once an emergency message has been sent, the process shall send a message to that effect to another process for output to the traveler. The process shall then await a response from the Manage Emergency Services function, and when received again send a message to the other process for output to the traveler. Output of the emergency message to the Manage Emergency Services function shall be repeated by the process at regular intervals until a response is received.

In Subsystem:

Personal Information Access

In Market Package:

EM3

Data Flows:

Input Flows

emergency_request_personal_traveler_acknowledge
traveler_personal_emergency_request

Output Flows

emergency_message_traveler_output
emergency_request_personal_traveler_details

User Service Requirements Mapped to Pspec:

USR = 5.0;
USR = 5.1.0;
USR = 5.1.1;
USR = 5.1.1.3;

6.8.3.1 Get Traveler Personal Request

Description:

Overview: This process shall receive traveler requests from a personal device (portable, or non portable) then provide support for trip planning, traffic, transit and other (yellow pages) services information, trip confirmation, yellow pages services confirmation, and payment requests. The process shall send these requests to the appropriate processes within the Provide Driver and Traveler Services function for further processing to generate responses. The interface to the traveler shall be provided through a separate process, from which input to this process originates.

In Subsystem:

Personal Information Access

In Market Package:

ATIS7

ATIS2

ATIS8

Data Flows:

Input Flows

traveler_personal_trip_planning_requests

Output Flows

traveler_personal_current_condition_request
traveler_personal_payment_information
traveler_personal_traffic_condition_request
traveler_personal_transaction_request
traveler_personal_transit_condition_request
traveler_personal_trip_confirmation
traveler_personal_trip_request
traveler_personal_yellow_pages_information_request
traveler_traffic_profile
traveler_transit_profile

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.1.0;
USR = 1.1.3;
USR = 1.1.3.2;
USR = 1.1.3.2.1;
USR = 1.1.3.2.2;
USR = 1.1.3.2.3;
USR = 1.1.3.2.4;
USR = 1.1.3.2.5;
USR = 1.1.3.2.6;
USR = 1.1.3.2.7;
USR = 1.1.3.2.8;
USR = 1.1.3.2.9;
USR = 1.1.3.2.10;
USR = 1.4.0;
USR = 1.4.1;
USR = 1.4.1.1;
USR = 1.4.1.2;
USR = 1.4.1.2(b);
USR = 1.4.1.2(c);
USR = 1.4.1.3;
USR = 1.5.0;
USR = 1.5.2;

6.8.3.2 Provide Traveler with Personal Travel Information

Description:

Overview: This process shall provide the traveler (using a personal device) with data about all requested trip, traffic, transit, other (yellow pages) services information, confirmation of any requested reservations, and payments made as part of confirmed trip plans. The data shall be sent by the process to an interface process which is responsible for its actual output to the traveler. This data shall include digitized map data to act as the background to the output when the data is shown in a suitable format. This process shall request data from other ITS functions or be sent it as a result of requests from another process.

In Subsystem:

Personal Information Access

In Market Package:

ATIS7

ATIS8

ATIS1

ATIS2

Data Flows:

Input Flows

map_data_for_traveler_personal_displays
traffic_data_for_broadcast_to_personal_devices
traffic_data_for_personal_devices
transit_deviations_for_broadcast_to_personal_devices
transit_deviations_for_personal_devices
transit_services_for_personal_devices
traveler_personal_payment_confirmation
traveler_personal_traffic_condition_request
traveler_personal_transaction_confirmation
traveler_personal_transit_condition_request
traveler_personal_trip_information
traveler_personal_yellow_pages_data

Output Flows

traffic_data_personal_request
transit_deviations_personal_request
transit_services_personal_request
traveler_personal_trip_planning_responses

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.1.0;
USR = 1.1.1;
USR = 1.1.1.1;
USR = 1.1.1.1.1;
USR = 1.1.1.1.2;
USR = 1.1.1.1.3;
USR = 1.1.1.1.4;
USR = 1.1.1.1.5;
USR = 1.1.1.1.6;
USR = 1.1.2;
USR = 1.1.2.1;
USR = 1.1.2.1.1;
USR = 1.1.2.1.2;
USR = 1.1.2.1.3;
USR = 1.1.2.1.4;

USR = 1.1.2.1.5;
USR = 1.1.2.1.6;
USR = 1.1.2.1.8;
USR = 1.1.4;
USR = 1.1.4.1;
USR = 1.1.4.1.1;
USR = 1.1.4.1.2;
USR = 1.1.4.1.3;
USR = 1.1.4.1.4;
USR = 1.4.0;
USR = 1.4.1;
USR = 1.4.1.3;
USR = 1.5.0;
USR = 1.5.2;
USR = 1.5.2.1;
USR = 1.5.2.2;
USR = 2.0;
USR = 2.3.0;
USR = 2.3.1;
USR = 2.3.1.3;

6.8.3.3 Provide Traveler Personal Interface

Description:

Overview: This process shall provide an interface in a personal device through which travelers can plan and confirm trips, as well as obtain current traffic and transit information. The process shall support trip planning and confirmation of other (yellow pages) services such as lodging, restaurants, theaters, and other tourist activities. The process shall be able to load in the traveler_personal_regular_data store frequently used information such as traveler identity (the owner of the personal device), home and work locations, etc. This will reduce the amount of input needed by the traveler for each trip request. The process shall also carry out input data verification and require input confirmation, with the traveler, before passing the data to other processes. The traveler's payment information and location (when traveler is using a portable device) shall be obtained by this process from other processes. The process shall support inputs from the traveler in both manual and audio form, and shall provide its outputs in audible and visual forms that are consistent with a personal device. This process shall include forms suitable for travelers with hearing and vision physical disabilities. The process shall display data for as long as required by the traveler and must enable viewing of previously output data. When used with a portable device, the process shall provide the traveler the option to filter the data (to be displayed) relevant to the travelers current location.

In Subsystem:

Personal Information Access

In Market Package:

ATIS6

ATIS7

ATIS1

ATIS2

ATIS5

ATIS8

Data Flows:

Input Flows

ft_personal_extra_trip_data
ft_personal_map_display_update_request
ft_personal_trip_planning_requests
traveler_location_for_planning
traveler_personal_credit_identity
traveler_personal_display_map_update_response
traveler_personal_trip_planning_responses

Input/Output

traveler_personal_regular_data

Output Flows

traveler_personal_display_map_update_request
traveler_personal_display_update_cost
traveler_personal_trip_costs
traveler_personal_trip_planning_requests
tt_personal_extra_trip_data_request
tt_personal_trip_planning_responses

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.1.0;
USR = 1.1.3;
USR = 1.1.3.2;
USR = 1.1.3.2.1;
USR = 1.1.3.2.2;
USR = 1.1.3.2.3;
USR = 1.1.3.2.4;
USR = 1.1.3.2.5;
USR = 1.1.3.2.6;
USR = 1.1.3.2.7;
USR = 1.1.3.2.8;
USR = 1.1.3.2.9;
USR = 1.1.3.2.10;
USR = 1.1.4;
USR = 1.1.4.2;
USR = 1.1.4.2.1;
USR = 1.3;
USR = 1.3.4;
USR = 1.3.4.1;
USR = 1.3.4.1(b);
USR = 1.3.4.1(c);
USR = 1.4.0;
USR = 1.4.1;
USR = 1.4.1.1;
USR = 1.4.1.2;
USR = 1.4.1.3;
USR = 1.4.1.4;
USR = 1.5;
USR = 1.5.2;
USR = 1.5.2.5;
USR = 1.5.2.5(b);
USR = 1.5.2.5(c);
USR = 1.5.2.5(d);
USR = 2.0;
USR = 2.3.0;

6.8.3.4 Update Traveler Personal Display Map Data

Description:

Overview: This process shall provide updates to the digitized map data used as the background for displays on travelers' personal devices. These displays include details of traffic, trip and travel information for use by travelers. The process shall obtain the new map data from a map provider process or some other appropriate data source on request from the traveler via the traveler interface process. The process shall load data into the map_data_for_traveler_personal_displays data store. The data will be compatible with the types of displays that are found on personal devices.

In Subsystem:

Personal Information Access

In Market Package:

ATIS3

ATIS4

Data Flows:

Input Flows

fmup_traveler_personal_display_update
fmup_traveler_personal_display_update_cost
traveler_personal_display_map_update_request
traveler_personal_display_update_payment_response
traveler_personal_regular_data

Output Flows

map_data_for_traveler_personal_displays
tmup_request_traveler_personal_display_update
tmup_request_traveler_personal_display_update_cost
traveler_personal_display_map_update_response
traveler_personal_display_update_payment_request

User Service Requirements Mapped to Pspec:

USR = 1.3;
USR = 1.3.4;
USR = 1.3.4.1;
USR = 1.3.4.1(b);
USR = 1.3.4.1(c);
USR = 1.5;
USR = 1.5.2;
USR = 1.5.2.5;
USR = 1.5.2.5(c);
USR = 1.5.2.5(d);

7.1.1.1 Read Tag Data for Tolls

Description:

Overview: This process shall be responsible for requesting the data from the toll tag being carried on-board the vehicle and used as a payment instrument. If there is no tag or the data it contains cannot be properly read, this process shall provide a message for the vehicle operator to contact the toll authority (or toll system operator). The process shall send a request to other processes to obtain an image of the vehicle. If the vehicle is exiting a closed toll system the data shall be checked by this process to see if it contains an entry point toll segment number. If not present, the process would be referred to another process for off-line resolution. If the toll segment identity is present, it shall be combined with the vehicle characteristics, e.g., size, type, etc., to form the data upon which the toll payment transaction can be based, and the data sent to another process. If the vehicle is entering a closed toll system, the entry point toll segment shall be written onto the tag so that it can be used as the mechanism for charging for the use of the toll road.

In Subsystem:

Toll Collection

In Market Package:

ATMS10

Data Flows:

Input Flows

toll_tag_data_collect
vehicle_toll_characteristic_data

Output Flows

get_toll_tag_violator_image
toll_tag_data_request
toll_tag_data_update
toll_tag_problem_message
vehicle_tag_for_tolls
vehicle_type_for_tolls

User Service Requirements Mapped to Pspec:

USR = 1.8.2;
USR = 1.8.2.13;
USR = 1.8.2.13(c);
USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.1;
USR = 3.1.1.6;

7.1.1.10 Determine Advanced Toll Bill

Description:

Overview: This process shall be responsible for receiving a request to pay an advanced toll. It shall obtain the price of the toll segment(s) for which advanced payment is being requested from a local data store and shall then forward it to the billing processes. The store of toll prices shall be maintained by another process.

In Subsystem:

Toll Collection

In Market Package:

ATMS10

Data Flows:

Input Flows

advanced_toll_needed
toll_price_data_for_advanced_toll

Output Flows

advanced_toll_billing

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.1;

7.1.1.11 Manage Toll Archive Data

Description:

Overview: This process shall obtain toll operational data and toll pricing data and distribute it to the Manage Archived Data function. As data is received into this process quality control metrics shall be assigned. The appropriate meta-data shall be generated and store along with the data. A catalog of the data shall be maintained to allow requesters to know what data is available from the archive store. The process shall run when a request for data is received from an external source, or when fresh data is received.

In Subsystem:

Toll Administration

In Market Package:

AD1

AD2

Data Flows:

Input Flows

fta_archive_commands
toll_archive_request
toll_archive_status
toll_operational_data
toll_prices_for_archive

Input/Output

toll_data_archive

Output Flows

toll_archive_data
tta_archive_status

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.3;
USR = 7.1.3.1.2;

7.1.1.2 Calculate Vehicle Toll

Description:

Overview: This process shall be responsible for calculating the toll for the detected vehicle based on the vehicle's characteristics and data obtained from the tag being carried by the vehicle. This process shall calculate the cost of the toll using segment(s) traveled by the vehicle. Segment information is obtained by reading data that contains standard prices for toll segments plus any variations to pricing received from the toll operator.

In Subsystem:

Toll Collection

In Market Package:

ATMS10

Data Flows:

Input Flows

fto_local_toll_price_variations
toll_price_data_for_vehicle_toll
vehicle_tag_for_tolls
vehicle_type_for_tolls

Output Flows

toll_charge

User Service Requirements Mapped to Pspec:

USR = 1.8.2;
USR = 1.8.2.13;
USR = 1.8.2.13(c);
USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.1;
USR = 3.1.1.2;
USR = 3.1.1.6;

7.1.1.3 Manage Bad Toll Payment Data

Description:

Overview: This process shall be responsible for maintaining a data store containing a list of invalid driver credit identities. This process shall use this data to verify credit identities and commercial vehicle carrier numbers provided for checking by the billing process. Verification shall ensure that the current toll payment transaction is using a credit identity or carrier identity that has not previously had an invalid transaction. Details of potential invalid credit identities or carrier numbers shall be sent by this process to the financial institution for verification. This process shall also receive from the financial institution details of invalid payment instrument data that has been found by other means.

In Subsystem:

Toll Administration

In Market Package:

ATMS10

Data Flows:

Input Flows

ffi_bad_toll_payment_updates
toll_bad_payment_check_request
toll_payment_violator_data

Input/Output

bad_toll_payment_list

Output Flows

tfi_toll_payment_violator_data
toll_bad_payment_check_response

User Service Requirements Mapped to Pspec:

USR = 1.8.2;
USR = 1.8.2.1;
USR = 1.8.2.1(f);
USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.4;
USR = 3.1.1.5;

7.1.1.4 Check for Advanced Tolls Payment

Description:

Overview: This process shall be responsible for checking to see if the required toll payment has already been made. The process shall determine the existence of an advanced payment for the toll segment(s) by comparing the received payment information with that in the store containing the list of advanced payments. If the payment has already been made then the process shall remove the requirement for local billing and remove the record of the advanced payment from the store. Details of each payment transaction shall be sent by the process to another process with the payment information received from the driver removed.

In Subsystem:

Toll Collection

In Market Package:

ATMS10

Data Flows:

Input Flows

advanced_toll_billing
toll_charge

Input/Output

advanced_toll_payment_list

Output Flows

advanced_toll_transactions
billing_for_tolls_needed
tto_transaction_reports

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.5;

7.1.1.5 Bill Driver for Tolls

Description:

Overview: This process shall be responsible for either obtaining payment for the current or advanced toll. The process shall achieve this either by requesting that the toll cost be deducted from the credit being stored by the toll tag that is acting as the payment instrument, or by informing the driver that payment for the toll will be debited to the credit identity provided by the tag. Before sending data to the tag, the process shall check that either the credit identity is not already in the list of bad payers, or the stored credit is not less that the toll cost. If either of these conditions is true, the process shall obtain an image of the driver and vehicle which can be forwarded to the appropriate enforcement agency via another process. When the appropriate payment transaction has been completed, the toll entry segment identity shall be cleared from the tag so that it can be used the next time that the vehicle is on a toll road. The tag may be in the form of some type of credit or debit card, or an electronic purse. Details of the transaction shall always be sent by this process to the process that manages toll transactions. Where an advanced toll payment is identified, the process shall take no action if the credit identity is on the bad payers list, or the stored credit is less than the toll cost, other than the payment is not confirmed.

In Subsystem:

Toll Collection

In Market Package:

ATMS10

Data Flows:

Input Flows

billing_for_tolls_needed
toll_bad_payment_check_response
toll_payment_confirmation

Output Flows

advanced_toll_payment_list
confirm_advanced_tolls_payment
current_toll_transactions
get_toll_payment_violator_image
toll_bad_payment_check_request
toll_payment_debited
toll_payment_pull_in_message
toll_payment_request
toll_payment_violator_data
toll_tag_data_clear

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.1;
USR = 3.1.1.8;

7.1.1.6 Collect Probe Data From Toll Transactions

Description:

Overview: This process shall calculate the time taken for vehicles to travel between successive toll plazas and send it to the Manage Traffic and Provide Driver and Traveler Services functions. The process shall periodically request the data from the process that manages toll financial processing and ensure that any references to the driver and/or vehicle identity plus any other payment information are removed from the data before it is sent to the other functions.

In Subsystem:

Toll Administration

In Market Package:

ATMS10

Data Flows:

Input Flows

toll_transactions_for_probe_data

Output Flows

probe_data_for_traffic
toll_transactions_for_probe_data_request
vehicle_toll_probe_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.1;
USR = 3.1.1.4;
USR = 3.1.1.6;

7.1.1.7 Update Toll Price Data

Description:

Overview: This process shall be responsible for maintaining a store of data containing the toll price, which may vary according to the type of vehicle. The process shall also act as the interface for the output and input of responses to toll price change requests from the Manage Traffic function, the provision of toll price information to the Centralized Payments facility, and to the toll administrator to enable changes to be made to the stored data. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Toll Administration

In Market Package:

ATMS10

Data Flows:

Input Flows

fta_toll_price_changes_response
fta_toll_price_data
toll_price_changes_request
toll_price_data_request
toll_price_direct_request

Input/Output

toll_prices

Output Flows

toll_price_changes_response
toll_price_data
toll_price_data_for_advanced_toll
toll_price_data_for_vehicle_toll
toll_price_direct_details
toll_prices_for_archive
tta_toll_price_changes_request

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.2;

7.1.1.8 Register for Advanced Toll Payment

Description:

Overview: This process shall be responsible for responding to requests for tolls to be paid in advance. It shall provide the toll administrator with the opportunity to review the requests for advanced toll payments. If approved, the advanced toll data shall be forwarded by the process to other processes for the actual toll cost to be obtained and payment transactions initiated.

In Subsystem:

Toll Administration

In Market Package:

ATMS10

Data Flows:

Input Flows

advanced_other_tolls_request
advanced_traveler_tolls_request
confirm_advanced_tolls_payment
fta_confirm_advanced_toll

Output Flows

advanced_other_tolls_confirm
advanced_toll_needed
advanced_traveler_tolls_confirm
tta_request_advanced_toll

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.1;

7.1.1.9 Manage Toll Financial Processing

Description:

Overview: This process shall be responsible for maintaining a log of all toll transactions that are carried out by other processes in the toll payments system. At periodic intervals the process shall output the accumulated records to the toll administrator and the archive function. It shall also output the data on request to the process that calculates probe data from the average travel time between toll plazas. The identity of the payee shall be removed from the data before it is used in any of these outputs. The process shall also be responsible for sending details of transactions to the financial institution to enable the users to be billed through their credit identities. For commercial vehicles, this will be done using the data provided by the vehicle's on-board tag and shall enable billing to the financial institution to be made by carrier.

In Subsystem:

Toll Administration

In Market Package:

ATMS10

Data Flows:

Input Flows

advanced_toll_transactions
current_toll_transactions
ffi_confirm_toll_payment
toll_transactions_for_probe_data_request

Input/Output

toll_transaction_records

Output Flows

tfi_request_toll_payment
toll_operational_data
toll_transactions_for_probe_data
tta_transaction_reports

User Service Requirements Mapped to Pspec:

USR = 1.8.2;
USR = 1.8.2.4;
USR = 1.8.2.4(e);
USR = 1.8.2.10;
USR = 1.8.2.10(a);
USR = 1.8.2.12;
USR = 1.8.2.12(a);
USR = 3.0;
USR = 3.1;
USR = 3.1.4;
USR = 3.1.4.3;

7.1.2 Produce Roadside Displays

Description:

Overview: This process shall be responsible for driving the displays that tell vehicles whether or not their driver's toll payment has been confirmed or rejected. The process shall receive the data for output via the displays from other processes. The data input and output forms shall use an appropriate form of display that shall be easily readable under all lighting conditions and over the range of speeds that vehicles are expected to use when passing through the toll plaza. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Toll Collection

In Market Package:

ATMS10

Data Flows:

Input Flows

toll_payment_pull_in_message
toll_tag_problem_message

Output Flows

td_toll_payment_confirmed
td_toll_payment_invalid

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.3;

7.1.3 Obtain Toll Violator Image

Description:

Overview: This process shall be responsible for obtaining an image of a violator for use by other processes. The form of the image data obtained by this process shall be very accurate so that there can be no mistake of the determination of the identity of the vehicle and/or driver, and shall be easily passed on by the other processes to the appropriate law enforcement agency(ies) so that punitive action may be taken. The process shall be capable of obtaining an image of the required accuracy under all lighting conditions and over the range of speeds with which vehicles will pass through the toll plaza.

In Subsystem:

Toll Collection

In Market Package:

ATMS10

Data Flows:

Input Flows

From_Vehicle_Characteristics
get_toll_payment_violator_image
get_toll_tag_violator_image

Output Flows

toll_violation_information

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.4;

7.1.4 Provide Driver Toll Payment Interface

Description:

Overview: This process shall be responsible for providing an interface through which drivers can request and pay for other services when paying their tolls at toll plazas. The services supported by this process include advanced payment for parking lot charges and transit fares. The process shall query the driver for sufficient information to enable the advanced parking lot charge and/or transit fare to be determined and the cost either billed to a credit identity provided by the driver's payment instrument, or deducted from credit stored on the instrument. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Vehicle

In Market Package:

ATMS10

ATMS16

Data Flows:

Input Flows

advanced_fares_and_charges_response
driver_toll_payment_credit_identity
fbv_vehicle_identity
fd_other_services_toll_request

Output Flows

advanced_fares_and_charges_request
driver_advanced_payment_at_toll
td_other_services_toll_response

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.1;
USR = 3.1.1.3;
USR = 3.1.1.7;

7.1.5 Detect Vehicle for Tolls

Description:

Overview: This process shall be responsible for producing a vehicle's characteristics from data received by sensors located at the roadside, at or near the toll collection point. The data shall be sent by the process to another process in a form suitable for use in calculating the toll cost for the vehicle. The process shall ensure that the data includes such things as vehicle size, weight, axle count, type, identifiable features, etc.

In Subsystem:

Toll Collection

In Market Package:

ATMS10

Data Flows:

Input Flows

From_Vehicle_Characteristics

Output Flows

vehicle_toll_characteristic_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.4;
USR = 3.1.1.8;

7.1.6 Distribute Advanced Charges and Fares

Description:

Overview: This process shall be responsible for receiving requests for advanced payment of tolls from the parking lot charge or transit fare collection facilities within the Provide Electronic Payment Services function. It shall pass the requests on to another process in the toll collection facility, and shall return transaction success or failure details to the requesting process. The process shall also receive requests for the advanced payment of parking lot charges and transit fares from the toll payment interface process. It shall send these requests to other processes in the Provide Electronic Payment Services function and when received, return the results to the toll payment interface process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

ATIS2

Data Flows:

Input Flows

advanced_fares_and_charges_request
advanced_other_tolls_confirm
transfer_charges_to_tolls
transfer_fares_to_tolls

Output Flows

advanced_fares_and_charges_response
advanced_other_tolls_request
transfer_tolls_to_charges
transfer_tolls_to_fares

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.2;
USR = 3.1.2.1;
USR = 3.1.3;
USR = 3.1.3.1;

7.1.7 Provide Payment Instrument Interface for Tolls

Description:

Overview: This process shall be responsible for providing the interface through which the payment information can be read from a vehicle tag. The process shall enable the use of the data from the tag for the purposes of paying for current tolls, plus if required, the cost of advanced parking lot charges, and/or transit fares, as well as providing the data for use in traffic flow analysis. The tag data which can be collected by the process shall include credit identity, stored credit value, and the toll segment identity at the vehicle's entry point so that closed toll system can be used. When stored credit is used, the process shall enable the deduction of the cost of the toll and (possibly) advanced payments from the credit value on the tag. The process shall support collection of data from tags on-board a range of vehicle types including private cars or vans, commercial vehicles, transit vehicles, including those used for demand responsive transit services.

In Subsystem:

Vehicle

In Market Package:

ATMS10

ATMS16

Data Flows:

Input Flows

driver_advanced_payment_at_toll
fpi_confirm_payment_at_toll_plaza
fpi_toll_tag_data
toll_payment_debited
toll_payment_request
toll_tag_data_clear
toll_tag_data_needed
toll_tag_data_request
toll_tag_data_update

Input/Output

toll_tag_data_store

Output Flows

driver_toll_payment_credit_identity
toll_payment_confirmation
toll_tag_data_collect
toll_tag_data_input
tpi_debited_payment_at_toll_plaza
tpi_request_payment_at_toll_plaza

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.3;
USR = 3.1.3.1;

7.2.1.1 Read Parking Lot Tag Data

Description:

Overview: This process shall be responsible for requesting the data from the parking lot tag being carried on-board the vehicle and used as the payment instrument being read. If there is no tag or the data it contains cannot be properly read, the process shall send a message for the vehicle to pull in for output by another process, and send a request to other processes to obtain an image of the vehicle. If there is no entry time data on the tag, then the process shall re-write this data plus the number of the entry lane onto the tag, so that it can be used as the mechanism for charging for the use of the parking lot. If the entry time is present, the process shall combine it with the vehicle characteristics, e.g., size, type, etc. to form the data upon which the parking lot payment transaction can be based, and send it to another process.

In Subsystem:

Parking Management

In Market Package:

ATMS16

Data Flows:

Input Flows

parking_lot_tag_data_collect
vehicle_parking_lot_characteristic_data

Output Flows

get_parking_lot_tag_violator_image
parking_lot_tag_data_request
parking_lot_tag_data_update
parking_lot_tag_pull_in_message
vehicle_tag_for_charges
vehicle_type_for_charges

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.3;
USR = 3.1.3.1;

7.2.1.10 Determine Advanced Charges

Description:

Overview: This process shall be responsible for receiving a request to pay an advanced parking lot charge. It shall obtain the required parking lot charge from a data store and shall then forward the data to the billing processes. The store of parking lot charges shall be maintained by another process.

In Subsystem:

Parking Management

In Market Package:

ATMS16

Data Flows:

Input Flows

advanced_charges_needed
parking_lot_prices

Output Flows

advanced_parking_lot_billing

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.3;
USR = 3.1.3.2;
USR = 3.1.4;
USR = 3.1.4.1;

7.2.1.2 Calculate Vehicle Parking Lot Charges

Description:

Overview: This process shall be responsible for calculating the parking lot charge for the detected vehicle based on its characteristics and data obtained from the tag being carried by the vehicle. The process shall obtain the cost of the use of the parking lot by reading data from a store that contains the standard prices for parking lot charges.

In Subsystem:

Parking Management

In Market Package:

ATMS16

Data Flows:

Input Flows

parking_lot_prices
vehicle_tag_for_charges
vehicle_type_for_charges

Output Flows

parking_lot_charge

User Service Requirements Mapped to Pspec:

USR = 1.8.2;
USR = 1.8.2.13;
USR = 1.8.2.13(c);
USR = 3.0;
USR = 3.1;
USR = 3.1.3;
USR = 3.1.3.1;
USR = 3.1.3.3;

7.2.1.3 Collect Bad Charge Payment Data

Description:

Overview: This process shall be responsible for maintaining a data store containing a list of invalid driver credit identities. The process shall use this data to check credit identities provided for checking by the billing process. This checking shall ensure that the current parking lot payment transaction is using a credit identity that has not previously had an invalid transaction. Details of possible invalid credit identities shall be sent by the process to the financial institution for verification. The process shall also receive from the financial institution details of invalid payment instrument data that has been found by other means.

In Subsystem:

Parking Management

In Market Package:

ATMS16

Data Flows:

Input Flows

charge_payment_violator_data
ffi_bad_charges_payment_updates

Output Flows

bad_charge_payment_list
tfi_parking_lot_payment_violator_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.3;
USR = 3.1.3.1;

7.2.1.4 Check for Advanced Parking Lot Payment

Description:

Overview: This process shall be responsible for checking to see if the required parking lot charge payment has already been made. The process shall determine the existence of an advanced payment for the parking lot charges by comparing the received payment information with that in the store containing the list of advanced payments. If the payment has already been made then the process shall remove the requirement for local billing and remove the record of the advanced payment from the store. Details of each payment transaction shall be sent by the process to another process with the payment information received from the driver removed.

In Subsystem:

Parking Management

In Market Package:

ATMS16

Data Flows:

Input Flows

advanced_parking_lot_billing
parking_lot_charge

Input/Output

advanced_charges_payment_list

Output Flows

advanced_charge_transactions
billing_for_charges_needed

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.3;
USR = 3.1.3.1;
USR = 3.1.3.2;

7.2.1.5 Bill Driver for Parking Lot Charges

Description:

Overview: This process shall be responsible for either obtaining payment for the current or advanced parking lot charge. The process shall achieve this either by requesting that the charge be deducted from the credit being stored by the parking lot tag that is acting as the payment instrument, or informing the driver that payment for the charge will be debited from the credit identity provided by the tag. Before sending data to the tag, the process shall check that either the credit identity is not already in the list of bad payers, or the stored credit is not less that the parking lot charge. If either of these conditions is true the process shall obtain an image of the driver and vehicle which can be forwarded to the appropriate enforcement agency via another process. When the appropriate payment transaction has been completed, the parking lot entry time data shall be cleared from the tag so that it can be used for the next visit by the vehicle to a parking lot. The tag may be in the form of some type of credit or debit card, or an electronic purse. Details of the transaction shall always be sent to the process that manages parking lot transactions which will also send details to the financial institution if a credit or debit card is involved. Where an advanced parking lot charge payment is identified, no action is taken if the credit identity is on the bad payers list, or the stored credit is less than the charge, other than the payment is not confirmed.

In Subsystem:

Parking Management

In Market Package:

ATMS16

Data Flows:

Input Flows

bad_charge_payment_list
billing_for_charges_needed
parking_lot_payment_confirmation

Output Flows

advanced_charges_payment_list
charge_payment_violator_data
confirm_advanced_charges_payment
current_charge_transactions
get_charge_payment_violator_image
parking_lot_payment_debited
parking_lot_payment_pull_in_message
parking_lot_payment_request
parking_lot_tag_data_clear

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.3;
USR = 3.1.3.1;

7.2.1.6 Manage Parking Lot Financial Processing

Description:

Overview: This process shall be responsible for maintaining a log of all transactions that are carried out by other processes in the Process Electronic Parking Lot Payment facility. The identity of the payee shall have been removed from the data before it is stored. At periodic intervals the process shall output the accumulated records to another process in the Provide Electronic Payment Services function. It shall also output the same data on request to the parking operator, either in hardcopy form, or as a visual display. The process shall be responsible for sending details of transactions to the financial institution to enable the users to be billed through their credit identities. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Parking Management

In Market Package:

ATMS16

Data Flows:

Input Flows

advanced_charge_transactions
current_charge_transactions
ffi_confirm_charges_payment
fpo_transaction_reports_request

Input/Output

parking_lot_transaction_records

Output Flows

tfi_request_charges_payment
tpo_transaction_reports

User Service Requirements Mapped to Pspec:

USR = 1.8;
USR = 1.8.1;
USR = 1.8.1.6;
USR = 1.8.1.6(e);
USR = 1.8.2;
USR = 1.8.2.12;
USR = 1.8.2.12(b);
USR = 3.0;
USR = 3.1;
USR = 3.1.3;
USR = 3.1.3.1;

7.2.1.7 Update Parking Lot Data

Description:

Overview: This process shall be responsible for maintaining a store of data containing the parking lot charges, which may vary according to the type of vehicle. The process shall also act as the interface to the parking service provider to enable changes to be made to the stored data, for the output and input of responses to parking lot price change requests from the Manage Traffic function, and for requests for parking lot price data from the Centralized Payments facility. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Parking Management

In Market Package:

ATMS16

ATMS19

Data Flows:

Input Flows

fpo_parking_lot_charge_change_response
fpo_parking_lot_data
other_parking_lot_price_data_request
parking_charge_request_for_archive
parking_lot_capacity_update_confirm
parking_lot_charge_change_request
parking_lot_charge_direct_request
parking_lot_price_data_request

Output Flows

other_parking_lot_price_data
parking_charge_response_for_archive
parking_lot_capacity_update
parking_lot_charge_change_response
parking_lot_charge_direct_details
parking_lot_price_data
parking_lot_prices
tpo_parking_lot_charge_change_request

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.3;
USR = 3.1.3.1;
USR = 3.1.3.3;

7.2.1.8 Register for Advanced Parking Lot Payment

Description:

Overview: This process shall be responsible for responding to requests for parking lot charges to be paid in advance. It shall provide the parking operator with the opportunity to deny the request for advanced payment of a parking lot charge. If approved, the advanced parking lot charge data shall be forwarded by the process to other processes for the actual cost to be obtained and the payment transactions initiated.

In Subsystem:

Parking Management

In Market Package:

ATMS16

Data Flows:

Input Flows

advanced_other_charges_request
advanced_traveler_charges_request
confirm_advanced_charges_payment
fpo_confirm_advanced_parking_payment
parking_lot_bookings_confirm

Output Flows

advanced_charges_needed
advanced_other_charges_confirm
advanced_traveler_charges_confirm
parking_lot_bookings_request
tpo_request_advanced_parking_payment

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.3;
USR = 3.1.3.1;
USR = 3.1.3.2;
USR = 3.1.4;
USR = 3.1.4.1;

7.2.1.9 Manage Parking Lot Reservations

Description:

Overview: This process shall be responsible for maintaining a store of parking lot data. This data shall cover the capacity of the parking lot, i.e., the maximum number of spaces available, which may vary according to the type of vehicle. The process shall also act as the interface for inquiries from other ITS functions both for details of parking lot capacity, both now and in the future and for the reservation of spaces as part of travelers' confirmed trips. The parking lot data also contains data on the hours of operation of parking lots. This data is used in transactions requiring electronic payment of parking lot services, as well as for a traveler making a parking lot reservation.

In Subsystem:

Parking Management

In Market Package:

ATMS16

ATMS19

Data Flows:

Input Flows

fpo_parking_lot_hours_of_operation
parking_lot_bookings_request
parking_lot_capacity_update
parking_lot_data_request
parking_lot_reservation_request

Input/Output

parking_lot_data

Output Flows

parking_lot_availability
parking_lot_bookings_confirm
parking_lot_capacity_update_confirm
parking_lot_reservation_confirm

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.4;
USR = 3.1.4.3;

7.2.2 Produce Parking Lot Displays

Description:

Overview: This process shall be responsible for driving the displays that tell vehicles whether or not their parking lot charge payment has been confirmed or rejected. The process shall receive the data for output via the displays from other processes. The data input and output formats shall use an appropriate form of display that shall be easily readable under all lighting conditions and over the range of speeds that vehicles are expected to use when entering or leaving a parking lot. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Parking Management

In Market Package:

ATMS16

Data Flows:

Input Flows

parking_lot_payment_pull_in_message
parking_lot_tag_pull_in_message

Output Flows

td_parking_lot_payment_confirmed
td_parking_lot_payment_invalid

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.3;
USR = 3.1.3.1;

7.2.3 Obtain Parking Lot Violator Image

Description:

Overview: This process shall be responsible for obtaining an image of a violator for use by other processes. The form of the image data obtained by this process shall be very accurate so that there can be no mistake of the determination of the identity of the vehicle and/or driver, and shall be easily passed on by the other processes to the appropriate law enforcement agency(ies) so that punitive action may be taken. The process shall be capable of obtaining an image of the required accuracy under all lighting conditions and over the range of speeds with which vehicles will enter or leave parking lots.

In Subsystem:

Parking Management

In Market Package:

ATMS16

Data Flows:

Input Flows

From_Vehicle_Characteristics
get_charge_payment_violator_image
get_parking_lot_tag_violator_image

Output Flows

parking_lot_violation_information

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.3;

7.2.4 Provide Driver Parking Lot Payment Interface

Description:

Overview: This process shall be responsible for providing an interface through which drivers can request other services when paying their charges at parking lots. The services supported by this process include advanced parking lot payment, as well as advanced payment for tolls and transit fares. The process shall query the driver for sufficient information to enable the advanced toll, parking lot charge, and/or transit fare to be determined and the cost either billed to a credit identity provided by the driver's payment instrument, or deducted from credit stored on the instrument. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Vehicle

In Market Package:

ATMS10

ATMS16

Data Flows:

Input Flows

advanced_tolls_and_fares_response
driver_parking_payment_credit_identity
fbv_vehicle_identity
fd_other_services_parking_request

Output Flows

advanced_tolls_and_fares_request
driver_advanced_payment_at_lot
td_other_services_parking_response

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.4;
USR = 3.1.4.1;
USR = 3.1.4.4;

7.2.5 Detect Vehicle for Parking Lot Payment

Description:

Overview: This process shall be responsible for producing a vehicle's characteristics from data received by sensors located at or near the parking lot entry and exit lanes. The data shall be sent by the process to another process in a form suitable for use in calculating the parking lot charge for the vehicle. The process shall ensure that the data includes such things as vehicle size, type, identifiable features, etc.

In Subsystem:

Parking Management

In Market Package:

ATMS16

ATMS16

Data Flows:

Input Flows

From_Vehicle_Characteristics

Output Flows

vehicle_parking_lot_characteristic_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.3;
USR = 3.1.3.1;
USR = 3.1.3.3;

7.2.6 Distribute Advanced Tolls and Fares

Description:

Overview: This process shall be responsible for receiving requests for advanced payment of parking lot charges from the toll or transit fare collection facilities within the Provide Electronic Payment Services function. It shall pass the requests on to another process in the Provide Electronic Parking Lot Payment facility, and shall return transaction success or failure details to the requesting process. The process shall also receive requests for the advanced payment of tolls and transit fares from the parking lot payment interface process. It shall send these requests to other processes in the Provide Electronic Payment Services function and when received, return the results to the Parking Lot payment interface process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

ATIS2

Data Flows:

Input Flows

advanced_other_charges_confirm
advanced_tolls_and_fares_request
transfer_fares_to_charges
transfer_tolls_to_charges

Output Flows

advanced_other_charges_request
advanced_tolls_and_fares_response
transfer_charges_to_fares
transfer_charges_to_tolls

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.1;
USR = 3.1.1.3;
USR = 3.1.2;
USR = 3.1.2.1;
USR = 3.1.2.4;
USR = 3.1.4;
USR = 3.1.4.1;

7.2.7 Provide Payment Instrument Interface for Parking

Description:

Overview: This process shall be responsible for providing the interface through which the payment information can be read from a vehicle tag. The process shall enable the use of the data from the tag for the purposes of paying the current parking lot charge and if required, advanced payments for tolls and/or transit fares. It shall be possible for the process to collect either the credit identity or the stored credit value data from the tag, and to update the stored credit value as a result of the parking lot charge and (possibly) advanced charges having been paid. The time at which the vehicle entered the parking lot shall also be collected from the tag by the process so that the charge for the use of the lot can be calculated. The process shall support collection of data from tags on-board a range of vehicle types including private cars or vans, commercial vehicles, transit vehicles, including those used for demand responsive transit services.

In Subsystem:

Vehicle

In Market Package:

ATMS10

ATMS16

Data Flows:

Input Flows

driver_advanced_payment_at_lot
fpi_confirm_payment_at_parking_lot
fpi_parking_tag_data
parking_lot_payment_debited
parking_lot_payment_request
parking_lot_tag_data_clear
parking_lot_tag_data_needed
parking_lot_tag_data_read
parking_lot_tag_data_request
parking_lot_tag_data_update

Output Flows

driver_parking_payment_credit_identity
parking_lot_payment_confirmation
parking_lot_tag_data_collect
parking_lot_tag_data_input
parking_lot_tag_data_write
tpi_debited_payment_at_parking_lot
tpi_request_payment_at_parking_lot

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.3;
USR = 3.1.3.1;

7.3.1.1 Register for Advanced Transit Fare Payment

Description:

Overview: This process shall be responsible for responding to requests for transit fares to be paid in advance. The advanced transit fare data shall be forwarded by the process to other processes for the actual cost to be obtained and the payment transactions initiated.

In Subsystem:

Public Transport Management

In Market Package:

APTS4

Data Flows:

Input Flows

advanced_other_fares_request
advanced_traveler_fares_request
confirm_advanced_fares_payment

Output Flows

advanced_fares_needed
advanced_other_fares_confirm
advanced_traveler_fares_confirm

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.2;
USR = 3.1.2.1;
USR = 3.1.2.4;
USR = 3.1.2.8;

7.3.1.2 Determine Advanced Transit Fares

Description:

Overview: This process shall be responsible for receiving a request to pay an advanced transit fare. It shall obtain the required transit fare data from a local store of transit fares and shall then forward the data to the billing processes. The store of fare data shall be maintained by another process.

In Subsystem:

Public Transport Management

In Market Package:

APTS4

Data Flows:

Input Flows

advanced_fares_needed
transit_fares_for_advanced_payments
transit_services_for_advanced_fares

Output Flows

advanced_fare_billing

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.2;
USR = 3.1.2.1;
USR = 3.1.2.2;
USR = 3.1.2.8;

7.3.1.3 Manage Transit Fare Financial Processing

Description:

Overview: This process shall be responsible for maintaining a log of all the transactions carried out by other processes in the Process Electronic Transit Fare Payment facility. The identity of the payee shall have been removed from the data before it is stored. At periodic intervals the process shall output the accumulated records to the transit fleet manager, the transit system operator and to another process in the Provide Electronic Payment Services function. The process shall also be responsible for sending details of transactions to the financial institution to enable the users to be billed through their credit identities. The input and output forms shall include those that are suitable for travelers with physical disabilities.

In Subsystem:

Public Transport Management

In Market Package:

APTS4

Data Flows:

Input Flows

advanced_fare_transactions
current_fare_transactions
ffi_confirm_fare_payment

Input/Output

transit_fare_transaction_records

Output Flows

tfi_request_fare_payment
transit_fare_transactions
ttfm_transaction_reports
ttso_transaction_reports

User Service Requirements Mapped to Pspec:

USR = 1.8.2;
USR = 1.8.2.10;
USR = 1.8.2.10(a);
USR = 2.3.0;
USR = 2.3.3;
USR = 2.3.3.1;
USR = 2.3.3.1(c);
USR = 2.3.3.3;
USR = 3.0;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.1;
USR = 3.1.2.3;
USR = 3.1.2.5;

7.3.1.4 Check for Advanced Transit Fare Payment

Description:

Overview: This process shall be responsible for checking to see if the required transit fare payment has already been made. The process shall determine the existence of an advance payment for the transit fare by comparing the received payment information with that in the store containing the list of advanced payments. If the payment has already been made then the process shall remove the requirement for local billing and remove the record of the advanced payment from the store. Details of each payment transaction shall be sent by the process to another process with the payment information received from the transit user removed.

In Subsystem:

Public Transport Management

In Market Package:

APTS4

Data Flows:

Input Flows

advanced_fare_billing
advanced_fare_payment_list
request_roadside_fare_payment
request_vehicle_fare_payment

Output Flows

advanced_fare_transactions
billing_for_fares_needed

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.1;
USR = 3.1.2.4;
USR = 3.1.2.8;
USR = 3.1.4;

7.3.1.5 Bill Transit User for Transit Fare

Description:

Overview: This process shall be responsible for obtaining payment for a transit fare transaction using data provided by the transit user. The process shall achieve this either by requesting that the fare be deducted from the credit being stored by the tag that is acting as the payment instrument for the transit user, or informing the transit user that payment for the fare will be debited to the credit identity provided by the tag. Before sending data to the tag, the process shall check that the transit user's credit identity is not already in the list of bad payers, and if it is request an image of the user which can be forwarded to the appropriate enforcement agency via another process. The tag may be in the form of cash, some type of credit or debit card, an electronic purse, or an intelligent transit ticket upon which pre-payment has been recorded, etc. Details of the transaction shall always be sent by the process to the process that manages transit fare transactions. The process shall load details of advanced transit fare payments into a data store for use by another process when the transit user eventually passes a fare payment point. If requested the process shall provide a copy of the current bad payers list to processes in the transit vehicle fare collection facility for use in on-board payment validation.

In Subsystem:

Public Transport Management

In Market Package:

APTS4

Data Flows:

Input Flows

bad_fare_payment_list
bad_tag_list_request
billing_for_fares_needed
transit_roadside_fare_payment_confirmation
transit_vehicle_fare_payment_confirmation

Output Flows

advanced_fare_payment_list
bad_tag_list_update
confirm_advanced_fares_payment
confirm_roadside_fare_payment
confirm_vehicle_fare_payment
current_fare_transactions
fare_payment_violator_data
get_fare_violator_payment_image
transit_roadside_fare_payment_debited
transit_roadside_fare_payment_request
transit_vehicle_fare_payment_debited
transit_vehicle_fare_payment_request

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.2;
USR = 3.1.2.1;
USR = 3.1.2.3;
USR = 3.1.2.8;

7.3.1.6 Collect Bad Transit Fare Payment Data

Description:

Overview: This process shall be responsible for maintaining a data store containing a list of invalid transit user credit identities. The process shall use this data to check credit identities provided for checking by the billing process. This checking shall ensure that the current transit fare payment transaction is using a credit identity that has not previously had an invalid transaction. Details of possible invalid credit identities shall be sent by the process to the financial institution for verification. The process shall also receive from the financial institution details of invalid payment instrument data that has been found by other means.

In Subsystem:

Public Transport Management

In Market Package:

APTS4

Data Flows:

Input Flows

fare_payment_violator_data
ffi_bad_fare_payment_updates

Output Flows

bad_fare_payment_list
tfi_fare_payment_violator_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.2;
USR = 3.1.2.1;
USR = 3.1.2.3;
USR = 3.1.2.5;

7.3.1.7 Update Transit Fare Data

Description:

Overview: This process shall be responsible for managing the store of data that provides the actual value of transit fares for each segment of each regular transit route. The process shall also act as the interface through which the transit system operator can output and make changes to the stored data, and copies of this data can be provided to the Centralized Payments facility on request. The process shall support inputs from the transit system operator in both manual and audio form, and shall provide its outputs in audible and visual forms. It shall enable the visual output to be either in hardcopy, or as a display. The process shall automatically output the new fares for use by process on-board a transit vehicle and at the roadside, as well as by other ITS functions.

In Subsystem:

Public Transport Management

In Market Package:

APTS4

Data Flows:

Input Flows

ftso_fare_updates
ftso_request_fare_output
transit_fare_data_request
transit_fare_direct_request

Output Flows

transit_fare_data
transit_fare_direct_details
transit_fares_for_advanced_payments
transit_roadside_fare_data
transit_vehicle_fare_data
ttso_transit_fare_output

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.2;
USR = 3.1.2.6;

7.3.2 Distribute Advanced Tolls and Parking Lot Charges

Description:

Overview: This process shall be responsible for receiving requests for advanced payment of transit fares from the toll and parking lot charge collection facilities within the Provide Electronic Payment Services function. It shall pass the advanced fare requests on to another process in the Process Electronic Transit Fare Payment facility, and when received, shall return transit success or failure details to the requesting process. The process shall also receive requests for advanced payment of tolls and parking lot charges from transit vehicle and roadside (transit stop) fare collection facilities. It shall send these requests to other processes in the Provide Electronic Payment Services function and when received, return the results to the requesting process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

ATIS2

Data Flows:

Input Flows

advanced_other_fares_confirm
advanced_tolls_and_charges_roadside_request
advanced_tolls_and_charges_vehicle_request
transfer_charges_to_fares
transfer_tolls_to_fares

Output Flows

advanced_other_fares_request
advanced_tolls_and_charges_roadside_confirm
advanced_tolls_and_charges_vehicle_confirm
transfer_fares_to_charges
transfer_fares_to_tolls

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.2;
USR = 3.1.2.1;
USR = 3.1.4;
USR = 3.1.4.1;
USR = 3.1.4.2;

7.3.3 Get Transit User Image for Violation

Description:

Overview: This process shall be responsible for obtaining an image of a transit user who is trying to carry out an invalid fare payment transaction. The process shall send the image request to other processes either at the roadside, i.e., a transit stop, or on-board a transit vehicle, depending on where the transaction is being attempted. However if the collection method is set to batch, then the process shall take no further action, as an image of the offending transit user will not be available. When the image is received, the process shall use it to form part of the data sent to a process in the Manage Emergency Services function for forwarding to the appropriate enforcement agency.

In Subsystem:

Public Transport Management

In Market Package:

APTS5

Data Flows:

Input Flows

get_fare_violator_payment_image
transit_user_roadside_image
transit_user_vehicle_image

Output Flows

fare_violation_information
request_transit_user_roadside_image
request_transit_user_vehicle_image

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.2;
USR = 3.1.2.5;

7.3.4 Provide Remote Terminal Payment Instrument Interface

Description:

Overview: This process shall be responsible for providing the interface through which payment information can be read from a transit user tag. The process shall support reading this data from transit users at the roadside, e.g., a transit stop, for use in paying the current transit fare and (if required) advanced payments. The process shall support advanced payments for tolls, parking lot charges, and/or transit fares. The process shall collect either the credit identity or the stored credit value data from the tag, and update the stored credit value as a result of the fare and (possibly) advanced charges.

In Subsystem:

Remote Traveler Support

In Market Package:

ATIS7

ATIS8

APTS4

ATIS2

ATIS5

Data Flows:

Input Flows

fpi_confirm_fare_payment_at_roadside
fpi_transit_roadside_tag_data
transit_roadside_fare_payment_debited
transit_roadside_fare_payment_request

Output Flows

tpi_debited_fare_payment_at_roadside
tpi_request_fare_payment_at_roadside
transit_roadside_fare_payment_confirmation
transit_user_roadside_tag_data

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.3;
USR = 3.1.3.1;

7.3.5 Provide Transit Vehicle Payment Instrument Interface

Description:

Overview: This process shall be responsible for providing the interface through which the payment information can be read from a transit user tag. The process shall support the reading of this data from transit users embarking on-board transit vehicles, for use in paying the current transit fare, and if required, advanced payments. The process shall support advanced payments for tolls, and/or parking lot charges, and/or transit fares. It shall be possible for the process to collect either the credit identity or the stored credit value data from the tag, and to update the stored credit value as a result of the fare and (possibly) advanced charges having been paid.

In Subsystem:

Public Transport Vehicle Subsystem

In Market Package:

APTS4

Data Flows:

Input Flows

fpi_confirm_fare_payment_on_transit_vehicle
fpi_transit_vehicle_tag_data
transit_vehicle_fare_payment_debited
transit_vehicle_fare_payment_request

Output Flows

tpi_debited_payment_on_transit_vehicle
tpi_request_fare_payment_on_transit_vehicle
transit_user_vehicle_tag_data
transit_vehicle_fare_payment_confirmation

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.3;
USR = 3.1.3.1;

7.4.1.1 Process Commercial Vehicle Payments

Description:

Overview: This process shall be responsible for transacting payments for electronic credential and tax filing by processes in the Manage Commercial Vehicles function. The payment transaction shall be initiated by processes in the Administer Commercial Vehicles facility which may accept inputs from both the commercial vehicle fleet manager and the commercial vehicle driver acting in the role of fleet manager, i.e., the owner driver. The process shall send the transaction data to the financial institution and report the response back to the requesting process.

In Subsystem:

Commercial Vehicle Administration

In Market Package:

CVO04

Data Flows:

Input Flows

ffi_cv_payment_confirm
financial_request

Output Flows

financial_response
tfi_cv_payment_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.4;
USR = 1.4.0;
USR = 1.4.2;
USR = 1.4.2.1;
USR = 1.4.2.4;
USR = 1.4.3;
USR = 1.4.3.2;
USR = 3.0;
USR = 3.1;
USR = 3.1.4;
USR = 3.1.4.1;

7.4.1.2 Process Yellow Pages Services Provider Payments

Description:

Overview: This process shall be responsible for transacting payments for the registration of other (yellow pages) service providers. The process shall be initiated by receiving data from a process in the Provide Driver and Traveler Services function and shall send the data to the financial institution. The process shall send the response from the financial institution to the requesting process and shall send details of the transaction to another process for entry into a store of transaction records.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

Data Flows:

Input Flows

ffi_registration_payment_confirm
yellow_pages_service_provider_registration_request

Output Flows

tfi_registration_payment_request
yellow_pages_provider_payments_transactions
yellow_pages_service_provider_registration_response

User Service Requirements Mapped to Pspec:

USR = 2.0;
USR = 2.2;
USR = 2.2.1;
USR = 2.2.1.1;
USR = 2.2.1.1.4;
USR = 2.3;
USR = 2.3.4;
USR = 4.0;
USR = 4.1;
USR = 4.1.0;
USR = 4.1.2;
USR = 4.1.2.2;

7.4.1.3 Process Driver Map Update Payments

Description:

Overview: This process shall be responsible for transacting payments from the driver for updates to the navigable map database in the vehicle. The process shall receive the transaction request data from a process in the Provide Driver and Traveler Services function and shall send the data to the financial institution for action. The process shall send the response from the financial institution to the requesting process and shall send details of the transaction to another process for entry into the payment_transaction_records data store.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

ATIS2

Data Flows:

Input Flows

driver_map_update_payment_request
ffi_driver_map_payment_confirm

Output Flows

driver_map_update_payment_response
driver_map_update_payments_transactions
tfi_driver_map_payment_request

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3;
USR = 1.3.0;
USR = 1.3.3;
USR = 1.3.3.2;
USR = 1.3.3.2.1;
USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.4;

7.4.1.4 Process Traveler Map Update Payments

Description:

Overview: This process shall be responsible for transacting payments from the traveler for updates to the navigable map database carried in the personal device. The process shall receive the transaction request data from a process in the Provide Driver and Traveler Services function and shall send the data to the financial institution. The process shall send the response from the financial institution to the requesting process and shall send details of the transaction to another process for entry into the payment_transaction_records data store.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

ATIS2

Data Flows:

Input Flows

ffi_traveler_display_payment_confirm
ffi_traveler_map_payment_confirm
traveler_map_update_payment_request
traveler_personal_display_update_payment_request

Output Flows

tfi_traveler_display_payment_request
tfi_traveler_map_payment_request
traveler_map_update_payment_response
traveler_map_update_payments_transactions
traveler_personal_display_update_payment_response

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3;
USR = 1.3.0;
USR = 1.3.3;
USR = 1.3.3.2;
USR = 1.3.3.2.1;

7.4.1.5 Process Transit User Other Services Payments

Description:

Overview: This process shall be responsible for collecting advance payments for other (yellow pages) services. The transaction data shall be provided by processes in the Manage Transit function in response to reservation requests from a transit user either at the roadside, i.e., a transit stop, or on-board a transit vehicle. The process shall send the received transaction data to the financial institution and shall send the response to the requesting process. It shall also send details of the transaction to another process for entry into a store of transaction records.

In Subsystem:

Public Transport Management

In Market Package:

APTS4

Data Flows:

Input Flows

ffi_other_services_payment_confirm
other_services_roadside_request
other_services_vehicle_request

Output Flows

other_services_roadside_response
other_services_vehicle_response
tfi_other_services_payment_request
transit_user_payments_transactions

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.3;
USR = 1.3.0;
USR = 1.3.3;
USR = 1.3.3.2;
USR = 1.3.3.2.1;
USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.4;

7.4.1.6 Process Traveler Trip and Other Services Payments

Description:

Overview: This process shall be responsible for transacting advanced payments required for the confirmation of a trip by a traveler. Payments supported by the process shall comprise those for any tolls, parking lot charges, transit fares, or other (yellow pages) services that need to be paid for the trip to be confirmed. The process shall receive the transaction data from a process in the Provide Driver and Traveler Services function and shall send the data to the financial institution. Tolls, fares and parking lot charges are sent to the Route Traveler Advanced Payment function for processing. The process shall send the response from the financial institution to the requesting process and shall send details of the transaction to another process for entry into the payment_transaction_records data store.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

ATIS2

Data Flows:

Input Flows

ffi_traveler_other_services_payments_confirm
traveler_advanced_payments_confirm
traveler_other_services_payment_request
traveler_payment_request

Output Flows

tfi_traveler_other_services_payments_request
traveler_advanced_payments_request
traveler_other_services_payment_result
traveler_payment_response
traveler_trip_payments_transactions

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.4;

7.4.1.7 Collect Payment Transaction Records

Description:

Overview: This process shall be responsible for the collection and maintenance of a data store that contains transaction records for payments made for various services provided. The process shall load information into the payment_transaction_records data store for services comprising updates of map databases for drivers and travelers, registration of other (yellow pages) service providers (so that information about what they have to offer is available to travelers and transit users), advanced payment of tolls, parking lot charges, transit fares and other (yellow pages) services that form part of travelers' trips. The data shall be stored by the process with all references to the identity of the payment source, i.e., driver, traveler, commercial vehicle fleet manager, and any other payment information, removed.

In Subsystem:

Information Service Provider

In Market Package:

AD1

AD2

Data Flows:

Input Flows

driver_map_update_payments_transactions
traveler_map_update_payments_transactions
traveler_rideshare_payments_transactions
traveler_trip_payments_transactions
yellow_pages_provider_payments_transactions

Input/Output

payment_transaction_records

Output Flows

traveler_info_payments_transactions

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.4;
USR = 3.1.4.1;

7.4.1.8 Process Traveler Rideshare Payments

Description:

Overview: This process shall be responsible for transacting payments for ridesharing that are required for the confirmation of a traveler's trip. The process shall start the transaction by receiving data from a process in the Provide Driver and Traveler Services function and shall send the data to the appropriate financial institution. The process shall send the response from the financial institution to the requesting process and shall send details of the transaction to another process for entry into a store of transaction records.

In Subsystem:

Information Service Provider

In Market Package:

ATIS8

Data Flows:

Input Flows

ffi_traveler_rideshare_payment_confirm
rideshare_payment_request

Output Flows

rideshare_payment_confirmation
tfi_traveler_rideshare_payment_request
traveler_rideshare_payments_transactions

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.2;

7.4.2 Collect Price Data for ITS Use

Description:

Overview: This process shall be responsible for collecting data about the prices being charged for tolls, parking lots and transit fares. This process shall accept data sent to it by the other processes when they have updated their data and automatically sent it, or this process shall request a transfer of data from the other processes. The process shall load the data into the price_data_for_services data store from which some or all of it can be read on request from processes in other ITS functions.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

ATIS1

ATIS2

ATIS4

Data Flows:

Input Flows

parking_lot_charge_request
parking_lot_price_data
request_prices
toll_price_data
toll_price_request
transit_fare_data
transit_fare_request

Input/Output

price_data_for_services

Output Flows

parking_lot_charge_details
parking_lot_price_data_request
prices
toll_price_data_request
toll_price_details
transit_fare_data_request
transit_fare_details

User Service Requirements Mapped to Pspec:

USR = 1.0;
USR = 1.4;
USR = 1.4.0;
USR = 1.4.2;
USR = 1.4.2.3;
USR = 1.4.3;
USR = 1.4.3.5;
USR = 1.4.3.6;
USR = 3.0;
USR = 3.1;
USR = 3.1.2;
USR = 3.1.2.7;
USR = 3.1.4;
USR = 3.1.4.3;
USR = 3.1.5;
USR = 3.1.5.1;
USR = 3.1.5.1.1;

7.4.3 Route Traveler Advanced Payments

Description:

Overview: This process shall be responsible for receiving a traveler's request for advanced payment (for tolls, parking lot charges, and/or transit fares) and routing it to the appropriate part of the Provide Electronic Payment Services function. The process shall also receive responses to the advanced payment requests and shall return them to the originating process.

In Subsystem:

Information Service Provider

In Market Package:

ATIS7

ATIS2

Data Flows:

Input Flows

advanced_traveler_charges_confirm
advanced_traveler_fares_confirm
advanced_traveler_tolls_confirm
traveler_advanced_payments_request

Output Flows

advanced_traveler_charges_request
advanced_traveler_fares_request
advanced_traveler_tolls_request
traveler_advanced_payments_confirm

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.2;
USR = 3.1.2.4;
USR = 3.1.2.6;
USR = 3.1.3;
USR = 3.1.3.2;
USR = 3.1.4;
USR = 3.1.4;
USR = 3.1.4.1;
USR = 3.1.4.2;

7.5.1 Provide Vehicle Payment Instrument Interface

Description:

Overview: This process shall be responsible for providing the interface through which driver credit identities and stored credit may be entered into the ITS from on-board vehicle tags. The types of vehicles from which data is collected shall include private cars or vans, commercial vehicles, and transit vehicles, including those used for demand responsive transit services. This process shall also provide an interface through which the stored credit held by the tag can be debited for the payment of current or advanced tolls, plus advanced parking lot charges, and/or transit fares.

In Subsystem:

Vehicle

In Market Package:

ATMS10

ATMS16

Data Flows:

Input Flows

cv_driver_enrollment_cost
driver_advanced_payment_for_map
fpi_driver_vehicle_input_credit_identity
fpi_transit_user_vehicle_input_credit_identity
transit_user_advanced_payment_on_vehicle

Output Flows

cv_driver_credit_identity
driver_credit_identity
tpi_debited_driver_payment_at_vehicle
tpi_debited_transit_user_payment_at_vehicle
transit_user_vehicle_credit_identity

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.3;
USR = 3.1.3.1;

7.5.2 Provide Transit User Roadside Payment Instrument Interface

Description:

Overview: This process shall be responsible for providing the interface through which credit identities and stored credit values may be collected from tags being used by transit users. The process shall support the collection of this data at the roadside (which in this instance is a transit stop). Payments by the transit user for fares, other services, payment of advanced tolls, and/or parking lot charges shall be supported by the process. It shall also provide an interface through which stored credit held by the tag can be debited for the same types of payment.

In Subsystem:

Remote Traveler Support

In Market Package:

ATIS7

ATIS8

APTS4

ATIS2

ATIS5

Data Flows:

Input Flows

fpi_transit_user_roadside_input_credit_identity
transit_user_advanced_payment_at_roadside

Output Flows

tpi_debited_transit_user_payment_at_roadside
transit_user_roadside_credit_identity

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.3;
USR = 3.1.3.1;

7.5.3 Provide Personal Payment Instrument Interface

Description:

Overview: This process shall be responsible for providing the interface through which credit identity or stored credit may be collected from the tag being used by a traveler with a personal device. The process shall support the collection of this data from any location in which the device (and hence the tag) is being used. It shall provide an interface through which the credit identity can be used for the payment of advanced tolls, parking lot charges, transit fares, display updates, and/or map updates. The process shall also enable the stored credit value on the tag to be used for the same purposes.

In Subsystem:

Personal Information Access

In Market Package:

ATIS7

ATIS2

ATIS8

Data Flows:

Input Flows

fpi_traveler_personal_input_credit_identity
traveler_personal_display_update_cost
traveler_personal_map_update_cost
traveler_personal_trip_costs

Output Flows

tpi_debited_payment_at_personal_device
traveler_personal_credit_identity

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.3;
USR = 3.1.3.1;

7.5.4 Provide Commercial Fleet Payment Instrument Interface

Description:

Overview: This process shall be responsible for providing the interface through which credit identity or stored credit values may be collected from the tags used by commercial fleet managers. The process shall support the use of the tag data to complete payment for the filing of electronic credentials and tax information that will enable a commercial vehicle to be cleared to travel within the geographic area served by a particular jurisdictional authority. This process shall also enable the stored credit value to be debited as an alternative method of payment.

In Subsystem:

Fleet and Freight Management

In Market Package:

CVO04

CVO05

Data Flows:

Input Flows

cf_manager_enrollment_cost
fpi_commercial_manager_input_credit_identity

Output Flows

cf_manager_credit_identity
tpi_debited_commercial_manager_payment

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.3;
USR = 3.1.3.1;

7.5.5 Provide Traveler Kiosk Payment Instrument Interface

Description:

Overview: This process shall be responsible for providing the interface through which credit identities and stored credit values may be collected from payment instruments being used by travelers. The process shall support the collection of data at the roadside (which in this instance is a kiosk) and use this data for payments needed to confirm a traveler's trip. Payments supported by the process shall include those for advanced tolls, parking lot charges, transit fares, and/or other (yellow pages) services. It shall also provide an interface through which the stored credit held by the tag can be debited for the same types of payment.

In Subsystem:

Remote Traveler Support

In Market Package:

ATIS7

ATIS8

ATIS2

ATIS5

Data Flows:

Input Flows

fpi_traveler_roadside_input_credit_identity
traveler_roadside_trip_costs

Output Flows

tpi_debited_traveler_payment_at_roadside
traveler_roadside_credit_identity

User Service Requirements Mapped to Pspec:

USR = 3.0;
USR = 3.1;
USR = 3.1.0;
USR = 3.1.3;
USR = 3.1.3.1;

8.1 Get Archive Data

Description:

Overview: This process shall collect data from each major function within ITS and external sources for archive purposes that may not exist within current ITS data sources. This process shall respond to requests from the Manage Archive Data Administrator Interface process to import data or data catalogs. This process shall send requests for data or a catalog of available data to the other functions and terminators, either a subscription for data or a one-time request. This process shall receive meta data along with the data to describe the conditions under which the data was collected or any other information about the operational data. When data is received this process shall perform quality checks such as range validation or reformat the data as necessary to meet the archive schema. This process shall execute methods on the incoming data such as cleansing, summarizations, aggregations, or transformations applied to the data before it is stored in the archive. Any changes made to the data shall be recorded in the meta-data stored in the archive to assist in the reconstruction of the original data if possible. This process shall receive inputs from the Manage Archive Data Administrator Interface that contain the parameters for managing the processing on the data. This process forwards the collected onto the Manage Archive function along with updated meta-data and a record of any methods applied to the incoming data. This process shall also support the notification of the operational source functions of any errors that may be present in the data that could be caused by equipment failures or a transmission error.

In Subsystem:

Archived Data Management Subsystem

In Market Package:

AD1

AD2

AD3

AD4

Data Flows:

Input Flows

collected_roadside_data
cv_archive_data
em_archive_data
emissions_archive_data
fifd_intermodal_archive_data
fmtsp_multimodal_archive_data
fmup_map_archive_data
fods_other_data_source_archive_data
fws_weather_archive_data
import_administration_request
maintenance_archive_data
parking_archive_data
toll_archive_data
traffic_management_archive_data
transit_archive_data
traveler_archive_data

Output Flows

collected_roadside_data_status
cv_archive_request
cv_archive_status
em_archive_request
em_archive_status
emissions_archive_request
emissions_archive_status
import_administration_status
maintenance_archive_request

maintenance_archive_status
parking_archive_request
parking_archive_status
retrieved_archive_data
tifd_intermodal_archive_request
tifd_intermodal_archive_status
tmtsp_multimodal_archive_request
tmtsp_multimodal_archive_status
tmup_map_archive_request
tmup_map_archive_status
tods_other_data_source_archive_request
tods_other_data_source_archive_status
toll_archive_request
toll_archive_status
traffic_management_archive_request
traffic_management_archive_status
transit_archive_request
transit_archive_status
traveler_archive_request
traveler_archive_status
tws_weather_archive_request
tws_weather_archive_status

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.1;
USR = 7.1.1.1;
USR = 7.1.1.3;
USR = 7.1.2;
USR = 7.1.2.1;
USR = 7.1.2.1.1;
USR = 7.1.2.1.2;
USR = 7.1.2.1.3;
USR = 7.1.2.1.3(a);
USR = 7.1.2.1.3(b);
USR = 7.1.2.1.3(c);
USR = 7.1.2.1.4;
USR = 7.1.2.1.5;
USR = 7.1.2.1.5(a);
USR = 7.1.2.1.5(b);
USR = 7.1.2.2;
USR = 7.1.2.3;
USR = 7.1.2.4;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.1;
USR = 7.1.3.1.1(a);
USR = 7.1.3.1.1(b);
USR = 7.1.3.1.1(c);
USR = 7.1.3.1.1(d);
USR = 7.1.3.1.1(e);
USR = 7.1.3.1.10;
USR = 7.1.3.1.11;
USR = 7.1.3.1.2;
USR = 7.1.3.1.3;
USR = 7.1.3.1.3(a);
USR = 7.1.3.1.3(b);
USR = 7.1.3.1.3(c);
USR = 7.1.3.1.3(d);
USR = 7.1.3.1.3(e);

USR = 7.1.3.1.4;
USR = 7.1.3.1.4(a);
USR = 7.1.3.1.4(b);
USR = 7.1.3.1.4(c);
USR = 7.1.3.1.4(d);
USR = 7.1.3.1.4(e);
USR = 7.1.3.1.4(f);
USR = 7.1.3.1.4(g);
USR = 7.1.3.1.5;
USR = 7.1.3.1.5(a);
USR = 7.1.3.1.5(b);
USR = 7.1.3.1.5(c);
USR = 7.1.3.1.5(d);
USR = 7.1.3.1.5(e);
USR = 7.1.3.1.5(f);
USR = 7.1.3.1.5(g);
USR = 7.1.3.1.5(h);
USR = 7.1.3.1.6;
USR = 7.1.3.1.6(a);
USR = 7.1.3.1.6(b);
USR = 7.1.3.1.6(c);
USR = 7.1.3.1.6(d);
USR = 7.1.3.1.6(e);
USR = 7.1.3.1.6(f);
USR = 7.1.3.1.7;
USR = 7.1.3.1.7(a);
USR = 7.1.3.1.7(b);
USR = 7.1.3.1.8;
USR = 7.1.3.1.8(a);
USR = 7.1.3.1.8(b);
USR = 7.1.3.1.8(c);
USR = 7.1.3.1.8(d);
USR = 7.1.3.1.8(e);
USR = 7.1.3.1.8(f);
USR = 7.1.3.1.8(g);
USR = 7.1.3.1.8(h);
USR = 7.1.3.1.9;
USR = 7.1.3.1.9(a);
USR = 7.1.3.1.9(b);
USR = 7.1.3.1.9(c);
USR = 7.1.3.1.9(d);
USR = 7.1.3.1.9(e);
USR = 7.1.3.1.9(f);
USR = 7.1.3.2;
USR = 7.1.3.3;
USR = 7.1.3.4;
USR = 7.1.3.5;
USR = 7.1.3.5.1;
USR = 7.1.3.5.2;
USR = 7.1.3.6;
USR = 7.1.3.7;
USR = 7.1.3.8;
USR = 7.1.3.9;
USR = 7.1.4;
USR = 7.1.4.1;
USR = 7.1.4.1.1;
USR = 7.1.4.1.2;
USR = 7.1.4.1.3;
USR = 7.1.4.2;
USR = 7.1.4.2(c);
USR = 7.1.4.4;
USR = 7.1.4.4(a);
USR = 7.1.4.4(b);

USR = 7.1.4.4(c);
USR = 7.1.4.5;

8.2 Manage Archive

Description:

Overview: This process shall store the collected and formatted data in a permanent archive data store. This process shall receive the formatted data from the Get Archive Data function accompanied by any updates to the meta data that would describe the formatting operations performed on the data as it was imported. This process shall respond to requests from the administrator interface function to maintain the schema of the archive data, set update frequencies, backup schedules, user authentication schemes, cleansing algorithms. This process shall provide the administrator interface function with status of the data quality in the archive, frequency reports on use of the archive, updates to the measure of the volume of the data and other data archive metrics. This process shall receive inputs from the Coordinate Archives function to provide data and information about the archive schema to other archives. In turn the process shall receive data and schema of other archives to use to build a global schema. The process shall use the global schema to support requests from user systems for data that may be spread across multiple archives. The process shall maintain the access privileges information for the data held in the archive to maintain the security of the archive. The process shall employ such techniques as necessary to maintain the integrity of the data and ensure no data is lost from the archive. The process shall respond to requests for data to support user data products, user analysis, and inputs to government reporting systems. The process shall respond to such request by authenticating the originator of the request and providing the data that is available. The process shall also be capable of providing a sample or catalog of data contained within the archive to support the user requests.

In Subsystem:

Archived Data Management Subsystem

In Market Package:

AD1

AD2

AD3

AD4

Data Flows:

Input Flows

analyze_archive_data_request
archive_administration_request
archive_data_product_request
global_schema
government_report_data_request
other_archive_data_input
other_archive_data_request_input
retrieved_archive_data

Input/Output

archive_data

Output Flows

archive_administration_data
archive_data_for_analysis
archive_data_product
government_report_data
local_schema
other_archive_data
other_archive_data_request

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.1;
USR = 7.1.1.1;
USR = 7.1.1.2;
USR = 7.1.1.3;
USR = 7.1.1.4;
USR = 7.1.1.4.1;
USR = 7.1.1.4.2;
USR = 7.1.1.4.3;
USR = 7.1.1.4.4;
USR = 7.1.2.3;
USR = 7.1.4;
USR = 7.1.4.1;
USR = 7.1.4.1.1;
USR = 7.1.4.1.2;
USR = 7.1.4.1.3;
USR = 7.1.4.2;
USR = 7.1.4.2(a);
USR = 7.1.4.2(b);
USR = 7.1.4.2(c);
USR = 7.1.4.2(d);
USR = 7.1.4.3;
USR = 7.1.4.4;
USR = 7.1.4.5;
USR = 7.1.5;
USR = 7.1.5.1;
USR = 7.1.5.1(a);
USR = 7.1.5.1(b);
USR = 7.1.5.1(c);
USR = 7.1.5.1(d);
USR = 7.1.5.2;
USR = 7.1.5.2.3;
USR = 7.1.5.2.4;
USR = 7.1.6;
USR = 7.1.6.2;
USR = 7.1.6.2.1;
USR = 7.1.6.2.2;
USR = 7.1.6.3;
USR = 7.1.6.3.1;
USR = 7.1.6.3.2;

8.3 Manage Archive Data Administrator Interface

Description:

Overview: This process shall interface with the Archive Data Administrator terminator and receive inputs from the administrator concerning the management and administration of the archive. The process shall establish user authentication controls for the archive and send the information to the Manage Archive function. The process shall maintain the schema of the archive, including the data and meta data contained within the archive data. Updates to the schema shall be distributed to the Manage Archive function as well as the Get Archive Data function. The process shall send the parameters and requests to the Get Archive Data function to control what data is imported into the archive and how the data is to be formatted when it is received. The parameters sent shall include such things as the schema, data format, methods to apply to the data, cleansing parameters, quality metrics, and checking specifications. The process shall send requests to the Get Archive Data function for new data or a catalog of data that may be available. The process shall respond to requests from the Manage On Demand Archive Requests function by making requests of the Get Archive Data function to establish the source and identity of the data that may exist in ITS or non-ITS sources. Then the process shall respond to the user request with the confirmation that the request can be satisfied and specifications about the data once it is imported. In cases where the Manage Archive function will be managing a roadside data collection function, this process shall initiate and control the function by sending commands and requests to the Manage Roadside Data Collection function. This process receives the status from the other functions within Manage Archived Data and presents them to the administrator.

In Subsystem:

Archived Data Management Subsystem

In Market Package:

AD1

AD2

AD3

AD4

Data Flows:

Input Flows

archive_administration_data
collection_administration_status
fada_archive_administration_requests
import_administration_status
on_demand_archive_request

Output Flows

archive_administration_request
archive_request_confirmation
collection_administration_request
import_administration_request
tada_archive_administration_data

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.1.4;
USR = 7.1.1.4.1;
USR = 7.1.1.4.2;
USR = 7.1.1.4.3;
USR = 7.1.1.4.4;
USR = 7.1.2;
USR = 7.1.2.1;
USR = 7.1.2.1.1;
USR = 7.1.2.1.2;

USR = 7.1.2.1.4;
USR = 7.1.2.1.5;
USR = 7.1.2.1.5(a);
USR = 7.1.2.1.5(b);
USR = 7.1.2.2;
USR = 7.1.2.4;
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.2;
USR = 7.1.3.3;
USR = 7.1.3.4;
USR = 7.1.3.5;
USR = 7.1.3.5.1;
USR = 7.1.3.5.2;
USR = 7.1.3.6;
USR = 7.1.3.7;
USR = 7.1.3.8;
USR = 7.1.4.1;
USR = 7.1.4.1.1;
USR = 7.1.4.1.2;
USR = 7.1.4.1.3;
USR = 7.1.4.2;
USR = 7.1.4.2(a);
USR = 7.1.4.2(b);
USR = 7.1.4.2(c);
USR = 7.1.4.2(d);
USR = 7.1.4.4;
USR = 7.1.4.4(a);
USR = 7.1.4.4(b);
USR = 7.1.4.4(c);
USR = 7.1.4.5;
USR = 7.1.5.3;
USR = 7.1.6;
USR = 7.1.6.1;
USR = 7.1.6.1.1;
USR = 7.1.6.2;
USR = 7.1.6.2.1;
USR = 7.1.6.2.2;
USR = 7.1.6.3;
USR = 7.1.6.3.1;

8.4 Coordinate Archives

Description:

Overview: This process shall coordinate the information exchange between different Manage Archived Data functions represented through the Other Archives terminator. This process shall allow other archives to share data collected by other archive functions to share the data in response to local requests from users systems. This process shall use data collected from different archives to build a set of global schema which the data archive definitions for the local archive plus any archives known to the local archive. This process shall provide the global schema to the local Manage Archive function. This process shall receive the schema of the local archive to share with other archive functions. This process shall provide data to those other archives when requested. This process shall support analysis, data fusion, and data mining of archived information across geographically dispersed archives.

In Subsystem:

Archived Data Management Subsystem

In Market Package:

AD3

AD4

Data Flows:

Input Flows

foa_archive_coordination_data
local_schema
other_archive_data
other_archive_data_request

Output Flows

global_schema
other_archive_data_input
other_archive_data_request_input
toa_archive_coordination_data

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.4.4;
USR = 7.1.5;
USR = 7.1.5.1;
USR = 7.1.5.1(a);
USR = 7.1.5.1(b);
USR = 7.1.5.1(c);
USR = 7.1.5.1(d);
USR = 7.1.5.2;
USR = 7.1.5.2.3;
USR = 7.1.5.2.4;
USR = 7.1.6;
USR = 7.1.6.2;
USR = 7.1.6.2.2;
USR = 7.1.6.3;
USR = 7.1.6.3.1;

8.5 Process Archived Data User System Requests

Description:

Overview: This process shall monitor the archive data user systems interface for requests for data from the archive. This process shall support requests from users involved in planning, research, safety, as well as operations of transportation functions. This process shall receive requests for data and catalogs of data that may be contained in the archive. This process shall translate the requests into a format that can be understood by the Manage Archive function to retrieve data from the archive. When data or a catalog of data is received from the archive, this process shall generate the requested data product for the users systems. For archive data requiring financial payment this archive process the financial requests and manages an interface to a Finanicial Institution.

In Subsystem:

Archived Data Management Subsystem

In Market Package:

AD1

AD2

AD3

AD4

Data Flows:

Input Flows

archive_data_product
fadu_archive_data_product_request
ffi_archive_payment_confirm

Output Flows

archive_data_product_request
tadu_archive_data_product
tfi_archive_payment_request

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.1.4.1;
USR = 7.1.1.4.3;
USR = 7.1.1.4.4;
USR = 7.1.2;
USR = 7.1.2.5;
USR = 7.1.3.7;
USR = 7.1.4.4;
USR = 7.1.4.4(c);
USR = 7.1.5;
USR = 7.1.5.1;
USR = 7.1.5.1(a);
USR = 7.1.5.1(b);
USR = 7.1.5.1(c);
USR = 7.1.5.1(d);
USR = 7.1.5.2;
USR = 7.1.5.2.2;
USR = 7.1.5.2.3;
USR = 7.1.5.2.4;
USR = 7.1.6;
USR = 7.1.6.1;
USR = 7.1.6.1.1;
USR = 7.1.6.2;

USR = 7.1.6.2.2;
USR = 7.1.6.3;
USR = 7.1.6.3.1;
USR = 7.1.6.3.3;
USR = 7.1.6.4;
USR = 7.1.6.4.1;
USR = 7.1.6.4.1(a);
USR = 7.1.6.4.1(b);
USR = 7.1.6.4.1(c);
USR = 7.1.6.4.1(d);
USR = 7.1.6.4.1(e);
USR = 7.1.6.4.1(f);
USR = 7.1.6.4.1(g);
USR = 7.1.6.4.2;
USR = 7.1.6.4.2(a);
USR = 7.1.6.4.2(b);
USR = 7.1.6.4.2(c);
USR = 7.1.6.4.2(d);
USR = 7.1.6.4.3;
USR = 7.1.6.4.3(a);
USR = 7.1.6.4.3(b);
USR = 7.1.6.4.3(c);

8.6 Analyze Archive

Description:

Overview: This process shall support the interface with Archive Data User Systems for requests for analysis of the archive data. This process shall support analysis products that can provide users with the ability to perform activities such as data mining, data fusion, summarizations, aggregations, and recreation from archive data. This process shall receive the users systems requests and develop the request that the Manage Archive function can process to retrieve the data from the archive. This process shall be able to respond to users systems requests for a catalog of the analysis products available. When data and meta data are returned from the archive and the analysis is performed this process shall produce the output for the Archive Data User Systems terminator. For archive data requiring financial payment this archive process the financial requests and manages an interface to a Finanicial Institution.

In Subsystem:

Archived Data Management Subsystem

In Market Package:

AD2

AD4

Data Flows:

Input Flows

archive_data_for_analysis
fadu_archive_analysis_request
ffi_archive_analysis_payment_confirm

Output Flows

analyze_archive_data_request
tadu_archive_analysis_results
tfi_archive_analysis_payment_request

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.1.4.1;
USR = 7.1.1.4.3;
USR = 7.1.1.4.4;
USR = 7.1.1.5;
USR = 7.1.2;
USR = 7.1.2.6;
USR = 7.1.3.7;
USR = 7.1.4.4;
USR = 7.1.5;
USR = 7.1.5.1;
USR = 7.1.5.1(a);
USR = 7.1.5.1(b);
USR = 7.1.5.1(c);
USR = 7.1.5.1(d);
USR = 7.1.5.2;
USR = 7.1.5.2.1;
USR = 7.1.5.2.1(a);
USR = 7.1.5.2.1(b);
USR = 7.1.5.2.1(c);
USR = 7.1.5.2.1(d);
USR = 7.1.5.2.3;
USR = 7.1.5.2.4;
USR = 7.1.6;
USR = 7.1.6.1;
USR = 7.1.6.1.1;

USR = 7.1.6.2;
USR = 7.1.6.2.2;
USR = 7.1.6.3;
USR = 7.1.6.3.1;
USR = 7.1.6.3.2;
USR = 7.1.6.3.3;
USR = 7.1.6.4;
USR = 7.1.6.4.1;
USR = 7.1.6.4.1(a);
USR = 7.1.6.4.1(b);
USR = 7.1.6.4.1(c);
USR = 7.1.6.4.1(d);
USR = 7.1.6.4.1(e);
USR = 7.1.6.4.1(f);
USR = 7.1.6.4.1(g);
USR = 7.1.6.4.2;
USR = 7.1.6.4.2(a);
USR = 7.1.6.4.2(b);
USR = 7.1.6.4.2(c);
USR = 7.1.6.4.2(d);
USR = 7.1.6.4.3;
USR = 7.1.6.4.3(a);
USR = 7.1.6.4.3(b);
USR = 7.1.6.4.3(c);

8.7 Process On Demand Archive Requests

Description:

Overview: This process shall receive requests for data to be imported into the archive that is not already in the archive. The process shall forward the request to the Manage Archive Data Administrator Interface function for the administrator to handle the user request. The process shall receive the response from the administrator and forward the information to the Archive Data User System.

In Subsystem:

Archived Data Management Subsystem

In Market Package:

AD1

AD2

AD3

AD4

Data Flows:

Input Flows

archive_request_confirmation
fadu_on_demand_archive_request

Output Flows

on_demand_archive_request
tadu_on_demand_confirmation

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.2;
USR = 7.1.2.2;
USR = 7.1.4.4;
USR = 7.1.4.4(a);
USR = 7.1.4.4(b);
USR = 7.1.4.4(c);
USR = 7.1.6;
USR = 7.1.6.1;
USR = 7.1.6.1.1;
USR = 7.1.6.2;
USR = 7.1.6.2.2;
USR = 7.1.6.3;
USR = 7.1.6.3.1;
USR = 7.1.6.4;
USR = 7.1.6.4.1;
USR = 7.1.6.4.1(a);
USR = 7.1.6.4.1(b);
USR = 7.1.6.4.1(c);
USR = 7.1.6.4.1(d);
USR = 7.1.6.4.1(e);
USR = 7.1.6.4.1(f);
USR = 7.1.6.4.1(g);
USR = 7.1.6.4.2;
USR = 7.1.6.4.2(a);
USR = 7.1.6.4.2(b);
USR = 7.1.6.4.2(c);
USR = 7.1.6.4.2(d);

USR = 7.1.6.4.3;
USR = 7.1.6.4.3(a);
USR = 7.1.6.4.3(b);
USR = 7.1.6.4.3(c);

8.8 Prepare Government Reporting Inputs

Description:

Overview: This process shall support the preparation of inputs to reporting systems of the federal or state governments that require data from the ITS archive. This process shall respond to requests from the Government Reporting Systems terminator for data from the archive and generate the request in a form understood by the Manage Archive function. The data and any meta data necessary shall be returned from the Manage Archive function. This process shall receive the data and format it as requested and send it to the Government Reporting Systems terminator where it may be combined with other data before final submission.

In Subsystem:

Archived Data Management Subsystem

In Market Package:

AD1

AD2

AD4

Data Flows:

Input Flows

fgrs_government_data_report_request
government_report_data

Output Flows

government_report_data_request
tgrs_government_data_report_input

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.5;
USR = 7.1.5.2;
USR = 7.1.5.2.5;
USR = 7.1.5.2.5(a);
USR = 7.1.5.2.5(b);
USR = 7.1.5.2.5(c);
USR = 7.1.5.2.5(d);
USR = 7.1.5.2.5(e);
USR = 7.1.5.2.5(f);
USR = 7.1.5.2.5(g);
USR = 7.1.5.2.5(h);
USR = 7.1.5.2.5(i);
USR = 7.1.5.2.5(j);
USR = 7.1.6;
USR = 7.1.6.2;
USR = 7.1.6.2.2;
USR = 7.1.6.3;
USR = 7.1.6.3.1;

8.9 Manage Roadside Data Collection

Description:

Overview: This process shall manage the collection of archive data directly from collection equipment located at the roadside. This process shall collect traffic information as well as environmental or other information that may be collected by roadside devices. This process shall respond to requests from the Manage Archive Data Administer Interface process to input the parameters that control the collection process. The request for data and control parameters shall be sent to the Manage Traffic function where the information is collected and returned. This process shall forward the data onto the Get Archive Data function for import into the archive. The Get Archive Data function shall be able to return status about the imported data. This process shall use the status information to adjust the collection function and report back to the administrator function.

In Subsystem:

Archived Data Management Subsystem

In Market Package:

AD1

AD2

AD4

Data Flows:

Input Flows

collected_roadside_data_status
collection_administration_request
roadside_archive_data

Output Flows

collected_roadside_data
collection_administration_status
roadside_archive_control

User Service Requirements Mapped to Pspec:

USR = 7.0;
USR = 7.1;
USR = 7.1.0;
USR = 7.1.2;
USR = 7.1.2.1;
USR = 7.1.2.1.1;
USR = 7.1.2.1.2;
USR = 7.1.2.1.3;
USR = 7.1.2.1.3(a);
USR = 7.1.2.1.3(b);
USR = 7.1.2.1.3(c);
USR = 7.1.3;
USR = 7.1.3.1;
USR = 7.1.3.1.1;
USR = 7.1.3.1.1(a);
USR = 7.1.3.1.1(c);
USR = 7.1.3.1.3;
USR = 7.1.3.1.3(e);
USR = 7.1.3.1.7;
USR = 7.1.3.1.7(a);

9.1 Detect and Classify Speed Violations

Description:

Overview: This process is responsible for detecting and classifying speed violations at the roadside using a combination of a speed sensor and a vehicle identification device. This process is governed by parameters regarding speed thresholds to be considered as violations.

Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Market Package:

AD4

ATMS27

Data Flows:

Input Flows

ft_traffic_data
ft_traffic_images
fvc_vehicle_characteristics
local_speed_violation_data
posted_speed_limit
speed_violation_parameters

Output Flows

local_speed_violation_data
speed_violation_detection_data
speed_violation_video_image

User Service Requirements Mapped to Pspec:

USR = C2.6.2.2;
USR = C2.6.2.3;
USR = C2.6.6.6;
USR = C2.6.6.7;

9.2 Collect and Verify Speed Violations

Description:

Overview: This process collects data from the Detect and Classify Speed Violations and verifies that they are in fact violations. The data collected by this process includes video or other data used to identify individual vehicles as well as telemetry data from the field such as speed and time of day. Once this process has verified the violations it forwards the data to the Process Traffic Violations function. This process can be run as an intermmediate step before issuing tickets for speed violations.

Functional Requirements: none.

In Subsystem:

Traffic Management

In Market Package:

AD4

ATMS27

Data Flows:

Input Flows

speed_violation_detection_data
speed_violation_video_image

Output Flows

verified_speed_violation_data

User Service Requirements Mapped to Pspec:

USR = C2.6.2.6;
USR = C2.6.2.7;

9.3 Post Speed Limit

Description:

Overview: This process posts safe speed limits to passing motorists through use of such means as variable speed limit signs, invehicle signing functions, highway advisory radio broadcasts. These safe speed limits are defined by the Set Speed Limit function but they may also gather input from the local roadway conditions to determine the safe speed based on weather or other environmental conditions from the Process Roadway Warning Data function. The safe speed is also output to the function to Detect and Classify Speed Violations to use to enforce the new speed limit.

Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Market Package:

ATMS27

Data Flows:

Input Flows

roadway_warning_speed_conditions
traffic_speed_limit_setting

Output Flows

posted_speed_limit
td_driver_variable_speed_limit
vehicle_variable_speed_limit

User Service Requirements Mapped to Pspec:

USR = C2.6.2.1;
USR = C2.6.2.2;
USR = C2.6.2.3;
USR = C2.6.2.4;
USR = C2.6.2.5;
USR = C2.6.2.8;

9.4 Set Speed Limit

Description:

Overview: This process determines the safe speed limit for a given segment of the road network based on the roadway characteristics such as the geometry of the roadway, the current conditions of the roadway including recent weather conditions that may be provided by the Provide Traffic Operations Personnel Traffic Data Interface function. This process maintains a log of all of the speed limit settings including when changes were made to the speed limit on a given roadway segment. This information can be used to support the verification of speed violations.

Functional Requirements: none.

In Subsystem:

Traffic Management

In Market Package:

ATMS27

Data Flows:

Input Flows

roadway_speed_configuration
static_roadway_characteristics
variable_speed_limit_link_data

Output Flows

current_link_speed_limit
traffic_speed_limit_setting
variable_speed_limit_link_data

User Service Requirements Mapped to Pspec:

USR = C2.6.2.1;
USR = C2.6.2.2;
USR = C2.6.2.3;
USR = C2.6.2.4;
USR = C2.6.2.5;
USR = C2.6.2.8;

9.5 Establish Violation Parameters

Description:

Overview: This process establishes the parameters by which violations of speed limits and signals are measured. The inputs to this process from the Provide Traffic Operations Personnel Traffic Data Interface function and the Set Speed Limit function allow the things that constitute a violation to be varied based on road, weather, or traffic conditions. This process sends its output to the functions that detect and classify speed and signal violations.

Functional Requirements: none.

In Subsystem:

Traffic Management

In Market Package:

AD4

ATMS27

ATMS28

Data Flows:

Input Flows

current_link_speed_limit
roadway_violation_parameters

Output Flows

signal_violation_parameters
speed_violation_parameters

User Service Requirements Mapped to Pspec:

USR = C2.6.2.8;
USR = C2.6.3.4;

9.6 Detect and Classify Signal Violations

Description:

Overview: This process is responsible for detecting and classifying signal violations at the roadside using a combination of a presence sensor and a vehicle identification device. This process is governed by parameters that dictate the conditions to be considered as violations. Data on violations is collected and can be stored for retrieval in a stand-alone device or communicated to the Collect and Verify Signal Violations.

Functional Requirements: none.

In Subsystem:

Roadway Subsystem

In Market Package:

AD4

ATMS28

Data Flows:

Input Flows

ft_traffic_data
ft_traffic_images
fvc_vehicle_characteristics
local_signal_violation_data
signal_indication_for_enforcement
signal_violation_parameters

Output Flows

local_signal_violation_data
signal_violation_detection_data
signal_violation_video_data

User Service Requirements Mapped to Pspec:

USR = C2.6.3.1;
USR = C2.6.3.2;

9.7 Collect and Verify Signal Violations

Description:

Overview: This process collects data from the Detect and Classify Signal Violations and verifies that they are in fact violations. The data collected by this process includes video or other data used to identify individual vehicles as well as telemetry data from the field such as loop data and time of day. Once this process has verified the violations it forwards the data to the Process Traffic Violations function. This process can be run as an intermmediate step before issuing tickets for signal violations.

Functional Requirements: none.

In Subsystem:

Traffic Management

In Market Package:

AD4

ATMS28

Data Flows:

Input Flows

signal_violation_detection_data
signal_violation_video_data

Output Flows

verified_signal_violation_data

User Service Requirements Mapped to Pspec:

USR = C2.6.3.2;
USR = C2.6.3.3;
USR = C2.6.3.4;