

Logical Architecture Volume II

Process Specifications

Preface

Volume II of the Logical Architecture contract deliverable document documents the Process Specifications. This formatted version of the Teamwork model data is mechanically produced from the Teamwork CDIF (Case Data Interchange Format) output file. It is an accurate reproduction of the actual model, but reformatted for easier use. The following discussion describes the process steps and the resulting document structure.

Document Generation

The Teamwork process database is a formal, robust repository of all data flow diagrams and

P-Specs in the CASE process model of the ITS National Architecture. It also contains certain message frequency assumptions as defined by the Joint Development Team during the analysis process. It is not however easily turned into a user friendly document. To produce the following documentation several steps are required, each modifying the form, but not the content, of the teamwork database.

Teamwork Export:

Data is extracted from the model using a Teamwork utility function (twk_get). This utility produces an ASCII flat file (ITS_cdif) in the Teamwork AIX environment. This file is very large (5.5 MB) and completely describes the CASE model, including the geometry of its graphical artifacts, the Data Flow diagrams (DFDs).

REXX Parse:

Next a special program is used to automatically parse this large file into a set of smaller, more specific, ASCII files to facilitate importation into a relational database for analysis and documentation. The parsing program is written in REXX and runs in an OS/2 environment. In addition to finding, extracting and building the various ASCII files, this program performs some character substitution. Certain characters that are used in the Teamwork model cannot be directly carried over into Microsoft Excel models used in data loading analysis. For example, “*” in Teamwork is a comment delimiter while in Excel it is a multiplication operator. Also, some carriage return symbols are removed from the Teamwork data to eliminate “hard line breaks” and allow better text flow when reformatted in Access or Microsoft Word. With these exceptions, the data in the ASCII files is exactly the data in the Teamwork CDIF file and is an accurate representation of the CASE model.

Microsoft Access Import:

The ASCII flat files are then automatically imported into data tables in the Logical Architecture database (logical.mdb) using Access Visual Basic code and Macros.

Microsoft Access (Formatted) Report:

Finally, the P-Specs are generated using Access report utility and the data imported from Teamwork.

P-Spec Content

The top of each page includes when the CASE model was last changed, when the CDIF was extracted, and when the logical database was updated. Each Process Specification (P-Spec) is listed in numerical order and each P-Spec consists of the following components.

P-Spec Number and Name

The actual Teamwork P-Spec number and full name, in bold italicized type.

P-Spec Description

The P-Spec descriptions have been divided into two portions. The first paragraph is always an “overview” of the purpose and basic plan of the P-Spec. In many cases this is all that is required to understand the purpose of the process.

The text following the first paragraph is meant to provide more detailed specifics in exactly what this process needs, and what information flows this process will generate.

Extended Attributes

Teamwork provides the capability to add additional data elements to the P-Spec that augment the utility of the CASE model, but are not constrained by the rigor of Teamwork’s checking mechanisms. Providing that users implement their own reasonable checks and balances, these can be very useful features. The Joint Development Team uses extended attributes to provide automated traceability and “hooks” for data loading studies.

User Service Requirements

Each P-Spec is provided to meet a specific need (or needs), and the User Service Requirements that drive that need are listed here as extended attributes. It should be noted that a low level USR automatically implies that its higher level “parents” are also being served by this P-Spec. That is, if a P-Spec is needed to satisfy requirement 3.1, then it must also be needed to satisfy requirement 3. This section of the P-Spec is key to automated traceability of requirements to

P-Specs and vice versa.

Output Flow Dynamics Assumption(s)

The logical Architecture also uses P-Spec extended attributes to document output flow rate assumptions (in messages/second), primarily for data loading analysis.

Dynamic Expressions

The P-Spec output message rates are estimated by the analyst and stated in the form

output_data_flow_name = nnnn

where nnnn can be either a numeric value or an Excel compatible expression. Where expressions refer to another flow name, they are referencing the value of that flow’s message rate. These expressions are extracted and evaluated (i.e., converted to numeric values) during data loading analysis through the use of an Excel spreadsheet model linked to the Access database.

Parameters

Many of the Output Flow Dynamics Assumption(s) are parameterized. Parameters are used to connect assumptions made independent of the CASE model (i.e., outside Teamwork) with CASE definitions within the model. They are evaluated during data loading analysis using an Excel spreadsheet model (pmodel.xls). Parameters appear in the dynamic expressions in ALL CAPITAL typeface. They are defined in the Logical Architecture database. Teamwork does not check them. A complete list of Parameters used (and their definitions) is contained in Volume I of the Logical Architecture.

PROCESS SPECIFICATIONS

ITS National Architecture**Document Control Information***CASE Model: 06/08/96 @ 12:34:24**Extracted: 06/08/96 @ 13:10:33**Logical Database: 06/09/96 @ 15:06:22*

PROCESS SPECIFICATIONS

Page 2 of 475

1.1.1 Process Traffic Sensor Data**Input Flows**

fmmc_crossing_close_duration
fmmc_crossing_close_time
fp_pedestrian_data
fp_pedestrian_images
fre_physical_conditions
fre_tunnel_conditions
ft_traffic_data
ft_traffic_images
traffic_video_camera_control

Output Flows

hov_lane_data_input
incident_analysis_data
local_sensor_data_for_highways
local_sensor_data_for_roads
traffic_sensor_data

Description:

Overview: This process shall be responsible for collecting all of the data obtained from vehicles, pedestrians (travelers using other modes of transport) and multimodal crossings. The process shall be able to receive both passive (e.g. presence) and active (e.g. tag) data from vehicles. Where any of the data is provided in analog form, the process shall be responsible for converting it into digital form. The converted data shall be sent to other processes for distribution, further analysis and storage.

Solicited Input Processing: This process shall receive the following data flows as a result of its regular scan of data input sources:

- (a)'fmmc-crossing_close_duration';
- (b)'fmmc-crossing_close_time';
- (c)'fp-pedestrian_data';
- (d)'fp-pedestrian_images';
- (e)'fre-physical_conditions';
- (f)'fre-tunnel_conditions';
- (g)'ft-traffic_data';
- (h)'ft-traffic_images'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'incident_analysis_data';
- (b)'local_sensor_data_for_highways';
- (c)'local_sensor_data_for_highways';
- (d)'traffic_sensor_data'
- (e)'hov_lane_data_input'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor the solicited data flows shown above by running several times (e.g. four) during each second;
- (b) where necessary convert the data obtained in (a) from analog to digital form;
- (c) once per second send all of the data to other processes in the Manage Traffic function via the solicited output data flows shown above;

(d) complete a full scan of all inputs and generate the outputs in less than the time interval between successive activations, e.g. in not more than one quarter of a second (1/25 sec), regardless of the number of inputs to be scanned and outputs to be generated.

PROCESS SPECIFICATIONS

Page 3 of 475

PROCESS SPECIFICATIONS

Page 4 of 475

1.1.2.1 Process Traffic Data for Storage

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_tags
parking_lot_current_state
predicted_incident_data
processed_data
selected_strategy
vehicle_smart_probe_data_for_storage
wide_area_pollution_data

Output Flows

current_data
long_term_data

Description:

Overview: This process shall be responsible for storing data sent by other processes into the long term and current data stores. The data shall comprise the results of processing analog data by sensors in another process, that being sent to indicators (intersection controllers, pedestrian controllers, variable message signs, ramp metering equipment, etc.), parking lots and other street equipment, the responses that are being received from the indicators, plus details of current traffic conditions, predicted incidents, current incidents, parking lot states, ramp states, link journey times and road conditions provided by vehicle probes, and the selected traffic control strategy. The data stored by the process in the current data store shall show the values collected over the last five minute period, starting at zero minutes past the hour, and shall be updated each and every five minutes. The data in the long term data store shall show the data collected over the last hour, and for each hour that has passed in the current day and for the each data in a rolling two week period.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

(a)'current_incident_data';
(b)'current_lot_status';
(c)'current_ramp_state';
(d)'current_road_network_data';
(e)'current_highway_network_data';
(f)'current_road_network_use';
(g)'hov_lane_data';
(h)'indicator_control_storage_data_for_highways';
(i)'indicator_control_storage_data_for_roads';
(j)'indicator_input_storage_data_for_highways';
(k)'indicator_input_storage_data_for_roads';
(l)'link_data_from_tags';
(m)'parking_lot_current_state';
(n)'predicted_incident_data';
(o)'processed_data';
(p)'selected_strategy';
(q)'vehicle_smart_probe_data_for_storage';
(r)'wide_area_pollution_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'current_data';
- (b)'long_term_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;

User Service Requirements: (b) maintain the store of current data in such a way that it only contains data that was obtained

USR = 1.0; over the previous few minutes, e.g. five (5) minutes, so that it presents a rolling picture of the

USR = 1.6; current traffic conditions in the network, which is continually being updated in real time;

USR = 1.6.0; (c) maintain the store of long term data in such a way that it contains the data from the current

USR = 1.6.2; data store aggregated together to provide a complete historical record of the state of the system

USR = 1.6.2.5; over the last hour, every hour in the current day, every day in the current week, etc.;

USR = 1.6.2.5.1; (d) the process shall be responsible for the maintenance of both current and long term data stores,

using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

PROCESS SPECIFICATIONS

Page 5 of 475

Page 6 of 475

1.1.2.2 Process Traffic Data

Input Flows

static_data_for_sensor_processing

traffic_sensor_data

Output Flows

parking_lot_input_data

processed_data

ramp_data

strategy_data_for_highways

strategy_data_for_roads

unusual_data

Description:

Overview: This process shall be responsible for collecting all of the processed data supplied from traffic sensors. The process shall distribute it to processes in the Provide Device Control facility responsible for freeway, parking lot, ramp and road management. It shall also send the data to another process in the Provide Traffic Surveillance facility for loading into the stores of current and long term data.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'traffic_sensor_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval:

- (a)'static_data_for_sensor_processing'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'ramp_data';
- (b)'parking_lot_input_data';
- (c)'strategy_data_for_highways';
- (d)'strategy_data_for_roads';
- (e)'unusual_data';
- (f)'processed_data'..

Functional Requirements: This process shall meet the following functional requirements:

- (a) run whenever the any of the unsolicited input data flow listed above is received;
- (b) use the data solicited input data flow to determine data from which sensors is allocated to the various solicited output flows shown in (a) through (c) above, and send them to the appropriate processes in the Provide Device Control facility;
- (c) be capable of analyzing the input data to detect that congestion is occurring and to pass this through the solicited output flow in (d) to the Manage Incidents facility;
- (d) the process shall read data from the store of static data, using the appropriate mechanism(s) such as RDBMS.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 7 of 475
Page 8 of 475

1.1.2.3 Update Data Source Static Data

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

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 data for use by other processes within the Manage Traffic function. The store shall contain data showing the relationship between sensors and the road 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 data about the ownership of each link in the network which shall be used by processes involved in exchanging data with other Traffic Management Centers (TMC's). The contents of the store shall be provided by the Plan System Deployment function.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'link_data_update';
- (b)'new_sensor_static_data';
- (c)'request_sensor_static_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'static_data_for_sensor_processing';
- (b)'existing_sensor_static_data';
- (c)'link_details'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) the process shall be responsible for the maintenance of the store of static data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 9 of 475

1.1.2.4 Monitor HOV lane use

Input Flows

hov_lane_data_input
static_data_for_sensor_processing

Output Flows

hov_lane_data
hov_lane_violation

Description:

Overview: This process shall be responsible for monitoring the use of High Occupancy Vehicle (HOV) lanes and detecting any 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'hov_lane_data_input'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval:

(a)'static_data_for_sensor_processing'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

(a)'hov_lane_data';

(b)'hov_lane_violation'.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the unsolicited input flows listed above;

(b) be capable of processing the traffic video image data to determine both the number of vehicles using the HOV lane(s) and the identity of vehicle(s) using the lane(s);

(c) the vehicle identities must only be used when the number of occupants in a vehicle is found to be less than that needed for the vehicle to legitimately use the HOV lane;

(d) vehicle identities must not be passed to the data storage process and must only be sent to the Manage Emergency Services function in the event of a violation being detected;

(e) the process shall read data from the store of static data, using the appropriate mechanism(s) such as RDBMS.

PROCESS SPECIFICATIONS

Page 10 of 475

1.1.2.5 Process Tag Data for Link Time Data

Input Flows

link_time_calculation_read

probe_data_for_traffic

static_data_for_sensor_processing

vehicle_tag_data

Output Flows

link_data_from_tags

link_time_calculation_write

Description:

Overview: This process shall be responsible for processing tag data collected from roadside readers or obtained from an analysis of toll transaction records. The process shall receive the processed tag data from the data collection process and shall calculate the journey time for the links that have been traveled by the vehicles carrying the tags. This shall be achieved by noting the successive times at which the data is received from a tag and calculating the journey time from the difference. The data obtained from the toll transaction record analysis shall not need any further processing as it shall contain the average journey times between successive toll collection plazas. The process shall maintain a data store which contains the average journey time for each link in the road and freeway network calculated from the previously described data. Calculation of the actual average values shall employ some type of smoothing (or similar) technique and be stored for differing times of day in fifteen minute increments. The current queue time for a link shall be the difference between current journey time value and the average value for that time of day..

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

(a)'probe_data_for_traffic';

(b)'vehicle_tag_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval:

(a)'static_data_for_sensor_processing'- which contains data about links and tag reading points;

(b)'link_time_calculation_read'- which contains tag data received from reading points that has been previously processed.

Solicited Output Processing: This process shall provide the following output flows as a result of the above input being received:

(a)'link_data_from_tags'.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the solicited data flows shown above;

- (b) when the second data flow in (a) is received store the tag data and check for any previous receipt of the same tag data;
- (c) if any previous occurrences of a particular tag data are found, compute the link journey time
- (d) if the value in (c) is realistic, i.e. it is not due to the vehicle being parked somewhere for a period of time, check it against the journey time value for the time of day already held in the data store;
- (e) if the new value shows a minor variation on the stored values, then use it to recalculate the average link journey time for the time of day and update the data store;
- (f) if the new value shows a significant difference from the stored value then calculate the link queue time from the difference between the two values;
- (g) following completion of (e) or (f), or the receipt of the first unsolicited input flow shown above, generate the output of link journey and queue times, setting the values for any links for there is no journey time data to zero (0);
- (h) the process shall be responsible for the maintenance of the store of link calculation data, using the most appropriate mechanism(s) such as RDBMS, for storing the data;
- (i) the process shall read data from the store of static data, using the appropriate mechanism(s) such as RDBMS.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 11 of 475
Page 12 of 475

1.1.2.6 Process Collected Vehicle Smart Probe Data

Input Flows

vehicle_smart_probe_input_data

Output Flows

vehicle_smart_probe_data_for_storage
vehicle_smart_probe_data_indication

Description:

Overview: This process shall be responsible for the processing of vehicle smart probe data. The process shall take the data retrieved from the vehicle and convert it into a code to indicate the type of hazard. This conversion shall be based on a table of fixed report codes. The process shall send the converted data to the Provide Device Control facility for output to future passing vehicles. It shall send this data, together with the unit identity and location to the traffic data storage process for loading into the current and long term data stores.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flow:
(a)'vehicle_smart_probe_input_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above input being received:

- (a)'vehicle_smart_probe_data_for_storage';
- (b)'vehicle_smart_probe_data_indication'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited data flow shown above;
- (b) when the data flow in (a) is received convert it into a hazard code using data obtained from the store of static data;
- (c) if no code exists for the input, then generate a null (no hazard) code;
- (d) send the data produced in (b) to the vehicle smart probe data output process in the Provide Device Control facility;
- (e) combine the data in (b) with the unit location obtained from the store of static data and send it to the traffic data storage process;
- (f) the process shall read data from the store of static data, using the appropriate mechanism(s) such as RDBMS.

PROCESS SPECIFICATIONS

Page 13 of 475

1.1.3 Generate Predictive Traffic Model

Input Flows

current_incident_data
fws_predicted_weather
long_term_data
other_traffic_center_data

predicted_incidents
predictive_model_data
selected_strategy

Output Flows

prediction_data
unusual_congestion

Description:

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

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'current_incident_data';
- (b)'fws-predicted_weather';
- (c)'predicted_incidents';
- (d)'selected_strategy'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:

- (a)'long_term_data';
- (b)'other_traffic_center_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'prediction_data';
- (b)'predictive_model_data';
- (c)'unusual_congestion'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above storing the received data internally;
- (b) periodically (at least every 15 minutes) produce a prediction of the traffic flow conditions within the road network served by the Manage Traffic function, highlighting any segments on which unusual congestion will form;
- (c) the process shall be responsible for the maintenance of the store of predictive data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

PROCESS SPECIFICATIONS

Page 14 of 475
Page 15 of 475

1.1.4.1 Retrieve Traffic Data

Input Flows

current_data
long_term_data
predictive_model_data
request_traffic_operations_data
traffic_data_demand_request
traffic_data_deployment_request
traffic_data_media_request

Output Flows

retrieved_traffic_operations_data
traffic_data_for_demand
traffic_data_for_deployment
traffic_data_for_media
traffic_data_for_signage

Description:

Overview: This process shall be responsible for retrieving traffic data on request 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 from Manage Demand facility and the Plan System Deployment function, all requests shall be provided by interface processes. The process shall also periodically generate traffic data for output by other processes to in-vehicle signage units.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'request_traffic_operations_data';
- (b)'traffic_media_data_request';
- (c)'traffic_data_demand_request';
- (d)'traffic_data_for_deployment_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'current_data';
- (b)'long_term_data';
- (c)'predictive_model_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'retrieved_traffic_operations_data';
- (b)'traffic_data_for_demand';
- (c)'traffic_data_for_media';
- (d)'traffic_data_for_deployment';
- (e)'traffic_data_for_signage'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited data flows listed above;
- (b) when any of the flows in (a) are received, retrieve data from the current, long term and predictive model data stores, using the solicited input data flows listed above;
- (c) generate the solicited output flow listed above that corresponds to the input flow, loading into it the data appropriate to the recipient;
- (d) as part of (c) note that the content of the flow of traffic data for the media is different to that for the other outputs;
- (e) periodically generate the data flow sent to the traffic control process responsible for sending data to processes that broadcast to in-vehicle signage equipment;
- (f) the process shall be retrieve data from the stores of current, long term and predictive data, using the most appropriate mechanism(s) such as RDBMS.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 16 of 475
Page 17 of 475

1.1.4.2 Provide Traffic Operations Personnel Traffic Data Interface

Input Flows

ftop_traffic_data_parameter_updates
ftop_traffic_information_requests
map_data_for_traffic_display
retrieved_traffic_operations_data

Output Flows

request_traffic_map_display_update
request_traffic_operations_data
traffic_data_media_parameters
ttop_traffic_control_information_display
ttop_video_image_output

Description:

Overview: This process shall be responsible for providing 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 via a separate process to the media operator. This stored data shall comprise current and long term (historic) data on traffic conditions and roadside equipment activity, plus

predictions of traffic conditions. The data shall apply some or all of the road and freeway network served by 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 road and freeway network served by the function. The process shall obtain the map in digitized form from a local data store, which it shall enable the traffic operations personnel to update as and when required.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flow:
(a)'ftop-traffic_information_requests'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes:
(a)'retrieved_traffic_operations_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'request_traffic_operations_data';
(b)'request_traffic_map_display_update'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor the input data flows and provide acknowledgment of receipt of those from traffic operations personnel within 1 second;
(b) be capable of accepting input from traffic operations personnel in audio or other forms, where the latter may comprise input from any combination of keyboards or other forms of push-button devices, pointing devices, etc.;
(c) be capable of carrying out its own verification of input data received from traffic operations personnel and generating the correct solicited output data flow as a result of input data being received;
(d) as part of the output generation process, carrying out checks for data out of range, missing or containing spurious values and requesting re-input where required;
(e) be capable of simultaneously handling a large number of independent input/output data channels, i.e. supporting access by very many traffic operations personnel;
(f) providing all output to traffic operations personnel in a form that is readily understood by a human operator and which may be in audio or visual form, with the latter being available in a variety of formats, e.g. displays, or hardcopy (paper) output;
(g) only generate the outputs listed above as a result of receiving inputs from the traffic operations personnel or other processes;
(h) the use of the digitized map display shall be automatic and shall be at a resolution best suited to the quantity and scope of data being displayed, i.e. the map shall be to the largest possible scale.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 18 of 475
Page 19 of 475

1.1.4.3 Provide Media Operator Traffic Data Interface

Input Flows

fmo_traffic_information_output_request
fmo_traffic_information_request
map_data_for_traffic_display
retrieved_traffic_media_data

Output Flows

request_traffic_media_data
tmo_traffic_information

Description:

Overview: This process shall be responsible for providing the interface between the media operator 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 operator to request and be provided with displays of current, long term (historic) and predicted traffic data. The output of this data shall be against a background of a map of relevant part(s) of the road and freeway served by the Manage Traffic function. The media operator shall only be able to request and see displayed that data that the traffic operations personnel have made available, through the use of the data contained in the store of traffic data

retrieval parameters. The process shall also enable the media operator to select data for output through the process that interfaces to the Media System.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fmo-traffic_information_output_request';
- (b)'fmo-traffic_information_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:

- (a)'map_data_for_traffic_display';
- (b)'retrieved_traffic_media_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'request_traffic_media_data';
- (b)'tmo-traffic_information'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor the input data flows and provide acknowledgment of receipt of those from the media operator within 1 second;
- (b) be capable of accepting input from the media operator in audio or other forms, where the latter may comprise input from any combination of keyboards or other forms of push-button devices, pointing devices, etc.;
- (c) be capable of carrying out its own verification of input data received from the media operator and generating the correct solicited output data flow as a result of input being received;
- (d) as part of the output generation process, carrying out checks for data out of range, missing or spurious values and requesting re-input where necessary;
- (e) be capable of simultaneously handling a large number of independent input/output data channels, i.e. supporting very many media operators, some of whom may be remote;
- (f) providing all output to the media operator in a form that is readily understood by a human operator and which may be in audio or visual form, with the latter being available in a variety of formats, e.g. displays, or hardcopy (paper) output;
- (g) only generate the outputs listed above as a result of receiving inputs from the media operator or the other processes;
- (h) the use of the digitized map display shall be automatic and shall be at a resolution best suited to the quantity and scope of data being displayed, i.e. the map shall be to the largest possible scale.

PROCESS SPECIFICATIONS

Page 20 of 475

PROCESS SPECIFICATIONS

Page 21 of 475

1.1.4.4 Update Traffic Display Map Data

Input Flows

fmup_traffic_display_update
request_traffic_map_display_update

Output Flows

map_data_for_traffic_display
tmup_request_traffic_display_update

Description:

Overview: This process shall be responsible for providing 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 a specialist data supplier or some other appropriate data source.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flow:

- (a)'request_traffic_display_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to external functions:

- (a)'fmup-traffic_display_update'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'tmup-request_traffic_display_update';
- (b)'map_data_for_traffic_display'.

Functional Requirements: This process shall meet the following requirements:

- (a) continuously monitor for the receipt of the unsolicited data flow shown above;
- (b) when the data flow in (a) is received, generate the first solicited output data flow shown above and continuously monitor for receipt of the solicited input data flow shown above;
- (c) when the flow in (b) is received, output the second solicited output data flow shown above;
- (d) be capable of receiving the input data in a variety of formats and converting it into a single format suitable for use with the store of digitized map data;
- (e) be responsible for the management of the data in the store of digitized map data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 22 of 475

1.1.4.5 Provide Media System Traffic Data Interface

Input Flows

unusual_traffic_data

Output Flows

confirm_traffic_data_output

tm_traffic_information

Description:

Overview: This process shall be responsible for providing the interface through which traffic data can be output to the Media System. The output shall only comprise traffic data that is suitable for output by the Media System, e.g. notification of action(s) to be taken by drivers to avoid the impact of incidents. The traffic data shall be provided by another process within the Provide Traffic Surveillance facility of the Manage Traffic function. This interface is only for the output of data that has been requested by the media operator and only provides confirmation that the data has been successfully output.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'unusual_traffic_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'tm-traffic_information';
- (b)'confirm_traffic_data_output'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited data flow shown above;
- (b) when received convert the unsolicited data into a form that can be output either by synthesized voice and/or visual display, depending on the output facilities available to the Media System, and output it in the first solicited output data flow;
- (c) provide confirmation that the output has been transmitted through the second solicited output data flow.

PROCESS SPECIFICATIONS

Page 23 of 475

1.1.4.6 Provide Traffic Data Retrieval Interface

Input Flows

confirm_traffic_data_output

request_traffic_media_data

traffic_data_advisory_request

traffic_data_for_media

traffic_data_guidance_request

traffic_data_kiosk_request

traffic_data_media_parameters

traffic_data_portables_request

traffic_data_retrieval_parameters

traffic_data_ridesharing_request

Output Flows

retrieved_traffic_media_data
traffic_data_for_advisory_output
traffic_data_for_guidance
traffic_data_for_kiosks
traffic_data_for_portables
traffic_data_for_ridesharing
traffic_data_media_request
unusual_traffic_data

Description:

Overview: This process shall be responsible for acting as the interface between processes in other ITS functions that need traffic data and the actual retrieval process within the Provide Traffic Surveillance facility of the Manage Traffic function. This process shall use the parameters in the data store set up the traffic operations personnel to define exactly what data shall be retrieved as a result of each request. It shall be possible for requests for data retrieval to come from the Media Operator, or processes in other ITS functions. When requested by the Media Operator, the process shall output data to the Media System interface process, again subject to the constraints provided by the retrieval parameters set up by the traffic operations personnel.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'request_traffic_media_data';
- (b)'traffic_data_advisory_request';
- (c)'traffic_data_guidance_request';
- (d)'traffic_data_kiosk_request';
- (e)'traffic_data_portables_request';
- (f)'traffic_data_media_parameters';
- (g)'traffic_data_ridesharing_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'confirm_traffic_data_output';
- (b)'traffic_data_retrieval_parameters';
- (c)'traffic_data_for_media'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'unusual_traffic_data';
- (b)'retrieved_traffic_media_data';
- (c)'traffic_data_media_request';
- (d)'traffic_data_for_advisory_output';
- (e)'traffic_data_for_guidance';
- (f)'traffic_data_for_kiosks';
- (g)'traffic_data_for_portables';
- (h)'traffic_data_for_ridesharing'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of any of the unsolicited input data flows listed above;
- (b) when the flow received in (a) is a request for data, send the request to the data retrieval process using the request solicited output data flow shown above;
- (c) when the response to the request flow in (b) is received, assemble the data for output according to the data in the store of traffic data retrieval parameters;

User Service Requirements:

(d) when (c) is complete, send the retrieved data to the requesting process in the corresponding solicited output flow shown above;

USR = 1.0; (e) if the request from the media operator was for output of data to the media system, when the data

USR = 1.6; flow to the media system has been sent, continuously monitor for receipt of the data output

USR = 1.6.0; confirmation flow listed above, and output its contents in the data flow to the media operator

USR = 1.6.3; interface process;

USR = 1.6.3.4; (f) if the data flow in (a) contains new data for the store of traffic data retrieval parameters,

USR = 1.6.3.4.1; load it into the store;

(g) the process shall be responsible for the maintenance of the store of traffic data retrieval

Output Flow Dynamics Assumptions: parameters, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 24 of 475

PROCESS SPECIFICATIONS

Page 25 of 475

1.1.5 Exchange data with Other Traffic Centers

Input Flows

current_incidents_local_data
cv_incidents_for_other_TMC
emergency_data_for_other_TMC
fotc_data_request
fotc_identity
fotc_transfer_data
link_details
long_term_data
predicted_incidents_local_data
request_other_current_incidents_data
request_other_predicted_incidents_data
request_other_TMC_data

Output Flows

other_current_incidents
other_predicted_incidents
other_TMC_cv_incidents
other_TMC_emergency_data
other_TMC_strategy_data
other_traffic_center_data
request_local_current_incidents_data
request_local_predicted_incidents_data
totc_data_request
totc_identity
totc_transfer_data

Description:

Overview: This process shall be responsible for exchanging data with similar processes in other Traffic Management Centers (TMC's) that serve parts of the national road network which are either adjacent geographically or under the control of a different jurisdiction. The exchange of data shall be triggered by either a request from a remote TMC for data from the TMC to which the Manage Traffic function belongs, or because data needs to be sent from the local TMC to a remote TMC. This data shall comprise that for emergency vehicle green waves or special commercial vehicle routes which pass through the local road and freeway network but have a destination in an area served by a remote TMC, or be data about an incident that has an impact on the traffic conditions in the network served by a remote TMC. The data received from remote TMC's shall be used either to vary the current traffic control strategy to give priority to emergency vehicles or enable the passage of commercial vehicles with unusual loads, or as input to the local predictive model generation process.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'cv_incidents_for_other_TMC';
- (b)'emergency_data_for_other_TMC';
- (c)'fotc-data_request';
- (d)'fotc-identity';
- (e)'fotc-transfer_data';
- (f)'request_other_TMC_data'
- (g)'request_other_current_incidents_data';
- (h)'request_other_predicted_incidents_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:

- (a)'link_details';
- (b)'long_term_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from the other TM terminator:

- (a)'fotc-identity';
- (b)'fotc-transfer_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

User Service Requirements: (a)'other_current_incidents';

- (b)'other_predicted_incidents';
- USR = 1.0; (c)'other_TMC_cv_incidents';
- USR = 1.6; (d)'other_TMC_emergency_data';
- USR = 1.6.0; (e)'other_TMC_strategy_data';
- USR = 1.6.2; (f)'other_traffic_center_data';
- USR = 1.6.2.5; (g)'request_local_current_incidents_data';
- USR = 1.6.2.5.2; (h)'request_local_predicted_incidents_data';
- USR = 1.6.3; (i)'totc-data_request';
- USR = 1.6.3.6; (j)'totc-identity';
- (k)'totc-transfer_data'.

Output Flow Dynamics Assumptions:

Functional Requirements: This process shall meet the following functional requirements:

other_current_incidents = fotc-transfer_data; (a) continuously monitor for receipt of the unsolicited input flows listed above;

other_predicted_incidents = fotc-transfer_data; (b) when either of the first two unsolicited inputs are received, generate the last two solicited

other_TMC_cv_incidents = fotc-transfer_data; output data flows and send them to the other TM terminator;

other_TMC_emergency_data = fotc-transfer_data; (c) when the third and fourth unsolicited input data flows are received, if requested, read the data

other_TMC_strategy_data = fotc-transfer_data; from the long term data store that is relevant to the requesting TMC, and again generate the last

other_traffic_center_data = fotc-transfer_data; two solicited output data flows, sending them to the other TM terminator;

request_local_current_incidents_data = fotc-transfer_data; (d) when the fourth and fifth unsolicited input data flows are received, generate those of the first

request_local_predicted_incidents_data = fotc-transfer_data; four solicited output data flows for which data has been provided and send them to their receiving

totc-data_request = request_other_TMC_data; processes, or in the case of the fourth flow load the data into the store containing other traffic

totc-identity = request_other_TMC_data; center data;

totc-transfer_data = fotc-data_request; (e) when any of the last three unsolicited input data flows is received, generate the ninth and

tenth solicited output data flows and send them to the other TM terminator;

(f) the process shall be responsible for the maintenance of the store of data from other TMC's for use by the predictive modeling process, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 26 of 475
Page 27 of 475

1.1.6 Collect Vehicle Tag Data for Link Time Calculations

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

Description:

Overview: This process shall be responsible for collecting data from toll and parking tags on passing vehicles. This shall be achieved by transmitting a tag data request message at periodic intervals (more than once per second) and collecting any reply data that is received. This reply data shall be translated by the process into a form that does not represent actual data that is stored upon the tag, e.g. credit identity or stored credit value, and then passed on to another process for further analysis.

Unsolicited Output Processing: This process shall provide the following output flows regardless of whether or not it has received any input flows:

- (a)'parking_lot_tag_data_needed';
- (b)'toll_tag_data_needed'.

Solicited Input Processing: This process shall receive the following data flows as a result of data being sent to other processes:

- (a)'parking_lot_tag_data_input';
- (b)'toll_tag_data_input'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'vehicle_tag_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) output the unsolicited output flows list above at intervals of not less than once every tenth of a second, i.e. six hundred (600) times per minute;
- (b) when either of the solicited input flows shown above are received, convert the tag data into a form which is unique and can only apply to that data;
- (c) output the data obtained in (b) to the analysis process in the vehicle tag data message.

PROCESS SPECIFICATIONS

Page 28 of 475

1.1.7 Collect Vehicle Smart Probe Data

Input Flows

vehicle_smart_probe_data

Output Flows

vehicle_smart_probe_input_data

Description:

Overview: This process shall be responsible for collecting data from vehicle smart probes. It shall receive this data from passing vehicles and shall add its own identity and location before sending the data on to another process. The process shall maintain output containing its last received input data until new data is received from a passing vehicle.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flow:

- (a)'vehicle_smart_probe_data'.

Solicited Output Processing: This process shall provide the following output flow as a result of the above input being received:

- (a)'vehicle_smart_probe_input_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flow listed above;
- (b) when flow in (a) is received, add the process' location and identity to the data;
- (c) output the combined data produced in (b) to the analysis process in the vehicle smart probe input data message.

PROCESS SPECIFICATIONS

Page 29 of 475

1.2.1 Select Strategy

Input Flows

current_road_network_use
cv_incident_override
demand_overrides
emergency_vehicle_green_wave

ftop_strategy_override
ftop_video_camera_action_request
incident_strategy_override
indicator_fault_state
indicator_input_state_for_highways
indicator_input_state_for_roads
low_traffic_route
other_TMC_cv_incidents
other_TMC_emergency_data
other_TMC_strategy_data
static_data_for_strategy

Output Flows

cv_incidents_for_other_TMC
emergency_data_for_other_TMC
request_other_TMC_data
selected_highway_control_strategy
selected_parking_lot_control_strategy
selected_ramp_control_strategy
selected_road_control_strategy
selected_strategy
traffic_video_camera_control

Description:

Overview: This process shall be responsible for selecting the appropriate traffic control strategy to be implemented over the road and freeway network served by 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, etc. 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. roads, freeways, ramps and parking lots. When part of the selected strategy, or at the request of the traffic operations personnel, the process shall send commands to the traffic sensor data process to change the operating parameters of video cameras used to provide traffic 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 green waves, the passage of commercial vehicles with unusual loads, equipment faults and overrides from the traffic operations personnel. The selected strategy shall be sent to the process within the Provide Traffic Surveillance facility responsible for maintaining the store of long term data.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

(a)'current_road_network_use';
(b)'cv_incident_override';
(c)'demand_overrides';
(d)'emergency_vehicle_green_wave';
(e)'ftop-strategy_override';
(f)'ftop-video_camera_action_request';
(g)'incident_strategy_override';
(h)'indicator_fault_state';
(i)'indicator_input_state_for_highways';
(j)'indicator_input_state_for_roads';
(k)'low_traffic_route';
(l)'other_TMC_cv_incidents';
(m)'other_TMC_emergency_data';
(n)'other_TMC_strategy_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:

(a)'static_data_for_strategy'.

User Service Requirements:

USR = 1.0; Unsolicited Output Processing: This process shall provide the following output flows regardless of

USR = 1.6; any inputs that are received:

USR = 1.6.0; (a)'request_other_TMC_data';

USR = 1.6.3; (b)'traffic_video_camera_control'.

USR = 1.6.3.5;

USR = 1.6.3.6; Solicited Output Processing: This process shall provide the following output flows as a result of

the above inputs being received:

(a)'cv_incidents_for_other_TMC';

Output Flow Dynamics Assumptions: (b)'emergency_data_for_other_TMC';

(c)'selected_parking_lot_control_strategy';

cv_incidents_for_other_TMC = cv_incident_override; (d)'selected_ramp_control_strategy';

emergency_data_for_other_TMC = emergency_vehicle_route; (e)'selected_highway_control_strategy';

request_other_TMC_data = 12/(60*60); (f)'selected_road_control_strategy';

selected_highway_control_strategy = 12/60; (g)'selected_strategy'.

selected_road_control_strategy = 12/60;

selected_parking_lot_control_strategy = 4/60; Functional Requirements: This process shall meet the following functional requirements:

selected_ramp_control_strategy = 12/60; (a) continuously monitor for receipt of the unsolicited input flows listed above;

selected_strategy = 2*(12/60)+4/60; (b) determine the traffic control strategy that provides the best possible traffic conditions within

traffic_video_camera_control = ftp-video_camera_action_request; the road network served by the Manage Traffic function, using the definition of best as being the

one that minimizes stops and delays thus reducing stop-start travel and fuel consumption and the environmental impact of travel;

(c) in determining the strategy, the process shall take account of all the data being provided as input from other parts of the Manage Traffic function, unless countermanded by input from the traffic operations personnel, or the default strategy in the store of static data, as well as the fault state of all indicator equipment;

(d) if no input is available from other parts of the Manage Traffic function, then the strategy defined in the store of static data shall be used;

(e) if in (d) no strategy is specified, the process shall allow all controlled equipment to operate under local control, setting all variable message sign (vms) outputs to blank face indicating that there is no message;

(f) where the inputs from other parts of the Manage Traffic function lead to conflicts in the required strategy to be selected, the process shall observe the following order of priority: emergency vehicle route, incident strategy override, multimodal crossing inputs, operator strategy override, demand strategy override, low traffic volume route, commercial vehicle route, analysis of the road network use and background strategy selection from the store of static data;

(g) data for emergency and commercial vehicle routes sent from other Traffic Management Centers (TMC's) shall be given the same level of importance as those that originate locally;

(h) the process shall automatically cancel strategies selected by traffic operations personnel at midnight on the day that they were imposed if they have not been canceled previously, to avoid unintended effects on the traffic control strategies for other days;

(i) when a new strategy has been determined, it shall be sent to other processes in the Manage Traffic function for implementation;

(j) the output in (i) shall only be sent to those processes that serve equipment specified in the new strategy;

(k) changes in the current strategy must always be immediately sent to another part of the Manage Traffic function for loading into the long term data store.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 30 of 475
Page 31 of 475

1.2.2.1 Determine Indicator State for Freeway Management

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

Description:

Overview: This process shall be responsible for implementing 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 (variable message signs (vms), etc.) 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 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.

Data Flows: All input data flows are unsolicited and all output data flows are solicited with the exception of the following:

(a)'static_data_for_highways', which is data received from a local data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) immediately implement any strategy requests only using the indicators specified in the request;
- (c) it shall be possible for the strategy request to require implementation on one, some or all the indicators that are available (and not faulty) in the freeway network served by the Manage Traffic function;
- (d) strategy implementation must make use of the freeway sign sequences to ensure that signs are set in a manner that is safe for all freeway users;
- (e) emergency and commercial vehicle routes shall be implemented as 'rolling green waves' so that priority is given to the vehicles as they progress through the freeway network without them having to adjust their speed;
- (f) requests for high occupancy vehicle (hov) and transit priority shall be actioned immediately but not take precedence over emergency vehicle routes;
- (g) special consideration must be given to conditions in tunnels and in particular to the need to automatically implement alternative traffic management strategies to route traffic away from fires that may be detected;
- (h) the process shall use the strategy data input to monitor the effects of the currently selected strategies and make any small adjustments which will further improve the efficiency and lower the environmental impact of the current traffic flow;
- (i) transit priority shall be implemented on the indicators covering the requested route(s) and its confirmation of its implementation shall be sent back to the requesting process in the Manage Transit function;
- (j) the process shall implement any changes in control in a safe manner that does not in any way endanger vehicles and/or their drivers, or pedestrians;
- (k) send each change in strategy to another process in the Manage Traffic function for loading into the store of long term data;
- (l) send the required indicator state to another process in the Manage Traffic function for output to the roadside equipment that drives the indicators.

PROCESS SPECIFICATIONS

Page 32 of 475

PROCESS SPECIFICATIONS

Page 33 of 475

1.2.2.2 Determine Indicator State for Road Management

Input Flows

coordination_data_freeways_to_roads
coordination_data_ramps_to_roads
prediction_data
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

Description:

Overview: This process shall be responsible for implementing 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 (vms), 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.

Data Flows: All input data flows are unsolicited and all output data flows are solicited with the exception of the following:

(a)'static_data_for_roads', which is data received from a local data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) immediately implement any strategy requests only using the indicators specified in the request;
- (c) unless specifically requested otherwise, the control of all indicators that are intersection and pedestrian controllers shall use an adaptive methodology which is capable of responding smoothly and seamlessly, i.e. without any step changes in control parameters, in real time to any slight changes in vehicle delays;
- (d) where adaptive control cannot be implemented, or is not specified in the strategy request, the following traffic control methodologies shall be available to the process for implementation within some or all of the controlled network: fixed time control sequences (usually referred to as fixed time plans), the automatic selection of the most appropriate fixed time plan on the basis of current real time traffic data, the selection of special fixed time plans to cover such things as bridges opening when requested by the specific data input and the ability of one or more device(s) to operate under its own (local) control;
- (e) the process must be capable of implementing the required control strategy on one, some or all the indicators that are available (and not faulty) in the road network;
- (f) emergency and commercial vehicle routes shall be implemented as 'rolling green waves' so that priority is given to the vehicles as they progress through the road network without them having to adjust their speed;
- (g) requests for high occupancy vehicle (hov) and transit priority shall be actioned immediately but not take precedence over emergency vehicle routes;
- (h) the process shall use the strategy data input to monitor the effects of the currently selected strategies and make any small adjustments which will further improve the efficiency and lower the environmental impact of the traffic flow;
- (i) transit priority shall be implemented on the indicators covering the requested route(s) and its confirmation of its implementation shall be sent back to the requesting process in the Manage Transit function;
- (j) the process shall implement any changes in control in a safe manner that does not in any way endanger vehicles and/or their drivers, or pedestrians;
- (k) send each change in strategy to another process in the Manage Traffic function for loading into the store of long term data;
- (l) send the required indicator state to another process in the Manage Traffic function for output to the roadside equipment that drives the indicators.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 34 of 475
Page 35 of 475

1.2.3 Determine Ramp State

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

Description:

Overview: This process shall be responsible for implementing 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 open/close decisions on the data from sensors monitoring traffic conditions upstream and downstream of the ramps. The decision making process shall use an algorithm

to determine the ramps state so that the traffic conditions on the freeway do not significantly deteriorate due to the traffic entering from the ramp. The process shall coordinate its activities with the process responsible for controlling the road (surface street) network. In particular this coordination shall ensure that traffic conditions in the road (surface street) network do not significantly deteriorate due to the closure of a ramp.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'coordination_data_roads_to_ramps';
- (b)'ramp_data';
- (c)'selected_ramp_control_strategy';
- (d)'transit_ramp_overall_priority';
- (e)'transit_ramp_preemptions'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:

- (a)'static_data_for_ramps'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'coordination_data_ramps_to_roads';
- (b)'current_ramp_state';
- (c)'ramp_signal_state';
- (d)'transit_ramp_priority_given'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) in its control of the ramp metering equipment, the process shall use a special algorithm which enables traffic to be admitted to a freeway only if it can be absorbed into the existing flow without causing serious disruption, i.e. the existing flow is not at a level which produces the most efficient traffic flow;
- (c) the inputs to the algorithm in (b) shall be obtained by processing traffic flow data received from sensors which monitor freeway conditions both upstream and downstream of the ramp;
- (d) the process shall override the requirements in (b) and open the ramp if coordination data received from the process controlling the roads (surface streets) indicates that closing it will have a greater negative impact on the road (surface street) network that is upstream of the ramp;
- (e) give priority to high occupancy vehicles (hov) and transit vehicles, particularly when priority requests are received for the latter;
- (f) transit priority shall be implemented on the indicators covering the requested route(s) and its confirmation of its implementation shall be sent back to the requesting process in the Manage Transit function;
- (g) the process must be capable of implementing the required control strategy on one, some or all the ramps that are available (and not faulty) in the freeway network served by the function;
- (h) send each change in strategy to another process in the Manage Traffic function for loading into the store of long term data;
- (i) output coordination data to the process controlling the roads (surface streets).

PROCESS SPECIFICATIONS PROCESS SPECIFICATIONS

Page 36 of 475
Page 37 of 475

1.2.4.1 Output Control Data for Roads

Input Flows

indicator_input_data_from_roads

indicator_road_requested_state
parking_guidance_for_vms
static_data_for_road_control
vehicle_pollution_message_for_roads
vms_updates_for_roads

Output Flows

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
vehicle_sign_data_for_roads

Description:

Overview: This process shall be responsible for the transfer of 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, variable message signs (vms), 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'indicator_input_data';
- (b)'indicator_highway_requested_state';
- (c)'indicator_road_requested_state';
- (d)'ramp_signal_state';
- (e)'vehicle_pollution_message';
- (f)'vms_parking_guidance';
- (g)'vms_updates'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:

- (a)'static_data_for_control'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'indicator_control_data';
- (b)'indicator_control_monitoring_data';
- (c)'indicator_control_storage_data';
- (d)'indicator_data_fault';
- (e)'indicator_input_state';
- (f)'indicator_input_storage_data';
- (g)'vehicle_sign_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when any change occurs to the input data, change the appropriate indicator output data;
- (c) as a result of (b), update the vehicle signage data, adding the location and identity of the route segments from which the indicator data can be seen from the static data for vms allocation;
- (d) maintain communication with all indicators so that they will continue to obey the data contained in the data that is being sent to them;
- (e) immediately report all indicators that fail to respond to the commands in the data that they have been sent to the processes responsible for fault management.

1.2.4.2 Output Control Data for Freeways**Input Flows**

indicator_highway_requested_state
indicator_input_data_from_highways
parking_guidance_for_vms
ramp_signal_state
static_data_for_highway_control
vehicle_pollution_message_for_highways
vms_updates_for_highways

Output Flows

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
vehicle_sign_data_for_highways

Description:

Overview: This process shall be responsible for the transfer of 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 variable message signs (vms), 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'indicator_input_data';
- (b)'indicator_highway_requested_state';
- (c)'indicator_road_requested_state';
- (d)'ramp_signal_state';
- (e)'vehicle_pollution_message';
- (f)'vms_parking_guidance';
- (g)'vms_updates'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:

- (a)'static_data_for_control'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'indicator_control_data';
- (b)'indicator_control_monitoring_data';
- (c)'indicator_control_storage_data';
- (d)'indicator_data_fault';
- (e)'indicator_input_state';
- (f)'indicator_input_storage_data';
- (g)'vehicle_sign_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when any change occurs to the input data, change the appropriate indicator output data;
- (c) as a result of (b), update the vehicle signage data, adding the location and identity of the route segments from which the indicator data can be seen from the static data for vms allocation;
- (d) maintain communication with all indicators so that they will continue to obey the data contained in the data that is being sent to them;

(e) immediately report all indicators that fail to respond to the commands in the data that they have been sent to the processes responsible for fault management.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 40 of 475
Page 41 of 475

1.2.4.3 Output In-vehicle Signage Data

Input Flows

current_incident_data_for_vehicle_signage
predicted_incident_data_for_vehicle_signage
static_data_for_vehicle_signage
traffic_data_for_signage
vehicle_sign_data_for_highways
vehicle_sign_data_for_roads
vehicle_signage_incident_data

Output Flows

vehicle_sign_data

Description:

Overview: This process shall be responsible for the transfer of data to processes located at the roadside within the road (surface street) an freeway network served by the Manage Traffic function. This data shall contain a copy of the data being sent to indicators such as intersection and pedestrian controllers, variable message signs (vms), etc. covered by each roadside process, plus details of any current or predicted incidents in the area local to the location of each roadside process, and traffic conditions (speed and occupancy) on local links. The data shall be structured by this process so that it can output by each roadside process to vehicles for use by in-vehicle signage equipment. The data shall have been sent to this process by other processes within the Manage Traffic function, which are responsible for the output of traffic control data and the management of incidents.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'current_incident_data_for_vehicle_signage';
- (b)'predicted_incident_data_for_vehicle_signage';
- (c)'static_data_for_vehicle_signage';
- (d)'traffic_data_for_signage';
- (e)'vehicle_sign_data_for_highways';
- (f)'vehicle_sign_data_for_roads'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:

- (a)'vehicle_signage_incident_data'.

Solicited Output Processing: This process shall provide the following output flow as a result of the above inputs being received:

- (a)'vehicle_sign_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when any change occurs in either of the last three or the first two input data flows, change the contents of the signage output data flow to be the same;
- (c) as a result of (b), output the vehicle signage data, filtering the incident data so that the roadside processes only receive data which is relevant to their location, and only providing the sign data that relates to the signs covered by each roadside process;
- (d) the process shall perform the incident location filtering using the roadside process location data in the third input data flow;
- (e) when new incident data is received in either of the first two input data flows, load it into the store of incident data, updating any previously received data that relates to the same incident;
- (f) the process shall be responsible for the maintenance of the store of incident data, using the most appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 42 of 475
Page 43 of 475

1.2.5.1 Determine Parking Lot State

Input Flows

parking_lot_calculated_occupancy
parking_lot_operator_input_data
parking_service_provider_input_data
selected_parking_lot_control_strategy

Output Flows

parking_guidance_for_vms
parking_lot_current_state
parking_lot_occupancy
parking_lot_operational_data
parking_lot_operator_output_data
parking_service_provider_output_data

Description:

Overview: This process shall be responsible for implementing the selected control strategies on some or all of the parking lots in the road 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. The process shall also provide operational data to the Plan System Deployment function and the current parking lot occupancy to other processes in the function, at periodic intervals.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'parking_lot_calculated_occupancy';
- (b)'parking_lot_operator_input_data';
- (c)'parking_service_provider_input_data';
- (d)'selected_parking_lot_control_strategy';
- (e)'static_data_for_parking_lots'.

Unsolicited Output Processing: This process shall provide the following output flow regardless of whether or not any of the above inputs have been received:

- (a)'parking_lot_operational_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'parking_lot_current_state';
- (b)'parking_lot_occupancy';
- (c)'parking_lot_operator_output_data';
- (d)'parking_service_provider_output';
- (e)'vms_parking_guidance_for_highways';
- (f)'vms_parking_guidance_for_roads'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) calculate the current parking lot state by comparing the occupancy provided from one or more other processes with the threshold values contained in the static data provided by the fifth unsolicited input flow shown above;
- (c) if the current parking lot occupancy is available from more than one source, then use the calculated value as first priority;
- (d) if a requested state is received from the strategy selection process, then use it to over-write the calculated parking lot state;
- (e) use the parking lot state to determine the vms state needed to advise motorists of the spaces available in each lot or to guide them to lots which have spaces available, sending the data out through the data flow of variable messages sign (vnms) guidance data;
- (f) send each change in state to another process in the Manage Traffic function for loading into the store of long term data;
- (g) output the unsolicited data flow shown above to the Plan System Deployment function not less than once in a consecutive twenty four (24) period.

1.2.5.2 Provide Parking Service Provider Interface

Input Flows

fpsp_current_lot_state
fpsp_lot_occupancy
parking_service_provider_output_data
parking_service_provider_transit_update

Output Flows

parking_service_provider_input_data
tpsp_change_lot_state

Description:

Overview: This process shall be responsible for providing the interface to a local parking lot system that controls the use of the lot. The parking lot system shall provide inputs of occupancy and/or the current lot state. It shall respond to a request from this process to change the lot state by activating local variable message signs (vms) and controlling the use of entry/exit barriers, and to output through local vms, information about transit services that provide a park and ride (P+R) operation.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fpsp-current_lot_state';
- (b)'fpsp-lot_occupancy';
- (c)'parking_service_provider_output_data';
- (d)'parking_service_provider_transit_update'.

Unsolicited Output Processing: This process shall provide the following output flows regardless of any inputs being received:

- (a)'parking_service_provider_input_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'tpsp-change_lot_state'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) always attempt to provide output of the parking lots current state regardless of the inputs that are (or not) received from the system;
- (c) immediately respond to any requests for changes in the parking lot state received from other processes in the Manage Traffic function, regardless of any local inputs.

PROCESS SPECIFICATIONS

Page 46 of 475

1.2.5.3 Provide Parking Lot Operator Interface

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

Description:

Overview: This process shall be responsible for providing 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 variable message signs (vms) 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 vms.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fpo-current_lot_state';
- (b)'fpo-lot_occupancy';

- (c)'parking_lot_operator_output_data';
- (d)'parking_lot_operator_transit_update'.

Unsolicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'parking_lot_operator_input_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'tpo-change_lot_state'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) always attempt to provide output of the parking lots current state regardless of the inputs that are (or not) received from the operator;
- (c) immediately respond to any requests for changes in the parking lot state received from other processes in the Manage Traffic function, regardless of any local inputs.

PROCESS SPECIFICATIONS

Page 47 of 475

1.2.5.4 Determine P+R needs for Transit Management

Input Flows

- parking_lot_occupancy
- parking_lot_transit_response

Output Flows

- parking_lot_operator_transit_update
- parking_lot_transit_request
- parking_service_provider_transit_update

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'parking_lot_occupancy'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes:

- (a)'parking_lot_transit_response'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'parking_lot_operator_transit_update';
- (b)'parking_service_provider_transit_update';
- (c)'parking_lot_transit_request'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) calculate the need for transit services based on the rate at which vehicles are arriving at the parking lot and pass this information on to the Manage Transit function;
- (c) communicate the response from the Manage Transit function to the processes responsible for monitoring the operation of the parking lot.

PROCESS SPECIFICATIONS

Page 48 of 475

1.2.5.5 Calculate Parking Lot Occupancy

Input Flows

- parking_lot_input_data
- static_data_for_parking_lots

Output Flows

parking_lot_calculated_occupancy

Description:

Overview: This process shall be responsible for calculating 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 car park occupancy.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:
(a)'parking_lot_input_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:
(a)'static_data_for_parking_lots'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'parking_lot_calculated_occupancy'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor the unsolicited input shown above;
- (b) use the static data provided as the solicited input flow shown above to determine which part(s) of the data provided by the data flow in (a) apply to the parking lot's entry and exit lanes;
- (c) use the data identified in (b) to calculate the numbers of vehicles entering and leaving the parking lot and hence calculate the parking lot occupancy;
- (d) output the data obtained in (c) as the solicited output flow shown above.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 49 of 475
Page 50 of 475

1.2.6.1 Maintain Traffic and Sensor Static Data

Input Flows

existing_sensor_static_data
link_data
static_data_for_traffic_control_copy
static_data_request
supply_traffic_static_data

Output Flows

current_traffic_static_data
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

Description:

Overview: This process shall be responsible for maintaining the store of static and link data used as a source of this type of data 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. The data held in the store maintained by this process shall be provided to it by the Plan System Deployment function, to which this process shall be capable of providing the current data and receiving from it updates to the current data.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'existing_sensor_static_data';
(b)'link_data';
(c)'static_data_request';
(d)'supply_traffic_static_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:
(a)'static_data_for_traffic_control_copy'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'current_traffic_static_data';
(b)'link_data_for_guidance';
(c)'link_data_update';
(d)'new_sensor_static_data';
(e)'request_sensor_static_data';
(f)'static_data_for_traffic_control_update';
(g)'static_data_store_updated'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input flows listed above;
(b) communicate with other processes in the Manage Traffic function to obtain their current static data and to provide updates to that data;
(c) when new static data is received and it has been successfully loaded into the store, output the static data updated data flow so that other processes can receive their new copies of the data;
(d) the process shall be responsible for the maintenance of the stores of static and link data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 51 of 475
Page 52 of 475

1.2.6.2 Provide Static Data Store Output Interface

Input Flows

static_data_for_traffic_control_output
static_data_store_updated

Output Flows

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

Description:

Overview: This process shall be responsible for providing 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 other process shall be responsible for receiving the new data from the Plan System Deployment function and for providing copies of the current data to that function.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:
(a)'static_data_store_updated'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'static_data_for_highways';
(b)'static_data_for_highway_control';
(c)'static_data_for_road_control';
(d)'static_data_for_parking_lots';
(e)'static_data_for_ramps';
(f)'static_data_for_roads';
(g)'static_data_for_strategy';
(h)'static_data_for_vehicle_signage'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) when the flow in (a) has been received, read the data from the store of static data and send the solicited output flows listed above;
- (c) the process shall read data from the store of static data, using the appropriate mechanism(s) such as RDBMS.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 53 of 475
Page 54 of 475

1.2.7.1 Process Indicator Output Data for Roads

Input Flows

indicator_control_data_for_roads
indicator_preemption_override_for_roads
local_sensor_data_for_roads

Output Flows

indicator_input_data_from_roads
indicator_response_data_for_roads
intersection_state_data
td_lane_use_indication_for_roads
td_signal_indication
td_vms_indication_for_roads
tmmc_crossing_clear_at_roads
tmmc_stop_alternate_mode_at_roads
tmmc_stop_traffic_at_roads
tp_cross_request_received
tp_cross_road
tp_vms_indication

Description:

Overview: This process shall be responsible for implementing 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 variable message signs (vms), or provide the interface for data to be sent to the units (or systems) that manage multimodal crossings. The vms 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'indicator_control_data_for_roads';
- (b)'indicator_preemption_override_for_roads';
- (c)'local_sensor_data_for_roads'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'indicator_input_data_from_roads';
- (b)'indicator_response_data_for_roads';
- (c)'intersection_state_data
- (d)'td-lane_use_indication_for_roads';
- (e)'td-signal_indication';
- (f)'td-vms_indication_for_roads';
- (g)'tmmc-crossing_clear_at_roads';
- (h)'tmmc-stop_alternate_mode_at_roads';
- (i)'tmmc-stop_traffic_at_roads';
- (j)'tp-cross_request_received';
- (k)'tp-cross_road';
- (l)'tp-vms_indication'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) provide output data in a form, e.g. analog, which is easily understood by drivers and/or travelers, appears in a safe sequence, is unambiguous and does not provide conflicting instructions to drivers and travelers that are likely to result in circumstances which are life threatening;

- (c) all output must be maintained for a time period which is sufficient to enable them to be read, understood and reacted to, but not so long that they cause any new indication to be ignored;
- (d) if no input of fresh control data is received for a continuous period of four (4) seconds, the process shall start to change its outputs based on local sensor data, and shall clear (blank) the outputs containing advisory message texts;
- (e) the outputs to the multimodal crossings ('tmmc-....') shall be maintained for as long as the appropriate control signal is received from other processes, and if no such signals are being received shall be set to null, i.e. the multimodal crossing equipment is not expected to take any action.

PROCESS SPECIFICATIONS

Page 55 of 475

PROCESS SPECIFICATIONS

Page 56 of 475

1.2.7.2 Monitor Roadside Equipment Operation for Faults

Input Flows

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

roadside_fault_data

Description:

Overview: This process shall be responsible for monitoring the operation of the processes that output in-vehicle signage and indicator data in the road (surface street) and freeway network served by the Manage Traffic function. 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, or the in-vehicle signage process reports a fault.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:

- (a)'indicator_control_monitoring_data_for_highways';
- (b)'indicator_control_monitoring_data_for_roads';
- (c)'indicator_monitoring_suspend';
- (d)'indicator_response_data';
- (e)'vehicle_sign_data_output_fault';
- (f)'vehicle_smart_probe_data_output_fault'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'roadside_fault_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) only output the fault data if the indicator response data does not match the control monitoring data for a period of four (4) seconds, and a period of four (4) seconds has elapsed since the control monitoring data last changed;
- (c) report all in-vehicle signage and probe data processing faults as soon as they are detected.

PROCESS SPECIFICATIONS

Page 57 of 475

1.2.7.3 Manage Indicator Preemptions

Input Flows

emergency_vehicle_preemptions
 transit_vehicle_roadway_preemptions

Output Flows

indicator_monitoring_suspend
 indicator_preemption_override_for_highways
 indicator_preemption_override_for_roads

Description:

Overview: This process shall be responsible for receiving preemption requests from other functions within ITS. These requests shall enable the process to give selected vehicles such those that belong to Transit Authorities or Emergency Services, priority at intersections, pedestrian crossings and

multimodal crossings in the road (surface street) and freeway network served by the Manage Transit function. Sending of the priority request output shall also generate an output to the monitoring process to suspend its activities whilst the priority request is being actioned. This process shall only generate its outputs when input data is received.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:

- (a)'emergency_vehicle_preemptions';
- (b)'transit_vehicle_roadway_preemptions'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'indicator_monitoring_suspend';
- (b)'indicator_preemption_override_for_highways';
- (c)'indicator_preemption_override_for_roads'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) maintain both output flows for as long as any of the input flows are present;
- (c) remove the output flows immediately that the input flows cease to exist.

PROCESS SPECIFICATIONS

Page 58 of 475

1.2.7.4 Process In-vehicle Signage Data

Input Flows

vehicle_sign_data

Output Flows

vehicle_sign_data_output_fault

vehicle_signage_data

Description:

Overview: This process shall be responsible for the output of 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 take the form of that provided for any of the types of indicator that are supported by the function, i.e. intersection controller, pedestrian controller, variable message sign (vms), etc., plus data about incidents and link information such as speed and occupancy. The process shall be responsible for its own fault monitoring, which shall check that output data is always 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.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:

- (a)'vehicle_sign_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'vehicle_signage_data';
- (b)'vehicle_sign_data_output_fault'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) continuously generate the first solicited output flow, making sure that it accurately reflects the data received in (a);
- (c) detect any processing faults, such as input data does not match output data, or data output failed;
- (d) as soon a fault is detected in (c), send the second solicited output flow listed above to the fault monitoring process.

PROCESS SPECIFICATIONS

Page 59 of 475

1.2.7.5 Process Indicator Output Data for Freeways

Input Flows

indicator_control_data_for_highways

indicator_preemption_override_for_highways

local_sensor_data_for_highways

Output Flows

indicator_input_data_from_highways
indicator_response_data_for_highways
td_lane_use_indication_for_highways
td_ramp_state_indication
td_vms_indication_for_highways
tmmc_crossing_clear_at_highways
tmmc_stop_alternate_mode_at_highways
tmmc_stop_traffic_at_highways

Description:

Overview: This process shall be responsible for implementing 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, or generate the output for variable message signs (vms). The vms 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'indicator_control_data_for_highways';
- (b)'indicator_preemption_override_for_highways';
- (c)'local_sensor_data_for_highways'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'indicator_input_data_from_highways';
- (b)'indicator_response_data_for_highways';
- (c)'td-lane_use_indication_for_highways';
- (d)'td-ramp_state_indication';
- (e)'td-vms_indication_for_highways';
- (f)'tmmc-crossing_clear_at_highways';
- (g)'tmmc-stop_alternate_mode_at_highways';
- (h)'tmmc-stop_traffic_at_highways'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) provide output data in a form, e.g. analog, which is easily understood by drivers and/or travelers, appears in a safe sequence, is unambiguous and does not provide conflicting instructions to drivers and travelers that are likely to result in circumstances which are life threatening;
- (c) all output must be maintained for a time period which is sufficient to enable them to be read, understood and reacted to, but not so long that they cause any new indication to be ignored;
- (d) if no input of fresh control data is received for a continuous period of four (4) seconds, the process shall start to change its outputs based on local sensor data, and shall clear (blank) the outputs containing advisory message texts.

PROCESS SPECIFICATIONS

Page 60 of 475

1.2.7.6 Provide Intersection Collision Avoidance Data

Input Flows

intersection_state_data
local_sensor_data_for_roads

Output Flows

intersection_collision_avoidance_data

Description:

Overview: This process shall be responsible for providing 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 from the other vehicle(s).

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'intersection_state_data';
- (b)'local_sensor_data_for_roads'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'intersection_collision_avoidance_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) store the first input flow shown above to build up a picture of the way in which the intersection controller is operating;
- (c) process the second input flow in (a) to determine if there are any vehicles for which a collision is a distinct possibility, bearing mind the current and expected state of the intersection;
- (d) as a result of (c) if any vehicles are found to have a collision threat, continuously generate the solicited output flow shown above and send it to the vehicle(s) that is(are) likely to receive the impact from the other vehicle(s);
- (e) discontinue with (d) when the threat of collision to the vehicle(s) has gone away.

PROCESS SPECIFICATIONS

Page 61 of 475

1.2.7.7 Process Vehicle Smart Probe Data for Output

Input Flows

vehicle_smart_probe_data_indication

Output Flows

vehicle_smart_probe_data_output

vehicle_smart_probe_data_output_fault

Description:

Overview: This process shall be responsible for the output of 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 over a very short range so that it can only be received by those vehicles that are passing. The process shall perform its own fault detection and report and that are found to the fault monitoring process.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:

- (a)'vehicle_smart_probe_data_indication'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'vehicle_smart_probe_data_output';
- (b)'vehicle_smart_probe_data_output_fault'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) continuously generate the first solicited output flow, making sure that it accurately reflects the data received in (a);
- (c) detect all processing faults, such as input data does not match output data, data output failed, input data not received, input data does not contain a valid code;
- (d) as soon as a fault is detected in (c), generate the second solicited output flow shown above.

PROCESS SPECIFICATIONS

Page 62 of 475

1.2.8.1 Collect Indicator Fault Data

Input Flows

indicator_data_fault_for_highways

indicator_data_fault_for_roads

roadside_fault_data

Output Flows

indicator_new_fault_update

Description:

Overview: This process shall be responsible for collecting data about faults in the operation of indicators 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.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:

- (a)'indicator_data_fault_for_highways';
- (b)'indicator_data_fault_for_roads';
- (c)'roadside_fault_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'indicator_new_fault_update'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when any of the flows in (a) is received, pass the data on to another process for storage and the activation of fault reporting procedures.

PROCESS SPECIFICATIONS

Page 63 of 475

1.2.8.2 Maintain Indicator Fault Data Store

Input Flows

indicator_current_fault_update
indicator_current_faults_list
indicator_fault_clearance_update
indicator_new_fault_data
indicator_new_fault_update

Output Flows

indicator_current_fault_data
indicator_fault_state
indicator_new_fault

Description:

Overview: This process shall be responsible for collecting 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 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 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 fault(s).

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'indicator_current_fault_update';
- (b)'indicator_new_fault_update';
- (c)'indicator_fault_clearance_update';
- (d)'indicator_new_fault_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'indicator_current_fault_data';
- (b)'indicator_current_faults_list';
- (c)'indicator_fault_state';
- (d)'indicator_new_fault'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) the process shall be responsible for the maintenance of the store of the indicator fault state data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 64 of 475

1.2.8.3 Provide Indicator Fault Interface for C and M

Input Flows

fcm_fault_clearance
indicator_new_fault

Output Flows

indicator_fault_clearance_update
tcm_fault_data

Description:

Overview: This process shall be responsible for providing 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.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:
(a)'indicator_new_fault'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to the construction and maintenance terminator:
(a)'fcm-fault_clearance'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'indicator_fault_clearance_update';
(b)'tcm-fault_data'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input flows listed above;
(b) use the most appropriate communications protocol compatible with the physical communications

links

to exchange data with the Construction and Maintenance terminator;
(c) be capable of supporting more than one physical link to more than one Construction and Maintenance organization.

PROCESS SPECIFICATIONS

Page 65 of 475

1.2.8.4 Provide Traffic Operations Personnel Indicator Fault Interface

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

Description:

Overview: This process shall be responsible for providing the interface through which traffic operations personnel can have access to the data about faults on indicator equipment controlled by the Manage Traffic function. The process shall enable the personnel to monitor the number of 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'ftop-indicator_fault_data_input';
(b)'ftop-indicator_fault_data_request';
(c)'ftop-indicator_fault_data_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:
(a)'indicator_current_fault_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'indicator_current_fault_update';
- (b)'indicator_new_fault_data';
- (c)'ttop-current_indicator_faults'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) be capable of accepting input from Traffic Operations Personnel in audio or other forms, where the latter may comprise input from any combination of keyboards or other forms of push-button devices, pointing devices, etc.;
- (c) be capable of carrying out its own verification of input data received from Traffic Operations Personnel and generating the correct solicited output data flow as a result of input data being received;
- (d) as part of the output generation process, carrying out checks for data out of range, missing or spurious values and requesting re-input where required;
- (e) be capable of simultaneously handling a large number of independent input/output data channels, i.e. supporting very many Traffic Operations Personnel;
- (f) providing all output to Traffic Operations Personnel in a form that is readily understood by a human operator and which may be in audio or visual form, with the latter being available in a variety of formats, e.g. displays, or hardcopy (paper) output;
- (g) only generate the outputs listed above as a result of receiving inputs from the Traffic Operations Personnel or the other processes.

PROCESS SPECIFICATIONS

Page 66 of 475

PROCESS SPECIFICATIONS

Page 67 of 475

1.3.1.1 Analyze Traffic Data for Incidents

Input Flows

current_road_network_use
incident_analysis_data
static_data_for_incident_management
traffic_image_data
unusual_data

Output Flows

possible_detected_incidents

Description:

Overview: This process shall be responsible for analyzing and looking for anomalies in the data continuously received from traffic and from vehicle probes in the road (surface street) and freeway network 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 incidents.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:

- (a)'incident_analysis_data';
- (b)'current_road_network_use';
- (c)'traffic_image_data';
- (d)'unusual_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'static_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above unsolicited inputs being received:

- (a)'possible_detected_incidents'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) run whenever any of the unsolicited data flows listed above is received;
- (b) analyze the unsolicited data in (c) and look for any anomalies which indicate the traffic is not flowing as expected by reference to an internal historical data store - see (c) below;
- (c) as part of (b) build up an internal store of data which contains a profile of the historical traffic flows, congestion, etc. to act as reference data for the analysis process;

- (d) when anomalies in the traffic flow are detected in (b), report them as possible incidents using the solicited output data flow shown above in (a);
- (e) analyze the unsolicited input data in (a) and (b) above and look for signs of congestion in the traffic flows, e.g. low flow and high occupancy, long journey times;
- (f) when congestion is detected, send the solicited data flow in (b) to other parts of the Manage Traffic function so that further data can be obtained;
- (g) use the solicited input data in (a) to provide the relative location of possible incidents and congestion, and also to relate the actual unsolicited data in a special sense.

PROCESS SPECIFICATIONS

Page 68 of 475

1.3.1.2 Maintain Static Data for Incident Management

Input Flows

static_data_for_incident_management
static_data_request
supply_incident_static_data

Output Flows

current_incident_static_data

Description:

Overview: This process shall be responsible for maintaining the store of static data used by another process within the Manage Incidents facility. The static data shall have been received by this process from another process within the Manage Traffic function, and it shall be possible for that process to request a copy of the current static data.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'static_data_request';
- (b)'supply_incident_static_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'static_data_for_incident_management';
- (b)'current_incident_static_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) run when the either of the unsolicited data flows shown above is received;
- (b) when the first unsolicited data flow is received, load the contents of the store of static data into the second of the solicited data flows and send it to the Plan System Deployment function;
- (c) when the second unsolicited data flow is received, load the contents into the store of static data, overwriting any data already present;
- (d) when carrying out (c) make sure that read access to the store of static data by other processes is prevented;
- (e) be responsible for the management of the data in the store of static data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 69 of 475

1.3.1.3 Process Traffic Images

Input Flows

ft_traffic_images
incident_video_image_control

Output Flows

incident_video_image
traffic_image_data

Description:

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

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:

- (a)'ft-traffic_images'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

(a)'traffic_image_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) transform the data in (a) into a form in which it can be sent for analysis by another process;
- (c) send the data generated in (b) the data analysis process using the solicited output flow shown above;
- (d) at the same time as the data in (c) is output, generate the incident image data flow and send it to the traffic operations personnel interface process;
- (e) when the video camera control data flow is received, implement the data it contains to effect the required changes to the closed circuit television (cctv) system operational parameters.

PROCESS SPECIFICATIONS

Page 70 of 475

1.3.2.1 Store Possible Incident Data

Input Flows

fcm_incident_information
fep_event_information
fws_current_weather
fws_predicted_weather
logged_hazmat_route
media_incident_data_updates
pollution_incident
possible_detected_incidents

Output Flows

possible_incident_data_update
possible_incidents

Description:

Overview: This process shall be responsible for receiving 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 (related to the source of the data, time of its detection, etc.) to that data.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'logged_hazmat_route';
- (b)'fcm-incident_information';
- (c)'fep-event_information';
- (d)'fws-current_weather';
- (e)'fws-predicted_weather';
- (f)'pollution_incident';
- (g)'possible_detected_incidents';
- (h)'media_incident_data_updates'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'possible_incident_data_update';
- (b)'possible_incidents'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) run whenever any of the unsolicited data inputs listed above is received;
- (b) be capable of receiving the input data in a variety of formats and converting it into a single format suitable for use with the store of possible incidents data;
- (c) when possible incident data is being stored a level of confidence must be attached to it so that the accuracy of the data can be rated according to its source;
- (d) be responsible for the management of the data in the store of possible incidents data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 71 of 475

1.3.2.2 Review and Classify Possible Incidents**Input Flows**

incident_details
operations_incident_data_updates
possible_incident_data_update
possible_incidents
request_possible_incidents_data

Output Flows

current_incidents_new_data
incident_data_update
incident_details_request
possible_incidents_data_output
predicted_incident_data
predicted_incidents
predicted_incidents_new_data
tcm_incident_confirmation
tcm_request_incident_change
tep_event_confirmation

Description:

Overview: This process shall be responsible for re-classifying incidents loaded into the store of possible incidents as either current or predicted incidents. The process shall carryout this operation automatically on a regular basis, but shall report any incidents that it is unable to re-classify to the traffic operations personnel for manual re-classification. The process shall enable the traffic operations personnel to request a list of possible incidents and carry out the complete re-classification process manually.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'request_possible_incidents_data';
- (b)'incident_details';
- (c)'possible_incidents_data_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'operations_incident_data_updates';
- (b)'incident_details';
- (c)'possible_incidents'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'incident_details_request';
- (b)'possible_incidents_data_output';
- (c)'predicted_incidents';
- (d)'predicted_incident_data';
- (e)'predicted_incidents_new_data';
- (f)'incident_data_update'.
- (g)'tcm-incident_confirmation';
- (h)'tcm-request_incident_change';
- (i)'tep-event_confirmation';
- (j)'current_incidents_new__data'.

Unsolicited Output Processing: This process shall provide the following output data flow if no input is received in the'incident_details'data flow for a configurable period:

- (a)'incident_details_request'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) run every time any of the unsolicited data flows described above is received;
- (b) be capable of automatically determining which possible incidents can be converted into real incidents and further classifying them as predicted or current incidents;
- (c) the incident classification process shall use the level of confidence data attached to each set

of possible incident data;

(d) if the classification cannot be achieved automatically, send the data to the Traffic Operations Personnel via the third unsolicited output data flow, for manual classification;

(e) where necessary change the data for a possible incident into the standard form (see below), adding in any missing fields if necessary, and adding in the traffic impact data field;

User Service Requirements:

it into the predicted or current incidents

data stores, deleting it from the store of possible incidents, sending data flows to activate the

USR = 1.0; process responsible for reviewing either predicted or current incidents, and also sending the

USR = 1.7; appropriate message to other parts of ITS;

USR = 1.7.0; (g) if necessary updating the data retrieved from the store of possible_incidents for a possible

USR = 1.7.4; incident sent in from the Construction and Maintenance terminator so that it takes place at time(s)

that cause the minimum of impact to traffic and returning the amended data to the terminator via the

Output Flow Dynamics Assumptions:

second solicited output data flow;

(h) new data read from the store of possible_incidents which is found to complement data already

current_incidents_new_data = 12/(60*60); in the predicted or current incidents data stores, will be discarded, except that any additional

incident_details_request = 5/(60*60); data items in the new data shall be loaded into the appropriate data store for the incident that is

possible_incidents_data_output = 5/(60*60); already recorded;

predicted_incidents = 12/(60*60); (i) be responsible for the management of the data in the predicted and current incidents data stores,

predicted_incident_data = 6/(60*60); using the appropriate mechanism(s) such as RDBMS, for storing the data.

incident_data_update = 12/(60*60);

predicted_incidents_new_data = 12/(60*60);

tcm-incident_confirmation = 1/(60*60*24)*INCIDENTS;

tcm-request_incident_change = 1/(60*60*24)*INCIDENTS;

tep-event_confirmation = 1/(60*60);

PROCESS SPECIFICATIONS

PROCESS SPECIFICATIONS

Page 73 of 475

Page 74 of 475

1.3.2.3 Review and Classify Predicted Incidents

Input Flows

current_incidents_data

current_incidents_request

incident_data_update

incident_response_status

predicted_incidents_data

reclassify_incidents

Output Flows

current_incident_data

current_incidents

current_incidents_data_output

current_incidents_data_request

current_incidents_data_update

predicted_incident_data_for_vehicle_signage

request_predicted_incidents_data

Description:

Overview: This process shall be responsible for automatically re-classifying incidents from predicted to current. It shall carryout the re-classification process either upon receiving notice that the store of predicted incidents has been updated, or the time has been reached at which a predicted incident becomes a current incident, or in the absence of any other input, on a regular basis. The process shall request details of predicted incidents 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 predicted and current incidents to other ITS functions, and details of any new predicted incidents to the process responsible for the output

of data to roadside broadcast units.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'reclassify_incidents';
- (b)'incident_data_update';
- (c)'current_incidents_request';
- (d)'incident_response_status'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'predicted_incidents_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'current_incident_data';
- (b)'current_incidents_data_output';
- (c)'current_incidents';
- (d)'current_incidents_data_request';
- (e)'current_incidents_data_update';
- (f)'predicted_incident_data_for_vehicle_signage'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for the unsolicited input data flows listed above;
- (b) carrying out the incident re-classification process on receipt of either of the first, second or fourth unsolicited data flow listed above, or when predicted incidents are expected to become current, or in the absence of any inputs on a regular (time interval) basis;
- (c) when the second unsolicited input data flow indicates that a new predicted incident has been found, send the incident details to the process that outputs data to roadside broadcast processes, using the last solicited output data flow listed above;
- (d) when the second unsolicited input data flow indicates that a new current incident has been found, request the current incidents data and output that for the new incident to the process responsible for providing incident responses;
- (e) be capable of automatically re-classifying incidents from predicted to current based on the time at which the incident is expected to take place;
- (f) when an incident is re-classified from predicted to current, send out the data for the new current incident to other parts of ITS, and the data flow to activate the process responsible for responding to incidents.

PROCESS SPECIFICATIONS

Page 75 of 475

PROCESS SPECIFICATIONS

Page 76 of 475

1.3.2.4 Provide Predicted Incidents Store Interface

Input Flows

other_predicted_incidents
predicted_incidents_new_data
predicted_incidents_store
request_local_predicted_incidents_data
request_predicted_incidents_data

Output Flows

predicted_incidents_data
predicted_incidents_data_output
predicted_incidents_local_data
request_other_predicted_incidents_data

Description:

Overview: This process shall be responsible for providing the interface to, and managing the use of the store containing details of predicted incidents. The process shall enter details of all new predicted incidents 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 predicted incidents from within the local Manage Incidents facility, and from similar facilities in other Traffic Management Centers (TMC's). When requested, the process shall also be able to provide details of its predicted incidents to the Manage Incidents facilities in other TMC's.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'predicted_incidents_new_data';
- (b)'request_predicted_incidents_data';
- (c)'request_local_predicted_incidents_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval:

- (a)'predicted_incidents_store'- which contains data retrieved from a data store;
- (b)'other_predicted_incidents'- which contains data received from another process.

Unsolicited Output Processing: This process shall provide the following output flow regardless of any input flows that are received:

- (a)'request_other_predicted_incidents_data';

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'predicted_incidents_data';
- (b)'predicted_incidents_data_output';
- (c)'predicted_incidents_local_data';
- (d)'predicted_incidents_store'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) the data shall be stored in such a manner that retrieval and deletion of data is a quick and easy process;
- (c) each time data about a new predicted incident is loaded into the store, the process shall also pass that data on to other parts of the Manage Traffic function through the output of the predicted incidents data flow;
- (d) if the incident may affect traffic outside the local geographic or jurisdictional area served by the function, then the data about the incident shall be sent to other TMC's using the appropriate data flow;
- (e) when initially run, request data on predicted incidents that may affect local traffic from other TMC's using the appropriate data flow;
- (f) when data about predicted incidents in geographic or jurisdictional areas served by other TMC's is received it shall be loaded into the store of predicted incidents and processed as though the incident(s) had just occurred;
- (g) when a request for local predicted incident data is received, only data on those predicted incidents that may affect traffic outside the geographic or jurisdictional area served by the function shall be retrieved from the data store and sent to the requesting TMC in the appropriate data flow;
- (h) the process shall be responsible for the maintenance of the store of predicted incidents data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 77 of 475

PROCESS SPECIFICATIONS

Page 78 of 475

1.3.2.5 Provide Current Incidents Store Interface

Input Flows

current_incidents_data_request
current_incidents_data_update
current_incidents_new_data
current_incidents_store
other_current_incidents
request_local_current_incidents_data

Output Flows

current_incidents_data
current_incidents_local_data
request_other_current_incidents_data

Description:

Overview: This process shall be responsible for providing the interface to, and managing 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 Centers (TMC's). When requested, the process shall also be able to provide details of its current incidents to the Manage Incidents facilities in other TMC's.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'current_incidents_data_update';
- (b)'current_incidents_data_request';
- (c)'current_incidents_new_data';
- (d)'arequest_local_current_incidents_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'current_incidents_store'- which contains data retrieved from a data store;
- (b)'other_current_incidents'- which contains data received from another process.

Unsolicited Output Processing: This process shall provide the following output flow regardless of any input flows that are received:

- (a)'request_other_current_incidents_data';

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'current_incidents_data';
- (b)'current_incidents_local_data';
- (c)'current_incidents_store';
- (d)'request_other_current_incidents_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) each time data about a new current incident is received via either the first or fourth unsolicited input data flow listed above, it shall be loaded into the store;
- (c) when initially run, request data on current incidents that may affect local traffic from other TMC's using the appropriate data flow;
- (d) when data about current incidents in geographic or jurisdictional areas served by other TMC's is received it shall be loaded into the store of current incidents and processed as though the incident(s) had just occurred;
- (e) when a request for local current incident data is received, only data on those current incidents that may affect traffic outside the geographic or jurisdictional area served by the function shall be retrieved from the data store and sent to the requesting TMC in the appropriate data flow;
- (f) the process shall be responsible for the maintenance of the store of current incidents data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 79 of 475

PROCESS SPECIFICATIONS

Page 80 of 475

1.3.3 Respond to Current Incidents

Input Flows

current_incidents_data_output
defined_responses_data

Output Flows

current_incident_data_for_vehicle_signage
cv_incident_override
incident_alert
incident_response_clear
incident_response_log
incident_strategy_override
undefined_incident_response
vms_updates_for_highways
vms_updates_for_roads

Description:

Overview: This process shall be responsible for providing the responses to incidents that become current, i.e. active. The response to each type of incident shall be defined for the process in the

store of defined responses data. If the process cannot find a defined response for a particular incident, it shall send the details to the traffic operations personnel so that they can provide an update to the store of defined responses. The process shall output the defined 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'current_incidents_data_output';
- (b)'current_incident_data_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'defined_responses_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'current_incident_data_for_vehicle_signage';
- (b)'incident_alert';
- (c)'incident_response_clear';
- (d)'incident_response_log';
- (e)'incident_strategy_override';
- (f)'cv_incident_override';
- (g)'undefined_incident_response';
- (h)'vms_updates_for_highways';
- (i)'vms_updates_for_roads'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) analyze the current incident data against the data in the store of defined responses to determine the appropriate response;
- (c) generate the appropriate solicited output flows listed above as a result of determining the appropriate defined response to an incident;
- (d) generate the appropriate clearance data in the solicited output flows listed above when the duration of an incident expires;
- (e) if a defined response is not found for any incident, then the process shall send data about the incident to the Provide Operator Interfaces for Incidents facility and take no further action;
- (f) the process shall be responsible for the maintenance of the store of data in the log of incident responses, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 81 of 475

PROCESS SPECIFICATIONS

Page 82 of 475

1.3.4.1 Retrieve Incident Data

Input Flows

confirm_incident_data_output
current_incidents
map_data_for_incident_display
possible_incidents_data_output
predicted_incidents_data_output
request_incident_media_data
request_incident_operations_data

Output Flows

current_incidents_request
incident_data_output
request_possible_incidents_data
retrieved_incident_media_data
retrieved_incident_operations_data

Description:

Overview: This process shall be responsible for retrieving incident from the stores of predicted 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 go for output by the media system.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'request_incident_operations_data';
- (b)'request_incident_media_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'current_incidents';
- (b)'predicted_incidents_data';
- (c)'possible_incidents_data';
- (d)'confirm_incident_data_output';
- (e)'map_data_for_incident_display'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'incident_data_output';
- (b)'current_incidents_request';
- (c)'request_possible_incidents_data';
- (d)'retrieved_incident_media_data';
- (e)'retrieved_incident_operations_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input data flows listed above;
- (b) when either of the flows in (a) is received, request the required incident data from the appropriate store interface process using solicited output data flow list above;
- (c) when the appropriate solicited input data flow is received in response to (b), integrate the data from the store of digitized data with the incident data;
- (d) when (c) is completed send the data to the process from which the data flow in (a) was received;
- (e) if the data flow from the media operator in (a) contains a request for the output of data to the media system, load the retrieved incident data into the first solicited output data flow listed above;
- (f) as a result of (e), continuously monitor for receipt of the solicited input data flow listed above that contains confirmation of the output of the data;
- (g) when the data flow in (f) is received send its contents to the media operator interface process.

PROCESS SPECIFICATIONS

Page 83 of 475

PROCESS SPECIFICATIONS

Page 84 of 475

1.3.4.2 Provide Traffic Operations Personnel Incident Data Interface

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_reponses
ftop_request_possible_incidents_data
ftop_update_defined_incident_responses
incident_video_image
possible_defined_responses_output
retrieved_incident_operations_data
undefined_incident_response

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
 ttop_defined_incident_responses_data
 ttop_incident_information_display
 ttop_incident_video_image_output
 ttop_possible_defined_response_output
 ttop_possible_incidents_data
 ttop_undefined_response_details

Description:

Overview: This process shall be responsible for providing 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 defined incident responses, obtain and control incident video image data and manually re-classify incidents from possible to predicted and predicted to current. It shall also output to the personnel incident details to which to defined response currently exists. 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. The process shall enable the visual output to be in hardcopy, or as a display. Where appropriate and/or requested by the traffic operations personnel, the process shall provide the display in the form of an overlay onto a map of the relevant part(s) of the road and freeway network served by the function. The process shall obtain the map in digitized form from a local data store, which it shall request to be updated by another process as and when required.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'ftop-defined_incident_response_data_request';
- (b)'ftop-defined_incident_response_data_update';
- (c)'ftop-incident_camera_action_request';
- (d)'ftop-incident_data_amendment';
- (e)'ftop-incident_information_requests';
- (f)'ftop-output_possible_defined_reponses';
- (g)'ftop-request_possible_incidents_data';
- (h)'ftop-update_defined_incident_responses';
- (i)'undefined_incident_response'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'defined_incident_response_data';

User Service Requirements:

- (b)'retrieved_incident_operations_data'.

USR = 1.0; Solicited Output Processing: This process shall provide the following output flows as a result of

USR = 1.7; the above inputs being received:

USR = 1.7.0; (a)'defined_incident_response_data_request';

USR = 1.7.4; (b)'defined_incident_response_updates';

(c)'defined_incident_response_update_request';

Output Flow Dynamics Assumptions:

(d)'incident_video_image_control';
 defined_incident_response_data_request = $1/(60*60*24)$; (e)'operations_incident_data_updates';
 defined_incident_response_updates = $2/(60*60*24)$; (f)'possible_defined_responses_output_request';
 defined_incident_response_update_request = $1/(60*60*24*7)$; (g)'reclassify_incidents';
 incident_video_image_control = ftop-incident_camera_action_request;
 (h)'request_incident_operations_data';
 operations_incident_data_updates = $4/(60*60)$; (i)'request_incident_map_display_update';
 possible_defined_responses_output_request = $1/(60*60*24)$; (j)'ttop-
 defined_incident_responses_data';
 reclassify_incidents = $4/(60*60)$; (k)'ttop-incident_information_display';

request_incident_operations_data = 20/(60*60)*TRAFFIC_OPS; (l)'ttop-incident_video_image_output';
request_incident_map_display_update = 2/(60*60*24*7*52); (m)'ttop-possible_defined_response_output';
ttop-defined_incident_responses_data = 1/(60*60); (n)'ttop-possible_incidents_data';
ttop-incident_information_display = 1/(60*60); (o)'ttop-undefined_response_details'.
ttop-possible_defined_response_output = 1/(60*60*24);
ttop-possible_incidents_data = 1/(60*60); Functional Requirements: This process shall meet the following functional requirements:
ttop-undefined_response_details = undefined_incident_response; (a) continuously monitor the input data flows and provide acknowledgement of receipt of those from
ttop-incident_video_image_output = 1; Traffic Operations Personnel within 1 second;
(b) be capable of accepting input from Traffic Operations Personnel in audio or other forms, where the latter may comprise input from any combination of keyboards or other forms of push-button devices, pointing devices, etc.;
(c) be capable of carrying out its own verification of input data received from Traffic Operations Personnel and generating the correct solicited output data flow as a result of data being received;
(d) as part of the output generation process, checking for data out of range, missing or spurious values and requesting re-input where required;
(e) be capable of simultaneously handling a large number of independent input/output data channels, i.e. supporting very many Traffic Operations Personnel;
(f) providing all output to Traffic Operations Personnel in a form that is readily understood by a human operator and which may be in audio or visual form, with the latter being available in a variety of formats, e.g. displays, or hardcopy (paper) output;
(g) only generate the outputs listed above as a result of receiving inputs from the Traffic Operations Personnel or the other processes;
(h) when the request for changes to the parameters affecting the operation of the closed circuit television (cctv) system responsible for providing images of incidents is received from the traffic operations personnel, generate the data flow to the image processing facility;
(i) when video images of incidents are received, output them to the traffic operations personnel; the use of the digitized map display shall be automatic and shall be at a resolution best suited to the quantity and scope of data being displayed, i.e. the map shall be to the largest possible scale.

PROCESS SPECIFICATIONS

Page 85 of 475

PROCESS SPECIFICATIONS

Page 86 of 475

PROCESS SPECIFICATIONS

Page 87 of 475

1.3.4.3 Provide Media Operator Incident Data Interface

Input Flows

fmo_incident_details
fmo_incident_information_output_request
fmo_incident_information_request
fmo_media_operator_identity
retrieved_incident_media_data

Output Flows

media_incident_data_updates
request_incident_media_data
tmo_incident_information

Description:

Overview: This process shall be responsible for providing the interface between the media operator and the Manage Incidents facility. It shall enable the operator to request details of predicted and current incidents, to send information about incidents for output by the media system and to provide raw input data for possible incidents. The process shall support inputs from the media operator in both manual and audio form, and shall provide its outputs in audible and visual forms. The process shall enable the visual output to be in hardcopy, or as a display which is shown as an overlay on a map of the area to which the incident relates.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fmo-incident_details';
- (b)'fmo-media_operator_identity';
- (c)'fmo-incident_information_output_request';

(d)'fmo-incident_information_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

(a)'retrieved_incident_media_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

(a)'media_incident_data_updates';

(b)'tmo-incident_information';

(c)'request_incident_media_data'.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor the input data flows and provide acknowledgement of receipt of those from the Media Operator within 1 second;

(b) be capable of accepting input from the Media Operator in audio or other forms, where the latter may comprise input from any combination of keyboards or other forms of push-button devices, pointing devices, etc.;

(c) be capable of carrying out its own verification of input data received from the Media Operator and generating the correct solicited output data flow as a result of the input data being received;

(d) as part of the output generation process, carrying out checks for data out of range, missing or spurious values and requesting re-input where necessary;

(e) using the first solicited output data flow to send data on a possible incident when this data is received from the Media Operator;

(f) be capable of simultaneously handling a large number of independent input/output data channels, i.e. supporting very many Media Operators;

(g) providing all output to the Media Operator in a form that is readily understood by a human operator and which may be in audio or visual form, with the latter being available in a variety of formats, e.g. displays, or hardcopy (paper) output;

(h) only generate the outputs listed above as a result of receiving inputs from the Media Operator or the other processes;

(i) the use of the digitized map display shall be automatic and shall be at a resolution best suited to the quantity and scope of data being displayed, i.e. the map shall be to the largest possible scale.

PROCESS SPECIFICATIONS

Page 88 of 475

PROCESS SPECIFICATIONS

Page 89 of 475

1.3.4.4 Update Incident Display Map Data

Input Flows

fmup_incident_display_update

request_incident_map_display_update

Output Flows

map_data_for_incident_display

tmup_request_incident_display_update

Description:

Overview: This process shall be responsible for providing updates to the store of digitized map data used as the background for 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 specialist data supplier or some other appropriate data source, on receiving an update request from the traffic operations personnel interface process within the Manage Incidents facility.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

(a)'request_incident_map_display_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to external functions:

(a)'fmup-incident_display_update'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

(a)'tmup-request_incident_display_update';

(b)'map_data_for_incident_display'.

Functional Requirements: This process shall meet the following requirements:

- (a) continuously monitor for the receipt of the unsolicited data flow shown above;
- (b) when the data flow in (a) is received, generate the first solicited output data flow shown above and continuously monitor for receipt of the solicited input data flow shown above;
- (c) when the flow in (b) is received, output the second solicited output data flow shown above;
- (d) be capable of receiving the input data in a variety of formats and converting it into a single format suitable for use with the store of digitized map data;
- (e) be responsible for the management of the data in the store of digitized map data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 90 of 475

1.3.4.5 Provide Media System Incident Data Interface

Input Flows

incident_data_output

Output Flows

confirm_incident_data_output

tm_incident_information

Description:

Overview: This process shall be responsible for providing the interface between the media system and the Manage Incidents facility within the Manage Traffic function. The process shall only enable this interface for the output of incident data as requested by the media operator. It shall only return confirmation that the data has been output.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

(a)'incident_data_output'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

(a)'tm_incident_information';

(b)'confirm_incident_data_output'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) run when the unsolicited data shown above is received;
- (b) convert the unsolicited data into a form that can be output either by synthesized voice and/or visual display, depending on the output facilities available to the Media System, and output it in the first solicited output data flow;
- (c) provide confirmation that the output has been transmitted through the second solicited output data flow.

PROCESS SPECIFICATIONS

Page 91 of 475

1.3.5 Manage Possible Defined Responses Store

Input Flows

defined_incident_response_update_request

possible_defined_responses

possible_defined_responses_data

possible_defined_responses_output_request

Output Flows

defined_incident_response_changes

possible_defined_responses_output

Description:

Overview: This process shall be responsible for managing the store of data containing possible defined responses to incidents used within the Manage Incidents facility. These responses shall be those that another process within the facility has found to be worthy of inclusion in the store of defined 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 subsequent transfer to the process that manages the store of defined incident responses that are actually used by other processes in the Manage Incidents facility.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'defined_incident_response_update_request';
- (b)'possible_defined_responses_data';
- (c)'possible_defined_responses_output_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval:

- (a)'possible_defined_responses

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'defined_incident_response_changes';
- (b)'possible_defined_responses_output'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the request is received for the transfer of a possible response to the process managing the store of defined responses, the data shall be deleted from the store of possible responses once the transfer has been successfully completed;
- (c) the process shall be responsible for the maintenance of the store of possible defined responses data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 92 of 475

1.3.6 Manage Defined Incident Response Data

Input Flows

- defined_incident_response_changes
- defined_incident_response_data_request
- defined_incident_response_updates
- defined_responses_data

Output Flows

- defined_incident_response_data

Description:

Overview: This process shall be responsible for managing data held in the store of defined 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 defined responses in response to requests from the traffic operations personnel, and shall also update the store with new responses received from the process that manages the store of possible defined responses.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:

- (a)'defined_incident_response_data_request';
- (b)'defined_incident_response_changes';
- (c)'defined_incident_response_updates'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'defined_responses_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above unsolicited inputs being received:

- (a)'defined_responses_data';
- (b)'defined_incident_response_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) run whenever any of the unsolicited or solicited data flows shown above is received;
- (b) if the data flow in (a) is a request for data, retrieve it from the store of defined responses and return it to the requesting process in the second solicited output flow listed above;
- (c) if the data flow in (a) contains new data for the store of defined responses, load it into the store using the first solicited output data flow shown above;
- (d) be responsible for the management of the store of defined responses, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 93 of 475

1.3.7 Analyze Incident Response Log

Input Flows

incident_response_log

Output Flows

possible_defined_responses_data

Description:

Overview: This process shall be responsible for periodically analyzing the data in the log of incident responses within the Manage Incidents facility of the Manage Traffic functions. The process shall analyze the log in such a way that possible standard defined incident responses can be determined from the data it contains. Any such possible standard responses that are determined shall be passed by this process to the process that manages the store of possible defined responses.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval:
(a)'incident_response_log'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'possible_defined_responses_data'.

Functional Requirements: This process shall meet the following functional requirements:
(a) run periodically (at least once per day) and analyze the data in the log of incident responses data to determine any response patterns that could be used as standards for the responses to particular types of incident;
(b) pass any such possible standard defined responses as may be determined from the data to the process that enables them to be stored and reviewed by the traffic operations personnel.

PROCESS SPECIFICATIONS

Page 94 of 475

1.4.1 Provide Traffic Operations Personnel Demand Interface

Input Flows

demand_forecast_data
demand_forecast_result
demand_input_data
demand_management_result
demand_policy_data
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

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

Description:

Overview: This process shall be responsible for providing 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 personnel to have access to the data used as input by the demand forecasting process and the results of that process, to request that the input data be updated, to request that the demand forecasting process runs, and to run the process that implements the results. 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. The process shall enable the visual output to be in hardcopy, or as a display. Where appropriate and/or requested by the

traffic operations personnel, the process shall provide the display in the form of an overlay onto a map of the relevant part(s) of the road and freeway network served by the Manage Traffic function. The process shall obtain the map in digitized form from a local data store, which it shall request to be updated by another process as and when required.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'ftop-demand_policy_information_request';
- (b)'ftop-demand_policy_updates';
- (c)'ftop-demand_data_update_request';
- (d)'ftop-demand_data_request';
- (e)'ftop-demand_forecast_request';
- (f)'ftop-demand_policy_activation';.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'demand_management_result';
- (b)'demand_input_data';
- (c)'demand_policy_data';
- (d)'demand_forecast_data';
- (e)'demand_forecast_result';
- (f)'map_data_for_demand_display'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'ttop-demand_policy_information';
- (b)'ttop-demand_data';
- (c)'ttop-demand_forecast_data';
- (d)'ttop-demand_policy_activation_result';
- (e)'ttop-demand_forecast_result';

User Service Requirements:

- (g)'demand_forecast_request';
- USR = 1.0; (h)'demand_management_activate';
- USR = 1.8; (i)'request_demand_display_update'.
- USR = 1.8.0;
- USR = 1.8.1;

Functional Requirements: This process shall meet the following functional requirements:
USR = 1.8.1.1; (a) continuously monitor the input data flows and provide acknowledgement of receipt of those from

Traffic Operations Personnel within 1 second;

Output Flow Dynamics Assumptions:

(b) be capable of accepting input from traffic operations personnel in audio or other forms, where the latter may comprise input from any combination of keyboards or other forms of push-button demand_data_update_request = ftop-demand_data_update_request; devices, pointing devices, etc.; demand_forecast_request = ftop-demand_forecast_request; (c) be capable of carrying out its own verification of input data received from traffic operations demand_management_activate = ftop-demand_policy_activation; personnel and generating the correct solicited output data flow as a result of the input data being demand_policy_data = ftop-demand_policy_updates; received; request_demand_display_update = 2/((60*60*24*7*52); (d) as part of the output generation process, carrying out checks for data out of range, missing or ttop-demand_policy_information = ftop-demand_policy_information_request; spurious values and requesting re-input where necessary; ttop-demand_data = ftop-demand_data_request; (e) be capable of simultaneously handling a large number of independent input/output data channels, ttop-demand_forecast_data = ftop-demand_data_request; i.e. supporting very many traffic operations personnel; ttop-demand_policy_activation_result = ftop-demand_policy_activation; (f) providing all output to traffic operations personnel in a form that is readily understood by a human operator and which may be in audio or visual form, with the latter being available in a variety

- of formats, e.g. displays, or hardcopy (paper) output;
- (g) only generate the outputs listed above as a result of receiving inputs from the traffic operations personnel or the other processes;
- (h) periodically automatically generate the request for new digitized map data shown above in the last solicited output;
- (i) the use of the digitized map display shall be automatic and shall be at a resolution best suited to the quantity and scope of data being displayed, i.e. the map shall be to the largest possible scale.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 95 of 475
Page 96 of 475

1.4.2 Collect Demand Forecast Data

Input Flows

current_other_routes_use
current_transit_routes_use
demand_data_update_request
From_Weather_Service
parking_lot_charge_details
pollution_state_data
toll_price_details
traffic_data_for_demand
transit_fare_details
transit_running_data_for_demand
transit_services_for_demand
unusual_congestion

Output Flows

demand_input_data
parking_lot_charge_request
pollution_state_data_request
toll_price_request
traffic_data_demand_request
transit_conditions_demand_request
transit_fare_request
transit_services_demand_request

Description:

Overview: This process shall be responsible for collecting 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 function. It shall load all the data that it receives in a consistent format into the input store used by the demand forecasting process.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'unusual_congestion';
- (b)'prediction_data';
- (c)'current_transit_routes_use';
- (d)'current_road_network_use';
- (e)'From_Weather_Service';
- (f)'demand_data_update_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes:

- (a)'parking_lot_charge_details';
- (b)'pollution_state_data';
- (c)'price_data';
- (d)'toll_price_details';
- (e)'transit_fare_details';
- (f)'transit_running_data_for_demand';
- (g)'transit_services_for_demand';
- (h)'traffic_data_for_demand'.

Unsolicited Output Processing: This process shall periodically generate the following output flows to other processes and functions within ITS and the local store of input data:

- (a)'pollution_state_data_request';
- (b)'parking_lot_charge_request';
- (c)'toll_price_request';
- (d)'transit_conditions_demand_request';
- (e)'transit_services_demand_request';
- (f)'traffic_data_demand_request';
- (g)'transit_fare_request'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

User Service Requirements: (a)'demand_input_data'.

USR = 1.0; Functional Requirements: This process shall meet the following functional requirements:

- USR = 1.8; (a) run at five minute intervals and generate the unsolicited outputs listed above, unless requested
- USR = 1.8.0; to run by the unsolicited input in (f) above;
- USR = 1.8.1; (b) when running for whatever reason, scan all the unsolicited inputs listed above and collect the
- USR = 1.8.1.1; data that they are currently providing;
- USR = 1.8.1.2; (c) when all inputs have been obtained produce the solicited output shown above to load the collected
- USR = 1.8.1.3; data into the store of demand input data;
- USR = 1.8.1.4; (d) be capable of receiving the input data in a variety of formats and converting it into a single
- USR = 1.8.1.5; format suitable for use with the store of demand input data;
- USR = 1.8.2; (e) be responsible for the management of the data in the store of demand input data, using the
- USR = 1.8.2.1; appropriate mechanism(s) such as RDBMS, for storing the data.
- 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 = 1.8.2.5;
- USR = 1.8.2.6;
- USR = 1.8.2.7;
- USR = 1.8.2.8;
- USR = 1.8.2.9;
- 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;
- USR = 3.1.5.3;

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 97 of 475
Page 98 of 475
Page 99 of 475

1.4.3 Update Demand Display Map Data

Input Flows

fmup_demand_display_update
request_demand_display_update

Output Flows

map_data_for_demand_display

tmup_request_demand_display_update

Description:

Overview: This process shall be responsible for providing updates to the store of digitized map data used as the background 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 data supplier or some other appropriate data source, on receiving an update request from the traffic operations personnel interface process within the Manage Travel Demand facility.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'request_demand_display_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to external functions:
(a)'fmup-demand_display_update'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'tmup-request_demand_display_update';
(b)'map_data_for_demand_display'.

Functional Requirements: This process shall meet the following requirements:
(a) continuously monitor for the receipt of the unsolicited data flow shown above;
(b) when the data flow in (a) is received, generate the first solicited output data flow shown above and continuously monitor for receipt of the solicited input data flow shown above;
(c) when the flow in (b) is received, output the second solicited output data flow shown above;
(d) be capable of receiving the input data in a variety of formats and converting it into a single format suitable for use with the store of digitized map data;
(e) be responsible for the management of the data in the store of digitized map data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 100 of 475

1.4.4 Implement Demand Management Policy

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

Description:

Overview: This process shall be responsible for implementing the traffic and travel demand policy data produced by the demand forecasting process in the Manage Travel Demand facility of the Manage Traffic function. The new policy shall be implemented in such a way that it can influence the demand from travelers for all types of services provided by ITS functions. Where appropriate this shall lead to a change in the modal split of travelers' journeys away from the use of the private motor car by single occupants towards other forms of transport such as ridesharing, demand responsive transit, and regular transit services. The process shall when required, request changes to transit services, and/or the charges for tolls, and/or the use of parking lot spaces. It shall communicate the results of its policy implementation to the process that provides the interface to the traffic operations personnel.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'demand_management_activate'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'demand_forecast_data';
- (b)'parking_lot_charge_change_response';
- (c)'toll_price_changes_response';
- (d)'transit_services_changes_response'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'ahs_control_data';
- (b)'demand_management_result';
- (c)'demand_overrides';
- (d)'parking_lot_charge_change_request';
- (e)'toll_price_changes_request';
- (f)'transit_services_changes_request'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited data flow shown above;
- (b) when the flow in (a) is received, send data to other Manage Traffic facilities and ITS functions using the solicited output data flows listed above;
- (c) be capable of interpreting the contents of the store of demand policy data in a way that the outputs that are generated in (d) are readily understood by the receiving processes;
- (b) provide continuous feedback of the responses to the flows in (b) using the demand management result solicited output data flow shown above;
- (e) retrieve the data from the store of demand forecast data using the most appropriate mechanism(s) such as RDBMS.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 101 of 475
Page 102 of 475

1.4.5 Calculate Forecast Demand

Input Flows

demand_forecast_request
demand_input_data
demand_policy_data

Output Flows

demand_forecast_data
demand_forecast_result

Description:

Overview: This process shall be responsible for providing a forecast of traffic and travel demand in the geographic area served by the Manage Traffic function to which 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. The process shall ensure that the demand forecast it produces is able to change the way that services are provided by ITS functions so that travelers can make more time and energy efficient journeys. The travel demand shall be influenced by the output from the process in ways that where appropriate, will change the modal split of travelers' journeys away from the current predominance of the use of the private car by single occupants.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'demand_forecast_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval:

- (a)'demand_input_data';
- (b)'demand_policy_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'demand_forecast_data';

(b)'demand_forecast_result'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited data flow shown above;
- (b) when the data flow in (a) is received, use appropriate algorithms etc. to calculate future traffic and travel demand patterns across all the available modes of transportation using the stores of demand input and demand policy data;
- (c) provide appropriate output continuously on the progress of the calculation of the new traffic and travel demand forecast in the demand forecast result data flow shown above;
- (d) be capable of abandoning the calculation of forecast demand if this proves impossible to achieve, i.e. the process shall not 'lock-up';
- (e) be responsible for the management of the data in the store of demand forecast data, using the most appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 103 of 475

PROCESS SPECIFICATIONS

Page 104 of 475

1.5.1 Provide Traffic Operations Personnel Pollution Data Interface

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

Description:

Overview: This process shall be responsible for providing 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 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. Where appropriate and/or requested by the traffic operations personnel, the process shall provide the display in the form of an overlay onto a map of the relevant part(s) of the road and freeway network served by the Manage Traffic function. The process shall obtain the map in digitized form from a local data store, which it shall request to be updated by another process as and when required.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'ftop-pollution_data_information_request';
- (b)'ftop-pollution_parameter_updates'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'map_data_for_pollution_display';
- (b)'pollution_reference_data_output';
- (c)'pollution_state_data_output'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'pollution_reference_data_request';
- (b)'pollution_reference_data_update';
- (c)'pollution_state_data_output_request';
- (d)'ttop-pollution_data_display';
- (e)'request_pollution_map_display_update'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor the input data flows and provide acknowledgement of receipt of those from Traffic Operations Personnel within 1 second;
- (b) be capable of accepting input from Traffic Operations Personnel in audio or other forms, where the latter may comprise input from any combination of keyboards or other forms of push-button devices, pointing devices, etc.;
- (c) be capable of carrying out its own verification of input data received from Traffic Operations personnel and generating the correct solicited output data flow as a result of input data being received;
- (d) as part of the output generation process, carrying out checks for data out of range, missing or spurious values and requesting re-input where necessary;
- (e) be capable of simultaneously handling up to 10 independent input/output data channels, i.e. supporting up to 10 Traffic Operations Personnel;
- (f) providing all output to Traffic Operations Personnel in a form that is readily understood by a human operator and which may be in audio or visual form, with the latter being available in a variety of formats, e.g. displays, or hardcopy (paper) output;
- (g) only generate the outputs listed above as a result of receiving inputs from the Traffic Operations Personnel or the other processes;
- (h) the use of the digitized map display shall be automatic and shall be at a resolution best suited to the quantity and scope of data being displayed, i.e. the map shall be to the largest possible scale;
- (i) be responsible for the management of the data in the store of pollution reference data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

User Service Requirements:

USR = 1.9.1.2.1;
 USR = 1.9.1.2.2;
 USR = 1.9.2.2.1;
 USR = 1.9.2.2.3;

PROCESS SPECIFICATIONS PROCESS SPECIFICATIONS

Page 105 of 475
 Page 106 of 475

1.5.2 Process Pollution Data

Input Flows

fe_area_pollutant_levels
 pollution_state_roadside_collection
 pollution_state_static_acceptance_criteria

Output Flows

pollution_incident
 pollution_state_static_collection
 pollution_state_static_log_data
 wide_area_pollution_data

Description:

Overview: This process shall be responsible for processing 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fe-area_pollutant_levels'
- (b)'pollution_state_static_acceptance_criteria';
- (c)'pollution_state_roadside_collection'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'pollution_state_static_log_data';
- (b)'pollution_incident';
- (c)'pollution_state_static_collection';

(d)'wide_area_pollution_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) every second monitor the unsolicited input data flows shown above;
- (b) process the first unsolicited data flow and analyze the pollutant levels it is providing;
- (c) use the data obtained from the second unsolicited input data flow to determine whether or not the levels of the pollutants exceed that which is considered either safe and/or desirable;
- (d) continuously output a digitized values of the current pollution levels to the process that manages the store of pollution data state via the third solicited output data flow, and every five
- (5) minutes to other parts of ITS via the fourth solicited output data flow;
- (e) if the pollution level(s) are found to be unsafe, declare an incident by sending the first unsolicited data flow to the Manage Incidents facility, isolating it to individual sectors within the geographic area covered by ITS;
- (f) periodically the process shall send the current pollution levels to the log of pollution data using the first solicited output data flow.

PROCESS SPECIFICATIONS

Page 107 of 475

1.5.3 Update Pollution Display Map Data

Input Flows

fmup_pollution_display_update
request_pollution_map_display_update

Output Flows

map_data_for_pollution_display
tmup_request_pollution_display_update

Description:

Overview: This process shall be responsible for providing updates to the digitized map data used as the background for displays of pollution data produced by processes in the Manage Emissions facility of the Manage Traffic 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 traffic operations personnel interface process within the Manage Emissions facility.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'request_pollution_display_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to external functions:

- (a)'fmup-pollution_display_update'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'tmup-request_pollution_display_update';
- (b)'map_data_for_pollution_display'.

Functional Requirements: This process shall meet the following requirements:

- (a) continuously monitor for the receipt of the unsolicited data flow shown above;
- (b) when the data flow in (a) is received, generate the first solicited output data flow shown above and continuously monitor for receipt of the solicited input data flow shown above;
- (c) when the flow in (b) is received, output the second solicited output data flow shown above;
- (d) be capable of receiving the input data in a variety of formats and converting it into a single format suitable for use with the store of digitized map data;
- (e) be responsible for the management of the data in the store of digitized map data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 108 of 475

1.5.4 Manage Pollution State Data Store

Input Flows

pollution_state
pollution_state_data_output_request
pollution_state_data_request
pollution_state_static_collection
pollution_state_vehicle_collection

Output Flows

pollution_state_data
pollution_state_data_output

Description:

Overview: This process shall be responsible for managing 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'pollution_state_static_collection';
- (b)'pollution_state_data_request';
- (c)'pollution_state_data_output_request';
- (d)'pollution_state_vehicle_collection'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval:

- (a)'pollution_state'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'pollution_state_data';
- (b)'pollution_state_data_output'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) run whenever any of the unsolicited input data flows shown above are received;
- (b) when either the first or last of the unsolicited input flows is received, store the data that it contains in the store of pollution state data;
- (c) when either of the second or third of the unsolicited input flows is received, read the contents of the store of pollution state data and send it to the requesting process using the appropriate solicited output flow shown above;
- (d) be responsible for the management of the data in the store of pollution state data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 109 of 475

1.5.5 Process Vehicle Pollution Data

Input Flows

fbv_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

Description:

Overview: This process shall be responsible for obtaining pollution data about individual vehicles and analyzing 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 violating the acceptable levels of pollution output. When the process determines that a 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 and possible prosecution.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fbv-vehicle_characteristics';

- (b)'ft-vehicle_pollutant_levels';
- (c)'pollution_state_vehicle_acceptance_criteria';
- (d)'vehicle_status_details_for_emissions'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'pollution_state_vehicle_log_data'
- (b)'pollution_state_vehicle_collection';
- (c)'vehicle_pollution_message_for_highways';
- (d)'vehicle_pollution_message_for_roads';
- (e)'vehicle_pollution_alert'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor the unsolicited input data flows shown above;
- (b) process the first two unsolicited data flows from whatever their input form (e.g. analogue) into highly accurate digitized data that shows the levels of various atmospheric pollutants, such as ozone and its precursors, carbon monoxide, nitrous oxide, sulfur dioxide, hydrocarbons, particulates, etc. being produced by a vehicle with a particular identity;
- (c) use the data obtained from the third unsolicited input data flow to determine whether or not the levels of the pollutants produced by the vehicle exceed that which is considered either safe and/or desirable;
- (d) continuously output a digitized values of the current vehicle pollution levels to the process that manages the store of pollution state via the second solicited output data flow as an average for vehicles in a particular roadside location, i.e. removing the vehicle identity;
- (e) if the pollution level(s) produced by a particular vehicle are found to be unsafe, send the third unsolicited data flow to the Provide Device Control facility so that the vehicle's driver can be given visual warning of what the vehicle is doing to the atmospheric pollution levels, and send the same data to the Manage Emergency Service function for enforcement purposes using the

fourth

solicited output data flow;

- (f) data concerning the levels of pollution being detected in vehicles shall be periodically sent to the store that is acting as the log of pollution data using the first solicited output flow, having had all vehicle identity data removed, but vehicle type data retained.

PROCESS SPECIFICATIONS

Page 110 of 475

PROCESS SPECIFICATIONS

Page 111 of 475

1.5.6 Detect Roadside Pollution Levels

Input Flows

fe_roadside_pollutant_levels

Output Flows

pollution_state_roadside_collection

Description:

Overview: This process shall be responsible for processing the local area pollution data being 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 established thresholds for pollution incidents.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fe-area_pollutant_levels';

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'pollution_state_roadside_collection'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) every second monitor the unsolicited input data flows shown above;
- (b) process the first and second unsolicited data flows from whatever their input form (e.g. analogue) into digitized data that shows the levels of various atmospheric pollutants, such as ozone and its precursors, carbon monoxide, nitrous oxide, sulfur dioxide, hydrocarbons, particulates, etc. to a high degree of accuracy;

(c) continuously output a digitized values of the current pollution levels to the process that compares them with the threshold values for pollution incidents.

PROCESS SPECIFICATIONS

Page 112 of 475

1.5.7 Manage Pollution Data Log

Input Flows

pollution_data_log
pollution_state_static_log_data
pollution_state_vehicle_log_data

Output Flows

pollution_operational_data

Description:

Overview: This process shall be responsible for managing 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 Plan System Deployment function for use in planning future modifications to the ITS network. The process shall send the data at periodic intervals of not more than once in any twenty four hours.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'pollution_state_static_log_data';
- (b)'pollution_state_vehicle_log_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:

- (a)'pollution_data_log'.

Unsolicited Output Processing: This process shall provide the following output flow regardless of whether or not any of the above inputs have been received:

- (a)'pollution_operational_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when either of the unsolicited flows is received, load the data that it contains into the store of pollution log data;
- (c) periodically (not more than once per twenty four hours) retrieve the contents of the log and send it to the Plan System Deployment function using the solicited output flow shown above;
- (d) be responsible for the management of the data in the store of pollution log data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 113 of 475

1.5.8 Manage Pollution Reference Data Store

Input Flows

pollution_reference_data
pollution_reference_data_request
pollution_reference_data_update

Output Flows

pollution_reference_data_output
pollution_state_static_acceptance_criteria
pollution_state_vehicle_acceptance_criteria

Description:

Overview: This process shall be responsible for managing 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'pollution_reference_data_request';
- (b)'pollution_reference_data_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of

requests for data retrieval:
(a)'pollution_reference_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'pollution_reference_data_output';
- (b)'pollution_state_static_acceptance_criteria';
- (c)'pollution_state_vehicle_acceptance_criteria'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) run whenever either of the unsolicited input data flows shown above are received;
- (b) when the first of the unsolicited input flows is received, read the contents of the store of pollution reference data and send it to the requesting process using the appropriate solicited output flow shown above;
- (c) when the second of the unsolicited input flows is received, store the data that it contains in the store of pollution state data, if necessary over writing data already present;
- (d) when (c) is complete, output the received data to the appropriate process using either the first or third solicited output flow;
- (e) be responsible for the management of the data in the store of pollution reference data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 114 of 475

2.1.1 Manage Commercial Fleet Electronic Credentials and Tax Filing

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_retained_data
cf_roadside_activity_report
cf_route
cf_route_details
cf_static_route_data
cf_tag_data

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

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.

Data Flows: All input data flows are unsolicited and all output data flows are solicited with the exception of the following:

- (a)'cf_route_details', which contains data requested from and written to a data store;
- (b)'cf_driver_route', which contains data written to a data store;
- (c)'cf_enrollment_information','cf_enrollment_payment_confirmation','cf_route'and 'cf_static_route_data', which are received as the result of output to other processes.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the input is received, generate the appropriate outputs identified below;
- (c) if the input in (b) contains data to set up a route for a commercial vehicle, use the input data, plus any details of previously used routes in the local store to construct a route request and send it to either the static route selection process, or to the Provide Driver and Traveler Services function for a dynamic route, i.e. one that takes account of current and future traffic conditions;
- (d) when the route requested in (c) has been provided, or the input requests that the electronic

User Service Requirements:

credentials be obtained and the tax details filed, send the data to the commercial vehicle

USR = 4.0; administration process;

USR = 4.4; (e) repeat (d) to enable payment to be made;

USR = 4.4.1; (f) when the route has been determined and the electronic credentials and tax filing have been

USR = 4.6; obtained and paid for, load the route details into the data store for access by the commercial

USR = 4.6.1; vehicle driver;

(g) periodically send details of the taxes that have been paid to the Administer Commercial Vehicles function;

Output Flow Dynamics Assumptions:

(h) the process shall be responsible for the management of the data in the stores of commercial

cf_driver_route = cf_manager_route_request; fleet route details and retained data, and for writing to the store of driver instructions, using

cf_enrollment_request = cf_manager_enrollment_request; the most appropriate mechanism(s) such as RDBMS, for storing the data.

cf_enrollment_payment_request = cf_manager_enrollment_payment_request;

cf_hazmat_route_information = cf_hazmat_request;

cf_hazmat_vehicle_information = cf_hazmat_request;

cf_manager_activity_report = cf_manager_activity_report_request;

cf_manager_enrollment_payment_confirmation = cf_manager_enrollment_payment_request;

cf_manager_enrollment_information = cf_manager_enrollment_request;

cf_manager_route_data = cf_manager_route_request;

cf_route_request = (cf_manager_route_request)*(4/5);

cf_static_route_request = (cf_manager_route_request)/5;

cf_route_details = cf_manager_route_request;

cf_tag_initialization_data = cf_manager_route_request;

cf_tax_audit_data = CVO_MAN / (3*MONTH);

cf_request_activity_report = CVO_MAN/DAY;

cf_retained_data = 1/DAY;

PROCESS SPECIFICATIONS

PROCESS SPECIFICATIONS

Page 115 of 475

Page 116 of 475

2.1.2 Provide Commercial Fleet Static Route

Input Flows

cf_static_route_request

map_data_for_fleet_managers

Output Flows

cf_static_route_data

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.

Data Flows: The input data flow is unsolicited and the output flow is solicited. The following data flow contains data requested from and written to a data store:

(a)'map_data_for_fleet_managers'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above;
- (b) when the input is received, generate the requested route and send its details in the output identified above;
- (c) be responsible for the management of the data in the store of static data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 117 of 475

2.1.3 Provide Flt Mgr Electronic Credentials and Tax Filing Interface

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

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.

Data Flows: All input data flows from the commercial fleet manager terminator are unsolicited and all output flows are solicited. The other input flows are as a result of sending outputs to other processes.

Functional Requirements: This process shall meet the following functional requirements:

User Service Requirements: (a) continuously monitor for receipt of the input flows from the commercial vehicle manager terminator listed above;
USR = 4.0; (b) when the inputs are received, generate the appropriate output flows identified above to other
USR = 4.6; processes and monitor for the responses that they produce;
USR = 4.6.1; (c) when the responses in (b) are received, generate the appropriate output flows to the commercial
USR = 4.6.2; vehicle manager terminator, using the flows listed above..

PROCESS SPECIFICATIONS

Page 118 of 475

PROCESS SPECIFICATIONS

Page 119 of 475

2.1.4 Provide Fleet Manager Commercial Vehicle Communications

Input Flows

cf_on_board_vehicle_data
cf_request_vehicle_data
cf_retrieved_vehicle_data

Output Flows

cf_request_on_board_vehicle_data
cf_vehicle_data

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'cf_request_vehicle_data'

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:

(a)'cf_on_board_vehicle_data';
(b)'cf_retrieved_vehicle_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

(a)'cf_request_on_board_vehicle_data';
(b)'cf_vehicle_data'.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the unsolicited input flows listed above;
(b) be responsible for the management of the data in the store of retrieved vehicle data log, using

the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 120 of 475

2.1.5 Provide Commercial Vehicle Driver Routing Interface

Input Flows

cf_driver_route_instructions
fcvd_request_routing_instructions

Output Flows

cf_driver_route_instructions_request
tcvd_routing_instructions

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'fcvd-request_routing_instructions'.

Solicited Input Processing: This process shall receive the following data flows as a result of data being sent to another process:
(a)'cf_driver_route_instructions'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'cf_driver_route_instructions_request';
(b)'tcvd-routing_instructions'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input flow listed above;
(b) on receipt of the flow identified in (a), generate the first solicited output flow shown above and await the solicited input flow produced as a response;
(c) on completion of (b) generate the second solicited output flow, taking care to present the data in a form that will be readily understood by a commercial vehicle driver..

PROCESS SPECIFICATIONS

Page 121 of 475

2.1.6 Manage Driver Instruction Store

Input Flows

cf_driver_instructions
cf_driver_instructions_request
cf_driver_load_data
cf_driver_route
cf_driver_route_instructions_request

Output Flows

cf_driver_route_instructions
cf_driver_route_instructions_output

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'cf_driver_instructions_request';
(b)'cf_driver_load_data';

- (c)'cf_driver_route';
- (d)'cf_driver_route_instructions_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval:

- (a)'cf_driver_instructions'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'cf_driver_instructions';
- (b)'cf_driver_route_instructions';
- (c)'cf_driver_route_instructions_output'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) run whenever any of the unsolicited input data flows shown above are received;
- (b) when either the second or third unsolicited input flows is received, store the data that they contain in the store of driver instructions using the first solicited output flow, if necessary over writing data already present;
- (c) when either first or fourth of the unsolicited input flows is received, retrieve the requested data from the store and send it to the requesting process in the second or third solicited output data flow;
- (d) be responsible for the management of the data in the store of driver instructions, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 122 of 475

2.2.1 Manage CV Electronic Credential and Tax Filing Interface

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_route_details
cv_static_route_data

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

- (a)'cv_route_details', which contains data requested from and written to a data store;
- (b)'cv_enrollment_information','cv_enrollment_payment_confirmation','cv_route'and 'cv_static_route_data', which are received as the result of output to other processes.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the input is received, generate the appropriate outputs identified below:

- (c) if the input in (b) contains data to set up a route for a commercial vehicle, use the input data, plus any details of previously used routes in the local store to construct a route request and send it to either the static route selection process, or to the Provide Driver and Traveler Services function for a dynamic route, i.e. one that takes account of current and future traffic conditions;
- (d) when the route requested in (c) has been provided, or the input requests that the electronic credentials be obtained and the tax details filed, send the data to the commercial vehicle administration process;
- (e) repeat (d) to enable payment to be made;
- (f) the process shall be responsible for the management of the data in the store of commercial vehicle route details, using the most appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 123 of 475

2.2.2 Provide Vehicle Static Route

Input Flows

cv_static_route_request
map_data_for_cv_drivers

Output Flows

cv_static_route_data

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.

Data Flows: The input data flow is unsolicited and the output flow is solicited. The following data flow contains data requested from and written to a data store:

(a)'map_data_for_cv_drivers'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above;
- (b) when the input is received, generate the requested route and send its details in the output identified above;
- (c) be responsible for the management of the data in the store of map data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 124 of 475

2.2.3 Provide CV Driver Electronic Credential and Tax Filing Interface

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

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.

Data Flows: All input data flows from the commercial vehicle driver terminator are unsolicited and all output flows are solicited. The other input flows are as a result of sending outputs to other processes.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows from the commercial vehicle driver terminator listed above;
- (b) when the inputs are received, generate the appropriate output flows identified above and monitor for any responses.

PROCESS SPECIFICATIONS

Page 125 of 475

PROCESS SPECIFICATIONS

Page 126 of 475

2.2.4 Provide Commercial Vehicle Driver Communications

Input Flows

cv_on_board_vehicle_data
cv_received_vehicle_data
cv_request_vehicle_data

Output Flows

cv_request_on_board_vehicle_data
cv_vehicle_data

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.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:

- (a)'cv_request_vehicle_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:

- (a)'cv_received_vehicle_data';
- (b)'cv_on_board_vehicle_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'cv_request_on_board_vehicle_data';
- (b)'cv_vehicle_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the unsolicited input flow shown above is received, generate the first output flow, which

should result in the second solicited input flow;

(c) receipt of the second solicited input flow shall generate the second solicited output flow, but this flow shall be sent with a null entry if the second input flow is not generated in (b);

(d) be responsible for the management of the data in the store of the received on-board vehicle data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 127 of 475

2.3.1 Produce Commercial Vehicle Driver Message at Roadside

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

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 variable message signs (vms), 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the appropriate one of the three driver output flows identified above using what ever output devices are physically available;
- (c) simultaneously with (b) produce the on-board pull-in output data flow and send it to the commercial vehicle..

PROCESS SPECIFICATIONS

Page 128 of 475

2.3.2.1 Administer Commercial Vehicle Roadside Credentials Database

Input Flows

cv_credentials_data_request
cv_credentials_database_update
cv_credentials_information_response
cv_roadside_credentials_database

Output Flows

cv_credentials_data_output
cv_credentials_information_request

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.

Data Flows: The input data flows and output data flow are unsolicited. The following output data flow is solicited and writes data to a local data store:

- (a)'cv_roadside_credentials_database'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above;
- (b) when the input is received, generate the output identified above.

PROCESS SPECIFICATIONS

Page 129 of 475

2.3.2.2 Process Screening Transactions

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

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.

Data Flows: The input of screening data is unsolicited. All output flows and the other input flows are solicited, and the roadside_problem_credentials'flow contains data read from a store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the screening data flow listed above;
- (b) when the input in (a) is received, check the data against that in the store of problem electronic credentials;
- (c) if a match is found in (b) set the decision to pull-in otherwise set it to pass and send it to the inspector interface process;
- (d) wait for three (3) seconds for the response to (c) and either send the amended decision in the response or the original decision as a pull-in or pass message to the commercial vehicle display interface process;
- (e) encrypt the output data flow in (d) in such a way that its contents cannot be determined using any digital or analog techniques.

PROCESS SPECIFICATIONS

Page 130 of 475

2.3.3.1 Provide Commercial Vehicle Checkstation Communications

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

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'cv_get_on_board_data'.
- (b)'cv_inspection_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:
(a)'cv_on_board_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'cv_roadside_collected_data';
(b)'cv_inspection_data_output';
(c)'cv_request_on_board_data'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input flows listed above;
(b) use the appropriate mechanism(s) such as RDBMS, to store data in the store of collected vehicle data.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 131 of 475
Page 132 of 475

2.3.3.2 Provide Commercial Vehicle Inspector Handheld Terminal Interface

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

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'fci-inspection_data_input';
(b)'fci-start_inspection'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes:
(a)'cv_inspection_results'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'cv_inspector_safety_data_input';
(b)'cv_start_inspection';
(c)'tci-inspection_report'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input flows listed above;
(b) produce the inspection report in a format that is compatible with the output facilities provided by the handheld terminal, e.g. voice, liquid crystal display, etc.

REQUIREMENTS ATTRIBUTES:

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

DYNAMIC ATTRIBUTES:

```
cv_start_inspection = fci-start_inspection;  
tci-inspection_report = cv_inspection_results;  
cv_inspector_safety_data_input = fci-inspection_data_input;
```

PROCESS SPECIFICATIONS

Page 133 of 475

PROCESS SPECIFICATIONS

Page 134 of 475

2.3.3.4 Carry-out Commercial Vehicle Roadside Safety Screening

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
```

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.

Data Flows: The input of safety data is unsolicited. All output flows and the other input flows are solicited, and the'safety_problem_list'flow contains data read from a store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the safety data flow listed above;
- (b) when the input in (a) is received, check the data against that in the store of safety problem data;
- (c) if a match is found in (b) set the decision to pull-in otherwise set it to pass and send it to the inspector interface process;
- (d) wait for three (3) seconds for the response to (c) and either send the amended decision in the response or the original decision as a pull-in or pass message to the commercial vehicle display interface process;
- (e) encrypt the output data flow in (d) in such a way that its contents cannot be determined using any digital or analog techniques.

PROCESS SPECIFICATIONS

Page 135 of 475

PROCESS SPECIFICATIONS

Page 136 of 475

2.3.3.5 Carry-out Commercial Vehicle Roadside Inspection

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  
tcdv_inspection_results
```

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.

Data Flows: The input data flow from the inspector's hand held terminal interface process is unsolicited and all output flows are solicited. The data flow 'cv_roadside_collected_data' contains data requested from a data store, and 'cv_update_problem_data' contains data to be written to a store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow requesting an inspection, as listed above;
- (b) when the input is received, start the inspection process by comparing the data collected from on-board the commercial vehicle with safety criteria, generate outputs to the inspector, the roadside check facility log, the commercial vehicle and the commercial vehicle driver identified above, showing compliance or deviation;
- (c) if deviations are found send the details of the vehicle and the carrier to the safety problems list store using the flow listed above;
- (d) use the appropriate mechanism(s) such as RDBMS, to retrieve data from and write data to the two stores identified above.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 137 of 475
Page 138 of 475

2.3.4 Detect Commercial Vehicle

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the vehicle characteristics input flow shown above;
- (b) when the input in (a) is received, generate the flow requesting the commercial vehicle's on-board tag data;
- (c) as a result of (c) continuously monitor for receipt of the tag data flow;
- (d) when the flow in (c) is received, generate the screening, safety and border data flows shown above and send them to those of the credentials, safety and border crossing processes that are present in the roadside checkstation facility;
- (e) if no data is received in (c), or the data received cannot be interpreted into a set of tag data, generate the pull-in request data flow shown above and send it to the process in the roadside checkstation facility that issues pull-in requests.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 139 of 475
Page 140 of 475

2.3.5 Provide Commercial Vehicle RoadsideOperator Interface

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

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.

Data Flows: All input data flows from the inspector are unsolicited and all other input and output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows from the inspector listed above;
- (b) when the inputs in (a) are received, generate the appropriate outputs identified above;
- (c) as a result of (b) monitor for the receipt of any expected responses and pass those back to the inspector through the appropriate output data flow.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 141 of 475
Page 142 of 475

2.3.6 Provide Commercial Vehicle Reports

Input Flows

cv_archived_inspection_data
cv_archived_safety_data
cv_border_record
cv_roadside_facility_log
cv_roadside_operator_data_request
cv_screening_record

Output Flows

cv_border_daily_log

cv_roadside_daily_log
cv_roadside_operator_output

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

- (a)'cv_roadside_facility_log', which contains data requested from or written to a data store;
- (b)'cv_roadside_daily_log', which is sent daily without being generated by input from other processes.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when any inputs which are not a request for the contents of the log are received, load the data that they contain into the log;
- (c) when the input requests output of the contents of the log, retrieve the data and send it to the source of the request;
- (d) regardless of any other similar requests, daily read the data from the log and send it to the Administer Commercial Vehicles facility;
- (e) be responsible for the management of the data in the roadside facility log, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 143 of 475

2.3.7 Produce Commercial Vehicle Driver Message on Vehicle

Input Flows

cv_on_board_pull_in_output

Output Flows

tcvd_on_board_pull_in_output

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above;
- (b) when the input is received, generate the output flow identified above.

PROCESS SPECIFICATIONS

Page 144 of 475

2.3.8 Provide Commercial Vehicle Border Screening

Input Flows

cv_border_data
cv_border_database_update
cv_border_override
cv_roadside_border_database

Output Flows

cv_border_decision
cv_border_pull_in_output
cv_border_record
cv_on_board_border_record

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'cv_border_data';
- (b)'cv_border_database_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores, or data sent to another process:

- (a)'cv_roadside_border_database';
- (b)'cv_border_override'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'cv_border_decision';
- (b)'cv_border_pull_in_output';
- (c)'cv_on_board_border_record'
- (d)'cv_roadside_border_database'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for the unsolicited inputs flow listed above;
- (b) when the first unsolicited input flow is received, check the trip number against the data that is already contained in the local data store and that the lock tag data shows that it has not been touched;
- (c) if a match is found in (b) and the lock tag is present and has not been tampered, then generate a pass decision, otherwise generate a pull-in decision;
- (d) generate the first solicited output data flow shown above with the pull-in/pass decision that results from (c);
- (e) as a result of (d) continuously monitor for a period of three (3) seconds for receipt of the second solicited input data flow listed above;
- (f) if no data is received in (e), maintain the current decision, otherwise change the decision to that received in the flow;
- (g) as a result of (f) output the pull-in/pass decision in the second solicited output data flow listed above, and also generate the fourth solicited output data flow listed above;
- (h) encrypt the fourth solicited output data flow in such a way that its contents cannot be determined using any digital or analog techniques;
- (i) when the second unsolicited input flow is received in (a), load the data that it contains into the local data store;
- (j) use the appropriate mechanism(s) such as RDBMS, to retrieve data from and write data to the other stores identified above.

PROCESS SPECIFICATIONS PROCESS SPECIFICATIONS

Page 145 of 475

Page 146 of 475

2.4.1 Communicate Commercial Vehicle On-board Data to Roadside

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

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, but with a suitably equipped hand held terminal) 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'cv_inspection_data_output';
- (b)'cv_request_on_board_data';

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes:

- (a)'cv_on_board_data_current_copy'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'cv_inspection_data_update';
- (b)'cv_on_board_data';
- (c)'cv_on_board_data_needed'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the second unsolicited data flow is received generate the third solicited output data flow and await receipt of the solicited input data flow shown above;
- (c) when the data flow in (b) is received, generate the second solicited data flow shown above;
- (d) when the first unsolicited data flow in (a) is received, generate the first solicited output data flow shown above.

PROCESS SPECIFICATIONS

Page 147 of 475

2.4.2 Collect On-board Commercial Vehicle Sensor Data

Input Flows

fcv_brake_condition
fcv_cargo_safety_status
fcv_distance_travelled
fcv_driver_safety_status
fcv_driver_status
fcv_vehicle_safety_status
fcv_weight

Output Flows

cv_on_board_stored_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.

Data Flows: All input data flows are unsolicited and the output flow is solicited containing data to be written to a data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously generate the output of data to the data store listed above, using data scanned from the inputs also listed above;
- (b) complete a full scan of all inputs and generate the output in a matter of milliseconds with a maximum value not to exceed one hundredth of a second (1/100 sec), regardless of the number of inputs to be scanned;
- (c) be capable of accepting input data in a variety of formats, both digital and analog.

PROCESS SPECIFICATIONS

Page 148 of 475

2.4.3 Analyze Commercial Vehicle On-board Data

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

(a)'cv_on_board_stored_sensor_data', which contains data requested from a data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow containing driver input shown above;
- (b) when the input in (a) is received, generate the solicited output flow that send the data it contains to the process that manages the store of on-board vehicle data;
- (c) periodically read the contents of the data store containing commercial vehicle sensor data, analyze the contents and again generate the solicited output flow that send the data it contains to the process that manages the store of on-board vehicle data;
- (d) if during the analysis defined in (c) a critical safety problem is found, generate the output flow that highlights the problem to the commercial vehicle driver interface process;
- (e) read the data from the store of commercial vehicle sensor data, using the most appropriate mechanism(s) such as RDBMS.

PROCESS SPECIFICATIONS

Page 149 of 475

2.4.4 Provide Commercial Vehicle Driver Interface

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs from the driver are received, generate the appropriate outputs to the other processes;
- (c) when inputs from other processes are received, generate the appropriate outputs identified above to the driver.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 150 of 475
Page 151 of 475

2.4.5 Communicate Commercial Vehicle On-board Data to Vehicle Manager

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
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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows containing requests from the commercial vehicle manager or driver listed above for output of the on-board vehicle data;
- (b) when these inputs are received, generate the output data flow to the store management process that requests a copy of the data currently loaded into the store;
- (c) monitor for receipt of the data flow containing the current store contents in response to the data flow sent in (b);
- (d) when the data flow in (c) is received, generate the data flow to send the data to commercial vehicle manager or driver who requested the data;
- (e) if the data flow from the commercial vehicle manager contains a general output message, generate the data flow that sends this data for output by the commercial vehicle driver interface process;
- (f) if the input data flow with the general message is received from the commercial vehicle driver interface process, generate the output to the commercial vehicle manager but without the on-board vehicle data, only include the general message;
- (g) if the input data flow with the request for output of the on-board data is received from the commercial vehicle driver interface process, generate the output flow in (b) and follow through with step (c) above;
- (h) as a result of (g) send the on-board data to the commercial vehicle manager.

PROCESS SPECIFICATIONS

Page 152 of 475

2.4.6 Provide Commercial Vehicle On-board Data Store Interface

Input Flows

cv_inspection_data_update
cv_on_board_data_needed
cv_on_board_data_required
cv_on_board_data_update
cv_on_board_stored_data
cv_provide_credentials_data_for_inspections

Output Flows

cv_on_board_data_current_copy
cv_on_board_data_output

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

(a)'cv_on_board_stored_data', which contains data requested from or written to a data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows containing updates to the data in the store or request the output of the current contents of the store;
- (b) when any of the update inputs are received, write the new data into the store;
- (c) when any of the requests for data output are received, read the current contents of the store and generate the appropriate output flow to send the data to the requesting process;
- (d) when the data flow with the credentials details is received, load the data into the store overwriting any similar data that store already contains;
- (e) be responsible for the management of the data in the store of commercial vehicle data using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 153 of 475

2.5.1 Manage Commercial Vehicle Trips and Clearances

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

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'cf_enrollment_payment_request';
- (b)'cf_enrollment_request';
- (c)'cv_enrollment_payment_request';
- (d)'cv_enrollment_request';
- (e)'cv_request_enrollment_data';
- (f)'cf_tax_audit_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and queries to local data stores:

- (a)'cv_check_credentials_response';
- (b)'cv_confirmed_enrollment'
- (c)'cv_remote_enrollment_confirmation';
- (d)'cv_tax_and_credential_fees'- which is data being read from a store
- (e)'cv_update_new_credentials_response'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'cf_enrollment_information';
- (b)'cf_enrollment_payment_confirmation';
- (c)'cv_check_credentials_request';
- (d)'cv_enrollment_information';
- (e)'cv_enrollment_list';
- (f)'cv_enrollment_payment_confirmation
- (g)'cv_provide_enrollment_data';
- (h)'cv_remote_enrollment_request';

User Service Requirements:

- (j)'cv_update_new_credentials_request'.

USR = 4.0;

USR = 4.6; Functional Requirements: This process shall meet the following functional requirements:

USR = 4.6.1; (a) continuously monitor for receipt of the unsolicited input flows listed above;

Output Flow Dynamics Assumptions:

- (b) when the inputs containing enrollment requests are received, either from a local process or from the process that communicates with other administer commercial vehicles facilities, send the output $cf_enrollment_information = 1/DAY * CVO_VEHS$; data flow to check that the enrollment can be completed, using the supplied credentials;
- $cf_enrollment_payment_confirmation = 4/WEEK * CVO_VEHS$; (c) when the response to (b) is received, generate the enrollment information data flow and sent it
- $cv_check_credentials_request = 1/DAY * CVO_VEHS + 1/WEEK * CVO_DVR$; back to the process from which the request was sent;
- $cv_enrollment_information = 1/WEEK * CVO_DVR$; (d) when inputs containing enrollment payment information are received, either from a local process
- $cv_enrollment_payment_confirmation = 4/WEEK * CVO_DVR$; or from the process that communicates with other administer commercial vehicles facilities, generate
- $cv_enrollment_list = 4/WEEK * ITS_CVO_VEHS + 1/DAY * CVO_DVR$; the enrollment list data flow including in it the cost of the necessary payments and duties that are
- $cv_remote_enrollment_request = 1/WEEK * CVO_VEHS$; in the store of tax and credential fees, and send it to the process that obtains payment;
- $cv_provide_enrollment_data = cv_remote_enrollment_request$; (e) monitor for the response to (d), and when received, generate the update enrollment data flow;
- $cv_request_permits_and_duties_update = 4/WEEK * CVO_VEHS$; (f) when the response to (e) is received, generate the enrollment payment confirmation data flow and
- $cv_update_new_credentials_request = 4/WEEK * CVO_VEHS + 4/WEEK * CVO_DVR$; send it to the process from which the request in (d) was received;
- (g) periodically send a request to the process that interfaces to the government administrators to request an update to the tax and credentials data;
- (h) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 154 of 475

PROCESS SPECIFICATIONS

Page 155 of 475

2.5.2 Obtain Electronic Credential and Tax Filing Payment

Input Flows

cv_enrollment_list
cv_tax_and_credential_fees
financial_response

Output Flows

cv_confirmed_enrollment
financial_request

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'cv_enrollment_list'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:

(a)'authorization_code';
(b)'cf_credit_identity';
(c)'cv_tax_and_credential_fees'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

(a)'cv_confirmed_enrollment';
(b)'financial_request'.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the unsolicited input flows listed above;
(b) when the inputs are received, generate the outputs identified above;
(c) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 156 of 475

2.5.3 Update Permits and Duties Store

Input Flows

cv_request_permits_and_duties_update
fga_carrier_safety_ratings
fga_roadside_facility_locations
fga_tax_and_credential_fees

Output Flows

cv_roadside_facility_locations_update
cv_safety_history_update
cv_tax_and_credential_fees
tga_request_fees_updates

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.

Data Flows: The input data flow requesting an update to the data is unsolicited and all output flows are solicited. The following data flows contain data to be written to a data store:

(a)'cv_tax_and_credential_fees';
(b)'cv_safety_history_update'.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the input flow listed above which requests an update of the data from the Government Administrators;

- (b) when the input is received, generate the output identified above to the Government Administrators;
- (c) when data is received from the Government Administrators load it into the data stores using the two flows identified above;
- (d) use the appropriate mechanism(s) such as RDBMS, to write data to the two stores identified above.

PROCESS SPECIFICATIONS

Page 157 of 475

2.5.4 Communicate with Other Commercial Vehicle Administration

System

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

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.

Data Flows: With the exception of the input data flows listed below which are unsolicited, all other input and output flows are solicited as a result of input of either of these flows, with the exception of the 'cv_tax_and_credential_fees' data flow which contains data requested from a local data store.

- (a)'cv_remote_enrollment_request';
- (b)'focvas-enrollment_request'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the two unsolicited input flows listed above;
- (b) when the local input flow is received, generate the output identified above requesting data from the remote system;
- (c) when the data requested in (b) arrives from the remote system send it to the requesting process;
- (d) when the input requesting data from the remote system is received, collect the data from the local store and process, and send it back to the remote system;
- (e) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 158 of 475

PROCESS SPECIFICATIONS

Page 159 of 475

2.5.5 Manage Commercial Vehicle Credentials and Enrollment

Input Flows

cv_check_credentials_request
 cv_commit_local_enrollment
 cv_roadside_facility_location
 cv_safety_history
 cv_update_new_credentials_request
 fevoir_request_for_information
 fea_cv_enforcement_agency_response

Output Flows

cv_check_credentials_response
cv_commit_remote_enrollment
cv_database
cv_update_new_credentials_response
tcvoir_carrier_or_vehicle_information
tea_cv_request_for_information

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.

Data Flows: All input data flow containing the enrollment requests from commercial vehicle managers and drivers are unsolicited. All output flows are solicited. The following data flows write data to, or receive data from data stores:

- (a)'cv_roadside_facility_location', read only;
- (b)'cv_database', write only;
- (c)'cv_safety_history', read only.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above containing the enrollment requests from either commercial vehicle managers or drivers, or from a remote commercial vehicle administration system, or from an external entity;
- (b) when the inputs from the commercial vehicle managers or drivers are received, enroll the data by updating the database of commercial vehicle credential data identified above for the roadside checkstation facilities that are on the vehicle's route;
- (c) as part of (b), send requests for enrollment to any roadside facilities that are on the route but not in the area served by the function to the process that interfaces to other commercial vehicle administration systems using the commit data flow;
- (d) before carrying out any enrollments, send the output to the enforcement agency to check that there are no outstanding prosecutions for the carrier/driver/vehicle combination that might prevent the vehicle being cleared to pass a roadside checkstation facility;
- (e) when input is received from the information requesters, obtain the required data from the store of safety data and send it back to the requester;
- (f) be responsible for the management of the data in the database of commercial vehicle credential data, using the appropriate mechanism(s) such as RDBMS, for storing the data.
- (g) also use the appropriate mechanism(s) such as RDBMS, to retrieve data from the other store identified above.

PROCESS SPECIFICATIONS

Page 160 of 475

PROCESS SPECIFICATIONS

Page 161 of 475

2.5.6 Output Commercial Vehicle Enrollment Data to Roadside

Facilities

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

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.

Data Flows: There are no unsolicited input data flows and all output flows are solicited. Two input flows are generated as a result data queries to the following data stores:

- (a)'cv_facility_log';
- (b)'cv_safety_history'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) periodically, e.g. daily, request data from the two data stores identified above;
- (b) when the data has been received, generate the outputs identified above;
- (c) when any of the requesting data flows are received, retrieve the required data using the same flows as used in (a) above;
- (d) when the data has been received, generate the appropriate response flows identified above;
- (e) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the data stores identified above.

PROCESS SPECIFICATIONS

Page 162 of 475

2.5.7 Process Commercial Vehicle Violations

Input Flows

cv_facility_log

Output Flows

cv_violation_data

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.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:

- (a)'cv_facility_log'.

Unsolicited Output Processing: This process shall provide the following output flows regardless of any inputs that are received:

- (a)'cv_violation_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) run periodically (e.g. daily) and read the data from the store of log data obtained from commercial vehicle roadside checking and border facilities;
- (b) where any safety or credential problems have occurred that could make the driver and/or carrier liable for prosecution, send the data to the law enforcement interface process in the Manage Emergency Services function;
- (c) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 163 of 475

2.5.8 Process Data Received from Roadside Facilities

Input Flows

cf_request_activity_report
cv_border_daily_log
cv_facility_log
cv_roadside_daily_log
cv_update_safety_problems_list

Output Flows

cf_periodic_activity_report
cf_roadside_activity_report
cv_operational_data
cv_safety_history_write
tga_quarterly_reports

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..

Data Flows: All input data flow containing the roadside and border daily logs, plus the enrollment requests from commercial vehicle fleet managers and drivers are unsolicited. All output flows are solicited. The following data flows write data to, or receive data from data stores:

- (a)'cv_facility_log', read and write;
- (b)'cv_safety_history_write', write only.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above containing the facility logs, details of commercial vehicles that have failed their safety inspections, and requests from the commercial vehicle manager for activity reports;
- (b) when the input of facility logs are received, update the roadside log data store identified above, and if necessary the store of safety history data;
- (c) when processing the facility log, remove all traces of vehicle, carrier and driver identity and produce a log of activity which is sent to the Plan System Deployment function using the output flow listed above;
- (d) when an activity report request is received, generate the appropriate report data and send it to the requesting commercial vehicle manager;
- (e) if a periodic report request is received, generate the first report as in (d) and generate the subsequent reports at the requested intervals;
- (f) be responsible for the management of the data in the store of the facility log, using the appropriate mechanism(s) such as RDBMS, for storing the data;
- (g) also use the appropriate mechanism(s) such as RDBMS, to write data to the other store identified above.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 164 of 475
Page 165 of 475

2.6.1 Provide Commercial Vehicle Manager Tag Data Interface

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

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'cf_tag_initialization_data';
- (b)'fcvm-carrier_number';
- (c)'fcvm-driver_number';
- (d)'fcvm-request_tag_data_output';
- (e)'fcvm-trip_identity';
- (f)'fcvm-vehicle_number'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes:

- (a)'cf_tag_data_store_output';
- (b)'cf_clearance_enrollment_confirm'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'cf_tag_data';
- (b)'cf_tag_data_store_request';
- (c)'cf_tag_data_store_write';
- (d)'tcvm-confirm_enrollment_data_stored';
- (e)'tcvm-output_tag_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above as originating from the commercial vehicle manager;
- (b) when the inputs are received, generate the outputs identified above and write the data into the store identified above;
- (c) also send the data to the process that manages electronic credentials and tax filing so that it can be used for future enrollments;
- (d) use the appropriate mechanism(s) such as RDBMS, to write data to the store identified above.

PROCESS SPECIFICATIONS

Page 166 of 475

PROCESS SPECIFICATIONS

Page 167 of 475

2.6.2 Transmit Commercial Vehicle Tag Data

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

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'cv_request_electronic_clearance_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'cv_lock_tag_data'- which is data received from another process;
- (b)'cv_on_board_border_record'- which is data received from another process;
- (c)'cv_on_board_screening_record'- which is data received from another process;
- (d)'cv_tag_data_store_read'- which is data retrieved from a store.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'cv_electronic_clearance_data';
- (b)'cv_request_lock_tag_data';
- (c)'cv_tag_data_store_needed';
- (d)'cv_tag_data_store_update'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) when the input flow in (a) is received, de-encrypt the data and generate the solicited output data flows shown above that request the current tag data and the current lock tag data;
- (c) as a result of (b) await receipt of the solicited input data flow listed above containing the data from the tag and de-encrypt it when it arrives;
- (d) monitor for receipt of the first solicited data flow shown above (from the lock tag) and de-encrypt it when it arrives;
- (e) if no data is received in (d) do not wait more than one second before concluding that none is available and that the response data must be set to blank indicating no lock tag;
- (f) when (c) and (d) have been completed, send the retrieved data to the requesting commercial vehicle roadside checkstation or border crossing facility using the first solicited output data flow shown above;
- (g) as a result of (e) monitor for receipt of either of the second or third solicited data flows listed above, and when received, use the data they contain to generate the third solicited output data flow shown above;
- (h) all solicited output data flows whether to the tag or the roadside facility must be encrypted in such as way that it is not possible to determine their contents, using any digital or analog techniques.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 168 of 475
Page 169 of 475

2.6.3 Provide Commercial Driver Tag Data Interface

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

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fcvd-carrier_number';
- (b)'fcvd-driver_number';
- (c)'fcvd-request_tag_data_output';
- (d)'fcvd-trip_identity';
- (e)'fcvd-vehicle_number'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above input being received:

- (a)'cv_tag_data_store_request';
- (b)'cv_tag_data_store_write';
- (c)'tcvd-confirm_data_stored'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above as originating from the commercial vehicle driver;
- (b) when the inputs are received, generate the outputs identified above and write the data into the store identified above;
- (c) use the appropriate mechanism(s) such as RDBMS, to write data to the store identified above.

PROCESS SPECIFICATIONS

Page 170 of 475

2.6.4 Provide Lock Tag Data Interface

Input Flows

cv_request_lock_tag_data
fcv_lock_tag_data

Output Flows

cv_lock_tag_data
tcv_lock_tag_data_request

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'cv_request_lock_tag_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to a terminator:

- (a)'fcv-lock_tag_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'cv_lock_tag_data';
- (b)'tcv-lock_tag_data_request'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for the unsolicited input flow listed above;
- (b) when the flow in (a) is received, output the second solicited output data flow shown above;
- (c) continuously monitor for receipt of the solicited input flow shown above and when it is received, generate the first solicited output flow shown above;
- (d) all solicited output data flows must be encrypted in such as way that it is not possible to determine their contents, using any digital or analog techniques.

PROCESS SPECIFICATIONS

Page 171 of 475

2.6.5 Manage Commercial Vehicle Tag Data Store

Input Flows

cf_tag_data_store_request
cf_tag_data_store_write
cv_tag_data_store
cv_tag_data_store_needed
cv_tag_data_store_request
cv_tag_data_store_update
cv_tag_data_store_write

Output Flows

cf_tag_data_store_output
cv_provide_credentials_data_for_inspections
cv_tag_data_store_output
cv_tag_data_store_read

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'cf_tag_data_store_request';
- (b)'cf_tag_data_store_write';
- (c)'cv_tag_data_store_needed';
- (d)'cv_tag_data_store_request';
- (e)'cv_tag_data_store_write';
- (f)'cv_tag_data_store_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval:

- (a)'cv_tag_data_store'- which is data retrieved from a store.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'cf_tag_data_store_output';
- (b)'cv_provide_credentials_data_for_inspections';
- (c)'cv_tag_data_store_output';
- (d)'cv_tag_data_store_read'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when either of the input flows in (a) from the commercial vehicle manager or driver, containing data to be written to the tag is received, write the data it contains to the store, overwriting the same data already held in the store;
- (c) as a result of (b) generate the flow that sends data for use in safety inspections;
- (d) when the flow in (a) containing update data is received, write the data it contains to the store, overwriting the data from the oldest roadside facility that is already held in the store;
- (e) when any of the data flows in (a) from the commercial vehicle manager or driver, requesting data is received, retrieve the current data from the store and generate the corresponding solicited output data flow shown above;
- (f) the retrieved in (e) must not include the data sent by commercial vehicle roadside facilities;
- (g) it must not be possible for the commercial vehicle manager or driver to over-write the data on the tag that has been sent from the roadside facilities;
- (h) use the most appropriate mechanism such as RDBMS, to read and write data from and to the data store.

PROCESS SPECIFICATIONS

Page 172 of 475

2.7 Manage Cargo

Input Flows

From_Intermodal_Freight_Depot
From_Intermodal_Freight_Shipper

Output Flows

To_Intermodal_Freight_Depot
To_Intermodal_Freight_Shipper

Description:

Overview: This process shall be responsible for providing facilities for the management of cargo shipments. The process shall enable these shipments to be routed via intermodal shippers and depots and may not need the services of a commercial vehicle manager or driver.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the input is received, generate the appropriate outputs.

PROCESS SPECIFICATIONS

Page 173 of 475

3.1.1 Produce Collision and Crash Avoidance Data

Input Flows

collision_data
intersection_collision_avoidance_data

Output Flows

position_warnings
vehicle_change_position_requests
vehicle_crash_restraint_commands

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.

Data Flows: The input data flow is unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when either of the input flows is received, generate the outputs identified above;
- (c) the time to produce the outputs must be measured in milliseconds and be not greater than one tenth of a second (1/10 sec) regardless of the number of times any of the inputs are received;
- (d) the content of the position warning message shall be tailored to reflect the data input received.

PROCESS SPECIFICATIONS

Page 174 of 475

PROCESS SPECIFICATIONS

Page 175 of 475

3.1.2 Carry-out Safety Analysis

Input Flows

safety_data

Output Flows

safety_warnings
vehicle_and_driver_safety_status

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.

Data Flows: The input data flow is unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above;
- (b) when the input is received, generate the outputs identified above;
- (c) the time to produce the outputs must be measured in milliseconds and be not greater than one tenth of a second (1/10 sec) regardless of the number of time the input is received.
- (d) only send the data to the vehicle control processes if the input shows that there is a safety problems with the driver or the vehicle;
- (e) the content of the safety warning message shall be tailored to reflect the data input received.

PROCESS SPECIFICATIONS

Page 176 of 475

3.1.3 Process Vehicle On-board Data

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
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

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.

Data Flows: All input data flows are unsolicited and both output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously generate the outputs list above, using data scanned from the inputs also listed above;
- (b) complete a full scan of all inputs and generate the outputs in not more than one hundredth of a second (1/100 sec) regardless of the number of inputs to be scanned and the number of outputs generated;
- (c) the vehicle probe data shall contain details of the type of hazard found on the road around the vehicle, or be blank if there are none;
- (d) be capable of accepting input data in a variety of formats, both digital and analog.

PROCESS SPECIFICATIONS

Page 177 of 475

PROCESS SPECIFICATIONS

Page 178 of 475

3.2.1 Provide Driver Interface

Input Flows

ahs_status
control_status
vehicle_control_request

Output Flows

driver_ahs_input
driver_input
vehicle_control_status

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) the time to produce the outputs must be measured in milliseconds and be not greater than one tenth of a second (1/10 sec) regardless of the number of times the any of the inputs are received;
- (d) interpretation of the vehicle control data flow shall be as shown in the flow definition;
- (e) if mode 5 has been actioned and a request for modes 1 through 4 is detected in the vehicle control data flow, the flow to the ahs control process shall be sent with manual mode requested;
- (f) when the response to (e) shows that the vehicle is leaving the ahs lanes, then data for the requested automatic control mode shall be sent to the processes responsible for this mode of vehicle operation;
- (g) if the control statu input shows any of the failure modes (6 through 8), the flow to the ahs control process shall be sent with manual mode requested;
- (h) all changes in mode of vehicle operation shall be accompanied by the output of appropriate warning data to the processes that provide driver outputs;
- (i) all inputs for changes in vehicle control mode must be asserted for one (1) second before they are interpreted, whilst those for the control status must be present for one tenth (1/10) of a second.

PROCESS SPECIFICATIONS

Page 179 of 475

3.2.2 Provide AHS Control

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the driver ahs input data flow is are received, implement the instructions it contains;
- (c) if ahs mode is requested in (b) and the vehicle is not in that mode, initiate the ahs check-in procedure, and when successfully completed, the ahs route generation procedure, and when this is completed, initiate ahs control by sending data to the vehicle control processes;
- (d) if the ahs check-in procedure fails then set the ahs status output to check-in failed and do not proceed further with any ahs processing or send any control parameters to other processes;
- (e) when the ahs based route has been selected, send the list of route segments to the process that manages ahs operations for loading into the operational data store;
- (f) if the vehicle condition input shows that the vehicle cannot continue in ahs mode or the driver ahs input requests departure from ahs mode, send the ahs cancel data to the vehicle control processes before generating any other outputs;
- (g) any change in ahs status shall produce an ahs status data flow output;
- (h) this process must scan all inputs and complete the subsequent processing in not more than one hundredth (1/100) of a second regardless of the number of changes in input and any interruptions.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 180 of 475
Page 181 of 475

3.2.3.1 Provide Command Interface

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor the platoon status feedback data flow and if at any time it shows that the vehicle and/or driver is unsafe (value of 999) or that a built in self test (bist) failure has been detected, send unsafe or failure data messages to the processes responsible for the driver interface, platoon following, vehicle actuators and servo actuators;
- (b) if the platoon status feedback data flow is set to safe (value other than 999), continuously monitor the data flow driver input for any changes;
- (c) if the flow in (b) changes from manual (0) to any other state then if none of the feedback status messages are set to failure send data to the appropriate processes dependent on the mode of automatic vehicle control requested by the driver;
- (d) following (c) monitor the return data from the receiving process(es) and pass on the data to the driver interface process, setting the output to failure if no response is received within one (1) second of the data being sent;
- (e) all activity by this process must be completed within one hundredth (1/100) of a second regardless of the activity involved, or the number of inputs being processed.

PROCESS SPECIFICATIONS

Page 182 of 475

3.2.3.2 Manage Platoon Following

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above so that the vehicle maintains the required platoon following status in a manner that ensures its safety and that of other vehicles regardless of whether the vehicles behind or in front are joining or leaving the platoon;
- (c) if requested to leave the platoon, do so in a manner that ensures the safe operation of the remaining vehicles in the platoon and any other vehicles on the highway, as well as the safety of the driver plus other vehicle occupants;
- (d) all activity by this process must be completed within one hundredth (1/100) of a second regardless of the activity involved, or the number of inputs being processed.

PROCESS SPECIFICATIONS

Page 183 of 475

PROCESS SPECIFICATIONS

Page 184 of 475

3.2.3.3 Process data for Vehicle Actuators

Input Flows

actuator_commands
driver_manual_input
fbv_brake_servo_response
fbv_steering_servo_response
fbv_throttle_servo_response
platooning_selected
vehicle_change_position_requests
vehicle_crash_restraint_commands

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) respond immediately, i.e. within one thousandth (1/1000) of a second to requests for any changes in actuator state regardless of the source of the request;
- (c) if any actuator does not make the requested change, or moves without input, send an error response to the process responsible for providing servo controls, provide command interface and platoon following;
- (d) regardless of any other actions being carried out, continuously perform an operational self check using a watchdog to ensure that all actions are being carried out in time and that the processor is functioning correctly, and if a failure occurs set the BIST result in the data flow sent to the process responsible for monitoring actuatorstatus and providing the command interface;
- (e) the process shall assume manual mode of operation on start up and until instructed by input flows to change to any other condition.

PROCESS SPECIFICATIONS

Page 185 of 475

PROCESS SPECIFICATIONS

Page 186 of 475

3.2.3.4.1 Provide Speed Servo Control

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above to either maintain the requested speed or temporarily change to another value for as long as the override data input is present;
- (c) the output of the new throttle setting is to be sent to the vehicle control data interface process for onward communication to the process responsible for vehicle actuators;
- (d) regardless of any other actions being carried out, continuously perform an operational self check using a watchdog to ensure that all actions are being carried out in time and that the processes is

functioning correctly;

(e) if a failure occurs in (d) then set the BIST result part of the speed servo status feedback data flow to indicate a failure (999) and send it to the process that provides the command interface;

(f) the time to produce the outputs must be measured in milliseconds and be not greater than one tenth of a second (1/10 sec) regardless of the number of times the any of the inputs are received.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 187 of 475
Page 188 of 475

3.2.3.4.2 Provide Headway Servo Control

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above to either maintain the requested headway or temporarily change to another value for as long as the override data input is present;
- (c) the output of the new brake setting is to be sent to the vehicle control data interface process for onward communication to the process responsible for vehicle actuators;
- (d) regardless of any other actions being carried out, continuously perform an operational self check using a 'watchdog' to ensure that all actions are being carried out in time and that the processes is functioning correctly;
- (e) if a failure occurs in (d) then set the BIST result part of the speed servo status feedback data flow to indicate a failure (999) and send it to the process that provides the command interface;
- (f) the time to produce the outputs must be measured in milliseconds and be not greater than one tenth of a second (1/10 sec) regardless of the number of times the any of the inputs are received.

PROCESS SPECIFICATIONS

Page 189 of 475

3.2.3.4.3 Provide Lane Servo Control

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above to either maintain the requested lane position or temporarily change to another lane for as long as the override data input is present;
- (c) the output of the lane position setting is to be sent to the vehicle control data interface process for onward communication to the process responsible for vehicle actuators;
- (d) regardless of any other actions being carried out, continuously perform an operational self check using a 'watchdog' to ensure that all actions are being carried out in time and that the processes is functioning correctly;
- (e) if a failure occurs in (d) then set the BIST result part of the speed servo status feedback data flow to indicate a failure (999) and send it to the process that provides the command interface;
- (f) the time to produce the outputs must be measured in milliseconds and be not greater than one tenth of a second (1/10 sec) regardless of the number of times the any of the inputs are received.

PROCESS SPECIFICATIONS

Page 190 of 475

3.2.3.4.4 Provide Change Lane Servo Control

Input Flows

manual_steering_input_detected
platoon_change_lane_servo_override

Output Flows

feedback_change_lane_servo_status
override_lane_hold
steering_commands

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above to change from one lane to another for as long as the input is received;
- (c) the output of the lane change data is to be sent to the vehicle control data interface process for onward communication to the process responsible for vehicle actuators;
- (d) regardless of any other actions being carried out, continuously perform an operational self check using a 'watchdog' to ensure that all actions are being carried out in time and that the processes is functioning correctly;
- (e) if a failure occurs in (d) then set the BIST result part of the speed servo status feedback data flow to indicate a failure (999) and send it to the process that provides the command interface;
- (f) the time to produce the outputs must be measured in milliseconds and be not greater than one tenth of a second (1/10 sec) regardless of the number of times the any of the inputs are received.

PROCESS SPECIFICATIONS

Page 191 of 475

3.2.3.4.5 Provide Vehicle Control Data Interface

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) the time to produce the outputs must be measured in milliseconds and be not greater than one tenth of a second (1/10 sec) regardless of the number of times the any of the inputs are received.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 192 of 475
Page 193 of 475

3.2.3.5 Process Vehicle Sensor Data

Input Flows

fbv_vehicle_headway
fbv_vehicle_lane_position
fbv_vehicle_on_ahs_lane
fbv_vehicle_speed

Output Flows

feedback_sensor_status
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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) convert the inputs from what ever form they are in (analog, digital, etc.) into a digital output which can be used by other processes;
- (c) send the outputs every second to the receiving processes regardless of the state of the inputs;
- (d) this process shall complete its activity in less than one second regardless of the number of inputs being processed, or the amount of processing needed to convert the inputs into digital signals.

PROCESS SPECIFICATIONS

Page 194 of 475

3.2.3.6 Communicate with other Platoon Vehicles

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs to other processes within one hundredth (1/100) of a second, regardless of the number of inputs received at any one instant;
- (c) data that is received from processes within the vehicle for output to other vehicles shall also meet the processing criteria in (b) above;
- (d) if not data is received in (c) then do not send data to any other vehicle.

PROCESS SPECIFICATIONS

Page 195 of 475

3.2.4 Process Sensor Data for AHS input

Input Flows

fbv_vehicle_condition
fbv_vehicle_on_ahs_lane

Output Flows

ahs_vehicle_data

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) convert the inputs from what ever form they are in (analogue, digital, etc.) into digital form that can be used by other processes;
- (c) complete the processing required in (b) and the output of the resultant data in less than one hundredth (1/100) of a second, regardless of the number of inputs detected and the amount of processing needed to convert them into digital outputs.

PROCESS SPECIFICATIONS

Page 196 of 475

3.2.5 Check Vehicle for AHS eligibility

Input Flows

ahs_control_information
ahs_vehicle_checking_paramters
ahs_vehicle_condition

Output Flows

ahs_check_response
ahs_checking_data

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'ahs_vehicle_condition'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:
(a)'ahs_control_information';
(b)'ahs_vehicle_checking_paramters'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'ahs_check_response';
(b)'ahs_checking_data';
(c)'ahs_vehicle_checking_paramters'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input flows listed above;
(b) check the data obtained from the vehicle against the locally stored parameters to establish that the vehicle is eligible for ahs operation;
(c) if a vehicle satisfies the parameter check, return a positive response in the first output flow listed above and include in it the special vehicle control data used for ahs operation;
(d) if a vehicle fails the parameter check, only send back a negative response in the first output flow listed above;
(e) be responsible for the management of the data in the store of the parameters against which vehicles are checked for ahs operation, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 197 of 475

3.2.6 Manage AHS Check-in and Check-out

Input Flows

ahs_checking_data
ahs_control_data_changes
ahs_route_data

Output Flows

ahs_checking_details
ahs_control_information

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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'ahs_control_data_changes';
- (b)'ahs_checking_data';
- (c)'ahs_route_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'ahs_checking_details';
- (b)'ahs_control_information'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) store the input data identified above when ever it is received, and send the first output flow shown above back to the process responsible for checking vehicles for ahs operation;
- (c) periodically read the data from the ahs transaction store and send it to the Plan System Deployment function using the second output flow listed above.

PROCESS SPECIFICATIONS

Page 198 of 475

3.2.7 Manage AHS Operations

Input Flows

ahs_checking_details
ahs_control_data
ahs_usage_data

Output Flows

ahs_control_data_changes
ahs_operational_data

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.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:

- (a)'ahs_control_data';
- (b)'ahs_checking_details'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from a local data store:

- (a)'ahs_usage_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'ahs_control_data_changes';
- (b)'ahs_operational_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) store the input data identified above when ever it is received, and send the first output flow shown above to the process responsible for checking vehicles for ahs operation;
- (c) periodically read the data from the ahs transaction store and send it to the Plan System Deployment function using the second output flow listed above;
- (d) be responsible for the management of the data in the store containing the ahs transaction log, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 199 of 475

3.3.1 Provide Cargo Data for Incident Notification

Input Flows

cargo_data_request
fcv_cargo_data
fcv_cargo_safety_status

Output Flows

processed_cargo_data

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow requesting cargo status listed above;
- (b) when the input in (a) is received, generate the outputs using data collected by processing the inputs from the on-board commercial vehicle sensors identified above;
- (c) the processing of the sensor input and generation of the output shall be completed within one hundredth (1/100) of a second regardless of the number of inputs and the amount of processing needed to produce a digital output that can be used by other processes.

PROCESS SPECIFICATIONS

Page 200 of 475

3.3.2 Provide Communications Function

Input Flows

emergency_request_vehicle_acknowledge
vehicle_emergency_request

Output Flows

emergency_message_auto_output
emergency_request_vehicle_details

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the vehicle emergency request flow listed above;
- (b) when the input is received, immediately generate the output to the Manage Emergency Services function identified above;
- (c) when a response is received to the output in (b) send the data flow to the Provide Driver and Traveler Services function so that a message can be output to the driver;
- (d) the processing of the sensor input and generation of the output shall be completed within one hundredth (1/100) of a second regardless of the number of inputs and the amount of processing needed to produce a digital output that can be used by other processes.

PROCESS SPECIFICATIONS

Page 201 of 475

3.3.3 Build Automatic Collision Notification Message

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited, with the exception of the data read from the store of vehicle identity.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow of data from sensors indicating that the vehicle has been involved in a crash;
- (b) when the input in (a) is received, get the cargo state by sending a request to the monitoring process and extracting the data from the reply;
- (c) if no data is received in (b) within one hundredth of a second, assume that there is no cargo and guild the message for output to the communications process using the vehicle location and identity data inputs listed above;.
- (d) when all the data has been assembled, output the data flow to the communications process;
- (e) the processing of the sensor input and generation of the output shall be completed within one tenth (1/10) of a second regardless of the amount of processing needed to produce the output data flow to the communications process.

PROCESS SPECIFICATIONS

Page 202 of 475

3.4 Enhance Driver's Vision

Input Flows

From_Roadway

Output Flows

vision_data

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.

Data Flows: The input data flow is unsolicited and the output flow is solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continually receive data from the roadway through in-vehicle sensors;
- (b) process the data received in (a) to produce an enhanced picture of the driver's view from the vehicle, in the forward, reverse or sideways directions;
- (c) the enhancement provided in the data must both improve the field of view, i.e. the distance of the farthest point which can be seen and the width of view, and the clarity with which all objects within that distance can be seen;
- (d) the resulting data must be continuously sent in the vision data output to another process for combination with other driver output data;
- (e) complete a full scan of the input and generate the output in a matter of milliseconds with a maximum value not to exceed one hundredth of a second (1/100 sec) regardless of the size of the input (field of view) to be scanned.

PROCESS SPECIFICATIONS

Page 203 of 475

4.1.1 Process Transit Vehicle Sensor Trip Data

Input Flows

ftv_vehicle_trip_data

Output Flows

transit_vehicle_arrival_times

transit_vehicle_collected_trip_data

transit_vehicle_on_board_data

Description:

Overview: This process shall be responsible for collecting and processing 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.

Data Flows: The input data flow is unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above;
- (b) when the input is received, generate the outputs identified above.

PROCESS SPECIFICATIONS

Page 204 of 475

4.1.2.1 Determine Transit Vehicle Deviation and ETA

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_schedule_deviation

Description:

Overview: This process shall be responsible for determining the schedule deviation and estimated times of arrival (ETA) at transit stops for transit vehicle. This 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above using the methods described below and with the exceptions noted below;
- (c) generate the schedule deviation using the current service details (routes and schedules) and the vehicle's current location using interpolation or some other algorithmic method;
- (d) use similar methods to generate the transit vehicle's estimated time of arrival at the next transit stop for output to the stop as the vehicle approaches;
- (e) only if the deviation is small and in an urban area the process shall send the data to the process that generates corrective instructions;
- (f) if the deviation is large or not in an urban area, the process shall not produce the output identified in (e) above and shall instead just send the data to the process that manages transit vehicle deviations;
- (g) the process shall generate outputs even when the deviation is zero.

PROCESS SPECIFICATIONS

Page 205 of 475

4.1.2.2 Determine Transit Vehicle Corrective Instructions

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

Description:

Overview: This process shall be responsible for generating outputs that enable a transit vehicle

schedule to be 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.

Data Flows: The input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the input flow of transit vehicle deviations is received, generate the outputs identified above using any appropriate algorithms for determining the corrective instruction data for the transit vehicle driver.

PROCESS SPECIFICATIONS

Page 206 of 475

4.1.2.3 Provide Transit Vehicle Driver Interface

Input Flows

transit_vehicle_corrective_instructions
transit_vehicle_deviation_data

Output Flows

ttd_corrective_instructions
ttd_transit_vehicle_schedule_deviations

Description:

Overview: This process shall be responsible for providing a schedule correction interface for the transit driver in the transit vehicle. The interface provided by this process shall be concerned with providing 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 provided to the process by other processes in the Manage Traffic function. The output provided by the process shall be available in audio or visual form in such way that whilst 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'transit_vehicle_corrective_instructions';
- (b)'transit_vehicle_deviation_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'ttd-corrective_instructions';
- (b)'ttd-transit_vehicle_schedule_deviations'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) the output shall be presented in such a way that it provides the necessary information without jeopardizing the driver's ability to operate the vehicle safely, both for its passengers and for other vehicles on the roads and freeways;
- (c) the output shall be maintained until fresh data is received through the unsolicited input flows listed above.

PROCESS SPECIFICATIONS

Page 207 of 475

4.1.2.4 Provide Transit Vehicle Correction Data Output Interface

Input Flows

transit_vehicle_arrival_conditions

Output Flows

titsp_transit_arrival_changes

Description:

Overview: This process shall be responsible for providing the interface through which intermodal transportation service providers are informed of a transit vehicle schedule deviation. The output

provided by the process shall be as a result of 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 intermodal 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'transit_vehicle_arrival_conditions'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'titsp-transit_arrival_changes'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input flows listed above;
(b) provide the solicited output flow listed above in a form that will enable the intermodal service provide to take any remedial action to the service it provides so that transit users suffer the minimum of inconvenience as a result of a late running service;
(c) if no input is received, or it shows that there are no deviations, set the output to zero (or null) to reflect this condition.

PROCESS SPECIFICATIONS

Page 208 of 475

4.1.2.5 Request Transit Vehicle Preemptions

Input Flows

transit_vehicle_preemption_request

Output Flows

transit_vehicle_roadway_preemptions

Description:

Overview: This process shall be responsible for providing the interface through which requests for preemption can be output from a transit vehicle. The output shall be provided by the process as a result of data being received from another process in the Manage Transit function. The process shall provide the output in a form that it can be used by the controllers at intersections, pedestrian crossings and multimodal crossings on the roads (surface streets) and freeway network served by the Manage Traffic function to provide priority at the transit vehicle. If no data is received from the other process, or it shows that no preemption is needed, the process shall produce no output.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'transit_vehicle_preemption_request'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'transit_vehicle_roadway_preemptions'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input flow listed above;
(b) when the flow listed above is received, produce the solicited output flow listed above in a form that can be used by roadside intersection controllers to give priority to the transit vehicle;
(c) if no input flow is received, or it indicates that preemption is not required, produce no output data flow.

PROCESS SPECIFICATIONS

Page 209 of 475

4.1.3 Provide Transit Vehicle Location Data

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

Description:

Overview: This process shall be responsible for providing the transit vehicle's current location to a high degree of accuracy. The location shall be computed by this process from data sent by other processes that provide 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when any of the inputs are received, generate the outputs identified above by combining the new data with the last output values;
- (c) the calculation of the new location shall use the basic location data and refine it by use of data from on-board the vehicle, e.g. proximity of transit stop, vehicle doors open, etc.

PROCESS SPECIFICATIONS

Page 210 of 475

4.1.4 Manage Transit Vehicle Deviations

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
tftm_proposed_corrections

Description:

Overview: This process shall be responsible for managing 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the input of deviation from schedule is received use the current transit services and the forecast of traffic conditions provided by the data in the flow of predictive data received from the Manage Traffic function as input to algorithms that can produce a return to service strategy;
- (c) when the process in (b) is complete, send the strategy to the transit fleet manager for approval;
- (d) when the approval in (c) is received, send the remaining output flows, and await confirmation of the implementation of the requested priority by the Manage Traffic function;
- (e) if the confirmation on (d) is not given, then re-implement (b) through (d) using the fact that priority requests to the Manage Traffic function are not available.

PROCESS SPECIFICATIONS

Page 211 of 475

4.1.5 Provide Transit Vehicle Status Information

Input Flows

ftfm_request_transit_vehicle_data
transit_conditions_demand_request
transit_vehicle_information

Output Flows

transit_information_request
transit_running_data_for_demand
transit_vehicle_arrival_time
transit_vehicle_data
transit_vehicle_status
ttfm_transit_vehicle_data

Description:

Overview: This process shall be responsible for providing 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. The data shall be obtained by this process from another process that manages a store of transit vehicle operating data either on request from another process that needs the data, or at regular periodic intervals for those outputs that cannot be requested. When data is requested, this process shall only return data to the source of the request.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) at regular periodic intervals, e.g. once per day, request the contents of the store of transit vehicle operational data from the store management process;
- (c) when the data in (b) is received, send out the data flows listed above that do not have any corresponding data request flow;
- (d) when any other input listed above is received, output the flow requesting transit vehicle information and send the response back to the originating process.

PROCESS SPECIFICATIONS

Page 212 of 475

4.1.6 Manage Transit Vehicle Operations Data

Input Flows

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_operating_data
transit_vehicle_schedule_deviation

Output Flows

transit_deviation_data_received
transit_vehicle_collected_maintenance_data_request
transit_vehicle_deviations_details
transit_vehicle_information
transit_vehicle_user_data

Description:

Overview: This process shall be responsible for managing 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from and written to a data store: (a)'transit_vehicle_operating_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs that are not data requests are received, load the data that they contain into the data store using the flow identified above and generate the update and transit vehicle information output flows identified above;
- (c) periodically generate requests for data collected by sensors on-board the vehicle and store the solicited data when received
- (d) when inputs that are data requests are received, retrieve the data and only send it to the source of the request;
- (e) the transit user data flow that is sent to each transit stop shall be sent as each transit vehicle approached a stop to provide information for output to transit users at the stop;
- (f) be responsible for the management of the data in the store of transit vehicle operational data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 213 of 475

4.1.7 Provide Transit Vehicle Deviation Data Output Interface

Input Flows

transit_vehicle_arrival_deviations

Output Flows

titsp_transit_arrival_deviations

Description:

Overview: This process shall be responsible for providing the interface through which intermodal transportation service providers are informed of transit vehicle schedule deviations. The output provided by the process shall be as a result of 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 intermodal supplier so that transit users are not inconvenienced by the deviations. A zero (or null) output shall be provided when no deviations are present.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'transit_vehicle_arrival_deviations'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'titsp-transit_arrival_deviations'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) provide the solicited output flow listed above in a form that will enable the intermodal service provide to take any remedial action to the service it provides so that transit users suffer the minimum of inconvenience as a result of a late running service;
- (c) if no input is received, or it shows that there are no deviations, set the output to zero (or null) to reflect this condition.

PROCESS SPECIFICATIONS

Page 214 of 475

4.1.8 Provide Transit Vehicle Operations Data Interface

Input Flows

transit_conditions_advisories_request
transit_conditions_guidance_request
transit_deviation_data_received
transit_deviation_kiosk_request
transit_deviations_portables_request
transit_vehicle_deviations_details

Output Flows

tm_transit_schedule_variations
transit_deviations_for_kiosks
transit_deviations_for_portables
transit_running_data_for_advisory_output
transit_running_data_for_guidance
transit_vehicle_deviations_details_request

Description:

Overview: This process shall be responsible for providing the interface through which data about transit schedule deviations is made available to ITS functions outside the Manage Transit function and to the media. The process shall only provide data to the media when prompted by the arrival of new deviation data in the store of transit vehicle operational data. All other output shall only be provided by the process when a request is received from a process in another ITS function. The process shall obtain the required data from the process that manages the store of transit vehicle operating data.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'transit_conditions_advisories_request';
- (b)'transit_conditions_guidance_request';
- (c)'transit_deviation_data_received';
- (d)'transit_deviation_kiosk_request';
- (e)'transit_deviations_portables_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes:

- (a)'transit_vehicle_deviations_details'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'tm-transit_schedule_variations';
- (b)'transit_running_data_for_advisory_output';
- (c)'transit_running_data_for_guidance';
- (d)'transit_vehicle_deviations_details_request';
- (e)'transit_deviations_for_kiosks';
- (f)'transit_deviations_for_portables'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) generate the transit deviations details message when any of the inputs are received and return the data received in the solicited input flow to the source(s) of the input(s);
- (c) only provide output to the media when the transit vehicle deviation data received input data flow listed above is received and the data provided by the solicited input flow has also been received.

PROCESS SPECIFICATIONS

Page 215 of 475

4.1.9 Process Transit Vehicle Sensor Maintenance Data

Input Flows

ftv_vehicle_maintenance_data
transit_vehicle_collected_maintenance_data_request

Output Flows

transit_vehicle_collected_maintenance_data

Description:

Overview: This process shall be responsible for collecting and processing 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.

Data Flows: The input data flow is unsolicited and the output flow is solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flows listed above;
- (b) when the first input data flow is received, process it and if required translate it into a digital form;
- (c) when the second input data flow is received, generate the output data flow identified above.

PROCESS SPECIFICATIONS

Page 216 of 475

4.2.1.1 Process Demand Responsive Transit Trip Request

Input Flows

paratransit_schedule

paratransit_service_confirmation
paratransit_service_data
paratransit_trip_request

Output Flows

paratransit_personal_schedule
paratransit_request
paratransit_requested_services

Description:

Overview: This process shall be responsible for providing 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from and written to the store of request data:

(a)'paratransit_service_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the flow in (a) is a new schedule request, generate the corresponding output identified above and send it to the schedule generation process;
- (c) as a result of (b) continuously monitor for the receipt of the flow listed above containing details of the proposed schedule;
- (d) when the flow in (c) is received, load the data into the store of demand responsive transit schedules, including with it an identity number;
- (e) when (d) is successfully complete, generate the output listed above that contains the details of the proposed service for use by the requesting process;
- (f) when the flow in (a) is a schedule confirmation, read the schedule data corresponding to the identity number and generate the output to the schedule confirmation process identified above;
- (g) be responsible for the management of the data in the store of trip request data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 217 of 475

4.2.1.2 Compute Demand Responsive Transit Vehicle Availability

Input Flows

paratransit_transit_vehicle_availability
transit_vehicle_location

Output Flows

paratransit_available_vehicles

Description:

Overview: This process shall be responsible for providing the facility for the calculation of the location and availability of transit vehicles for use in demand responsive transit operation. 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 inputs to sensors. The output shall be loaded by the process into a store for use by another process.

Data Flows: All input data flows are unsolicited and the output flow shall be sent to the store of available transit vehicles using the flow'paratransit_available_vehicles'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the output identified above;
- (c) be responsible for the management of the data in the store of transit vehicle availability, using the appropriate mechanism(s) such as RDBMS, for storing the data..

4.2.1.3 Generate Demand Responsive Transit Schedule and Routes**Input Flows**

paratransit_available_vehicles
 paratransit_request
 paratransit_services
 transit_services_for_demand_response

Output Flows

paratransit_schedule
 transit_services_demand_response_request

Description:

Overview: This process shall be responsible for providing the 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 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

- (a)'paratransit_services', which contains data requested from and written to a data store;
- (b)'paratransit_available_vehicles', which also contains data requested from a data store;
- (c)'transit_services_for_demand_response', which is received as a result of output being sent to another process.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above using an appropriate routes and schedule generation algorithm;
- (c) be responsible for the management of the data in the store of generated routes and schedules, using the appropriate mechanism(s) such as RDBMS, for storing the data;
- (d) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the store of available transit vehicles identified above.

4.2.1.4 Confirm Demand Responsive Transit Schedule and Route**Input Flows**

paratransit_requested_services
 paratransit_services

Output Flows

paratransit_service_output
 paratransit_services_for_transit_drivers
 paratransit_transit_driver_instructions
 ttfm_paratransit_service

Description:

Overview: This process shall be responsible for providing 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.

Data Flows: The input data flow is unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

- (a)'paratransit_services'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above;

(b) when the input is received, generate the outputs identified above;

(c) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 220 of 475

4.2.1.5 Process Demand Responsive Transit Vehicle Availability Data

Input Flows

ftv_availability

Output Flows

paratransit_transit_vehicle_availability

Description:

Overview: This process shall be responsible for processing data input to sensor(s) on board a transit vehicle. Data on the vehicle's availability for use in demand responsive transit services shall be provided by the process from the input. This data shall be sent to another process for fusion with other data and loading into a store from which it can be accessed by another process in the Manage Transit function.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flow:

(a)'ftv-availability'.

Solicited Output Processing: This process shall provide the following output flow as a result of the above inputs being received:

(a)'paratransit_transit_vehicle_availability'.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the unsolicited input flow shown above;

(b) analyze the input flow and if required, transform it into digital data for use by other processes;

(c) send the data in the solicited output flow shown above to the process that will fuse the data with that for vehicle location.

PROCESS SPECIFICATIONS

Page 221 of 475

4.2.1.6 Provide Demand Responsive Transit Driver interface

Input Flows

paratransit_transit_driver_instructions

Output Flows

ttd_paratransit_information

Description:

Overview: This process shall be responsible for providing 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 whilst 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 visual form may include display and hardcopy formats.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flow:

(a)'paratransit_transit_driver_instructions'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

(a)'ttd-paratransit_information'.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the unsolicited input flows listed above;

(b) produce the output to the transit driver in such a way that it does not jeopardize the driver's safe operation of the transit vehicle, but conveys the required information in an easily understandable form;

(c) maintain the output for as long as the schedule is current, i.e. until the time of the last activity or the last arrival time has past.

PROCESS SPECIFICATIONS

Page 222 of 475

4.2.2 Provide Transit Plans Store Interface

Input Flows

paratransit_service_output
transit_plans
transit_routes_request
transit_routes_updates
transit_schedule_request
transit_schedule_updates
transit_services_demand_response_request

Output Flows

transit_routes_current_data
transit_schedule_current_data
transit_services_for_demand_response

Description:

Overview: This process shall be responsible for providing 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 schedule data shall be accessible as input to the regular transit route and schedule generation processes.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from and written to the store of transit plans:

(a)'transit_plans'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) be responsible for the management of the data in the store of transit plans data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 223 of 475

4.2.3.1 Generate Transit Routes

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

Description:

Overview: This process shall be responsible for generating 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 data containing the demand responsive transit routes and schedules that have been requested as a further input from which the new routes are generated. 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.

Data Flows: The input data flow for updating routes and services is unsolicited and all other input and output flows are then solicited as a result of its receipt. The following data flows are as a result of requests for data from stores:

- (a)'map_data_for_transit';
- (b)'transit_operational_data';
- (c)'transit_service_planning_parameters'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow requesting update of the routes and services from the transit fleet manager and shown in the list above;
- (b) when this input is received, initiate the generation process, reading in all the required data from stores, or requesting it from the store interface process;
- (c) use the data in (b) to produce the output of new transit routes data using an appropriate route generation algorithm;
- (d) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the stores identified by the flows shown above.

PROCESS SPECIFICATIONS

Page 224 of 475

4.2.3.2 Generate Schedules

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

Description:

Overview: This process shall be responsible for generating 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 that have been requested as a further input from which the new schedules are generated. 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.

Data Flows: The input data flow for updating routes and services is unsolicited and all other input and output flows are then solicited as a result of its receipt. The following data flows are as a result of requests for data from stores:

- (a)'transit_operational_data';
- (b)'transit_service_planning_parameters'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow requesting update of the schedules shown in the list above;
- (b) when this input is received, initiate the generation process, reading in all the required data and produce the output of the new transit schedules data using an appropriate schedule generation algorithm;
- (c) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the stores identified by the flows shown above.

PROCESS SPECIFICATIONS

Page 225 of 475

4.2.3.3 Produce Transit Service Data for external use

Input Flows

transit_service_external_data
transit_services_advisories_request
transit_services_demand_request
transit_services_guidance_request
transit_services_kiosk_request

transit_services_portables_request
transit_services_travelers_request

Output Flows

request_transit_service_external_data
titsp_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_portables
transit_services_for_travelers

Description:

Overview: This process shall be responsible for obtaining transit routes and services data and distributing 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 intermodal transportation service provider. For those 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'transit_services_advisories_request';
- (b)'transit_services_demand_request';
- (c)'transit_services_guidance_request';
- (d)'transit_services_kiosk_request';
- (e)'transit_services_travelers_request';
- (f)'transit_services_portables_request';
- (g)'transit_service_external_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to another process:

- (a)'transit_service_external_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'request_transit_service_external_data';
- (b)'titsp-transit_service_data';
- (c)'transit_services_for_advisory_data';
- (d)'transit_services_for_demand';
- (e)'transit_services_for_deployment';
- (f)'transit_services_for_guidance';
- (g)'transit_services_for_kiosks';
- (h)'transit_services_for_travelers';
- (i)'transit_services_for_portables'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when any of the unsolicited inputs shown above except the last are received, the process shall immediately generate the first solicited output shown above;
- (c) when the solicited input flow is received as a result of (b) begin generation of the requested data, only including the details necessary to meet the request, i.e. all of the transit routes and schedules need by provided in response to every request;
- (d) data shall only be sent to the source from which the data request originated;
- (e) before output, the process shall put the data into a format that is easily read and interpreted by external processes and can also be read by travelers and transit users with the minimum of

User Service Requirements:

- further processing;
- (f) if the second unsolicited input is received, i.e. fresh service data is received without being

USR = 2.0; requested, the data shall only be sent to the intermodal transportation service provider using the

USR = 2.3; second solicited output flow.

USR = 2.3.2;

USR = 2.3.2.2;

USR = 2.3.2.3;

PROCESS SPECIFICATIONS

Page 226 of 475

PROCESS SPECIFICATIONS

Page 227 of 475

4.2.3.4 Provide Transit Fleet Manager Interface for Services Generation

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
tftm_parameters
tftm_transit_services_output
update_routes
update_schedules

Description:

Overview: This process shall be responsible for providing 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 process shall support inputs from the transit fleet 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data written to a data store:

(a)'transit_service_planning_parameters'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows from the transit fleet manager listed above;
- (b) when the inputs in (a) are received, generate the appropriate outputs identified above;
- (c) continuously monitor for receipt of any input data flows that may be produced by the output flows generated in (b);
- (d) use the appropriate mechanism(s) such as RDBMS, to write data to the store of transit service planning parameters, the flow to which is identified above, when input of new/updated parameters is received from the transit fleet manager;
- (e) the process shall allow the schedule generation process to be initiated on its own, but shall always initiate that process if initiation of the routes generation process is requested, i.e. it shall not be possible to have old schedules applied to newly generated routes.

PROCESS SPECIFICATIONS

Page 228 of 475

PROCESS SPECIFICATIONS

Page 229 of 475

4.2.3.5 Manage Transit Operational Data Store

Input Flows

ftfm_passenger_loading_updates

transit_operational_data
transit_roadside_passenger_data
transit_vehicle_availability
transit_vehicle_data
transit_vehicle_passenger_data

Output Flows

transit_passenger_operational_data
ttfm_passenger_loading_error

Description:

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

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'transit_roadside_passenger_data';
- (b)'transit_vehicle_passenger_data';
- (c)'transit_vehicle_availability';
- (d)'transit_vehicle_data'.

Solicited Input Processing: This process shall receive the following input flow as a result of data being sent to the transit fleet manager terminator:

- (a)'ttfm-passenger_loading_updates'.

Solicited Output Processing: This process shall provide the following output flow to a data store as a result of the above inputs being received:

- (a)'transit_operational_data'.

Solicited Output Processing: This process shall provide the following output flow to another process as a result of reading data from a data store:

- (a)'transit_passenger_operational_data';
- (b)'ttfm-passenger_loading_error'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when any of the inputs is received, write the data into the store of operational data using the output flow shown above;
- (c) when writing the data to the store, rationalize the two counts of the numbers of passengers for each transit route segment, (one being reported by the fare collection process and the other by the transit vehicle monitoring process), reporting any differences to the transit fleet manager;
- (d) once per day, read the data from the transit operational data store and send it to the Plan System Deployment function using the solicited flow shown above;
- (e) be responsible for the management of the data in the store of transit operational data, using the most appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 230 of 475
Page 231 of 475

4.2.3.6 Produce Transit Service Data for Manage Transit Use

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

Description:

Overview: This process shall be responsible for obtaining transit routes and services data and distributing 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'transit_service_internal_data';
- (b)'transit_services_for_eta_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to another process:

- (a)'transit_service_internal_data'.

Unsolicited Output Processing: This process shall provide the following output flows regardless of any inputs that are received:

- (a)'request_transit_service_internal_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'transit_services_for_corrections';
- (b)'transit_services_for_eta';
- (c)'transit_services_for_advanced_fares';
- (d)'transit_services_for_vehicle_fares';
- (e)'transit_services_for_roadside_fares';
- (f)'transit_services_for_scenarios';
- (g)'transit_services_for_transit_drivers'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the unsolicited input shown above is received, the process shall immediately generate all of the solicited outputs shown above;
- (c) if the first unsolicited input is not received periodically, then the process shall generate the unsolicited output shown above.

PROCESS SPECIFICATIONS

Page 232 of 475

4.2.3.7 Provide Interface for Other TRM Data

Input Flows

fotrm_transit_services
transit_services_for_other_TRM

Output Flows

other_TRM_service_data
totrm_transit_services

Description:

Overview: This process shall be responsible for providing the interface through which transit routes and schedules can be exchanged with other transit centers (Other TRM). This data shall be output by the process when data input 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fotrm-transit_services';
- (b)'transit_services_for_other_TRM'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'other_TRM_service_data':

(b)'totrm-transit_services'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when either of the unsolicited input flows is received, generate the corresponding output flow.

PROCESS SPECIFICATIONS

Page 233 of 475

4.2.3.8 Provide Interface for Transit Service Raw Data

Input Flows

fitsp_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
transit_service_raw_data

Output Flows

transit_service_external_data
transit_service_internal_data
transit_services_data_for_output
transit_services_for_other_TRM

Description:

Overview: This process shall be responsible for providing and managing 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, who 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 intermodal transportation service providers. The process shall load this data into the data store for use by the local route and schedule generation processes.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fitsp-transit_service_data';
- (b)'other_TRM_service_data';
- (c)'request_transit_services_data_for_output';
- (d)'request_transit_service_external_data';
- (e)'request_transit_service_internal_data';
- (f)'transit_routes_data';
- (g)'transit_schedule_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:

- (a)'map_transit_data';
- (b)'transit_routes_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'transit_services_data_for_output';
- (b)'transit_service_external_data';
- (c)'transit_services_for_other_TRM';
- (d)'transit_service_internal_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when either of the last two unsolicited input flows is received, automatically output each of the solicited output flows shown above;
- (c) when either of the first two data flows is received, the data shall be loaded into the data

store using the output flow listed above;

(d) when either of the unsolicited inputs that request data is received, the requested data shall be read from the store using the solicited input flow listed above, and shall then be output to the requesting process using the relevant solicited output flow listed above;

(e) where required, the digitized data showing transit maps shall be included in the output flow generated in (d);

(f) be responsible for the management of the data in the store of the raw transit service data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 234 of 475

PROCESS SPECIFICATIONS

Page 235 of 475

4.2.3.9 Update Transit Map Data

Input Flows

fmup_transit_map_update

request_transit_map_update

Output Flows

map_data_for_transit

tmup_transit_map_update_request

Description:

Overview: This process shall be responsible for providing 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, on 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 it from the store loaded by this process.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

(a)'request_transit_map_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to external functions:

(a)'fmup-transit_map_update'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

(a)'tmup-transit_map_update_request';

(b)'map_data_for_transit'.

Functional Requirements: This process shall meet the following requirements:

(a) continuously monitor for the receipt of the unsolicited data flow shown above;

(b) when the data flow in (a) is received, generate the first solicited output data flow shown above and continuously monitor for receipt of the solicited input data flow shown above;

(c) when the flow in (b) is received, output the second solicited output data flow shown above;

(d) be capable of receiving the input data in a variety of formats and converting it into a single format suitable for use with the store of digitized map data;

(e) be responsible for the management of the data in the store of digitized map data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 236 of 475

4.3.1 Monitor Transit Vehicle Condition

Input Flows

transit_vehicle_maintenance_specs

transit_vehicle_status

Output Flows

transit_vehicle_maintenance

transit_vehicle_maintenance_information

Description:

Overview: This process shall be responsible for monitoring 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 and sent to the process that generates transit vehicle maintenance schedules.

Data Flows: The input data flows are unsolicited and the output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above and not that shown as being requested from the data store;
- (b) when the input in (a) is received, generate the outputs identified above using data obtained from the data store.

PROCESS SPECIFICATIONS

Page 237 of 475

4.3.2 Generate Transit Vehicle Maintenance Schedules

Input Flows

transit_vehicle_maintenance_information

Output Flows

transit_vehicle_availability

transit_vehicle_maintenance_schedule

transit_vehicle_maintenance_schedule_data

Description:

Overview: This process shall be responsible for generating 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.

Data Flows: The input data flow is unsolicited and all output flows are solicited with the exception of the following:

- (a)'transit_vehicle_maintenance_schedule', which contains data written to the transit vehicle operations data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, produce the maintenance schedule including details of the work that is to be done and when it shall be done;
- (c) when (c) is completed, generate the outputs identified above;
- (d) use the appropriate mechanism(s) such as RDBMS, to write data to the store identified above.

PROCESS SPECIFICATIONS

Page 238 of 475

4.3.3 Generate Technician Work Assignments

Input Flows

ftfm_technician_information_request

ftfm_technician_information_updates

transit_technician_data

transit_vehicle_maintenance_schedule_data

transit_vehicle_maintenance_verification_results

Output Flows

transit_technician_work_assignment

ttfm_technician_information

ttmp_work_schedule

Description:

Overview: This process shall be responsible for assigning transit maintenance personnel to a transit vehicle maintenance schedule. The maintenance schedule shall be produced by another process as input to this 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 process shall support inputs from the transit fleet manager in both manual and audio form, and shall provide its outputs to the transit fleet manager and transit maintenance personnel in audible and visual forms. It shall enable the visual output to be in hardcopy, or as a display.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the transit vehicle maintenance input is received, generate the maintenance schedule, using data in the store of transit maintenance technician data;
- (c) on completion of (b), output the work assignment to the maintenance personnel and the process that monitors and verifies maintenance work activity using the data flows identified above;
- (d) when a request for personnel data is received from the transit fleet manager, retrieve the requested data from the store and output it to the manager using the data flow identified above;
- (e) be responsible for the management of the data in the store of the raw transit service data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 239 of 475

4.3.4 Monitor And Verify Maintenance Activity

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

Description:

Overview: This process shall be responsible for verifying that the transit vehicle maintenance activities were performed correctly and the generation of a time stamped maintenance log for record keeping. 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the input of vehicle status is received, compare it with the maintenance schedule read from the store of transit vehicle operations data and the maintenance personnel work assignments generated by another process;
- (c) if the result is satisfactory, generate the outputs identified above.

PROCESS SPECIFICATIONS

Page 240 of 475

4.3.5 Report Transit Vehicle Information

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
tftm_transit_vehicle_maintenance_information

Description:

Overview: This process shall be responsible for providing 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 and shall produce the output to the transit fleet manager in an easily understood form. The process shall support inputs from the transit fleet 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.

Data Flows: The input data flow from the transit fleet manager is unsolicited and the output flow to the transit fleet manager is solicited. The remaining input data flow is obtained by requesting data from the transit vehicle operations data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above;
- (b) when the input are received, read the requested data from the store of transit vehicle operations data;
- (c) when (b) is complete, generate the output to the transit fleet manager identified above;
- (d) when the new transit vehicle maintenance specification data is received from the fleet manager, generate the output to the transit vehicle operations data store management process..

PROCESS SPECIFICATIONS

Page 241 of 475

4.3.6 Update Transit Vehicle Information

Input Flows

ftmp_transit_vehicle_maintenance_updates

Output Flows

transit_vehicle_maintenance_data_update

Description:

Overview: This process shall be responsible for providing the transit maintenance personnel with the capability to update transit vehicle maintenance information. The process shall load the data received from the transit maintenance personnel into the store of transit vehicle operations data for use by other processes.

Data Flows: The input data flow is unsolicited. The output flow is solicited and contains data that is written to the store of transit vehicle operations data.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above;
- (b) when the input is received from the transit maintenance personnel, load the received data into the store of transit vehicle operations data using the output flow identified above.

PROCESS SPECIFICATIONS

Page 242 of 475

4.3.7 Manage Transit Vehicle Operations Data Store

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
transit_vehicle_operations_data

Output Flows

transit_vehicle_maintenance_data
transit_vehicle_maintenance_specs

Description:

Overview: This process shall be responsible for the management of 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.

Data Flows: The input data flow is unsolicited and all output flows are solicited with the exception of the following:

- (a)'transit_vehicle_operations data', which contains data written to and read from the transit vehicle operations data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs containing new data are received, load the data into the store;
- (c) if the data in (b) contains new maintenance specifications, send then to the vehicle condition

and verify maintenance activities processes;

(d) when the input containing requests for data is received, retrieve the required data from the store and send it to the requesting process;

(e) use the appropriate mechanism(s) such as RDBMS, to manage the data store identified above.

PROCESS SPECIFICATIONS

Page 243 of 475

4.4.1.1 Manage Transit Security

Input Flows

fsa_transit_panic_button
fsa_transit_video_image
transit_media_incident_interface_parameters
transit_operator_security_action

Output Flows

transit_incident_details
transit_incident_information
transit_media_incident_information
transit_operator_incident_information
tsa_broadcast_message
tsa_panic_button_acknowledge
tsa_video_camera_control

Description:

Overview: This process shall be responsible for providing management of 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 that are revealed. Any problems that are revealed shall be passed by the process to the transit system operator for review and input of the required action. Information about incidents shall also be sent by the 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 in the secure area environment.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which is received as a result of output to another process:

(a)'transit_operator_security_action'.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the input flows listed above that come from the secure area environment, or from the transit system operator interface process;

(b) when the panic button input is received, generate the data flow that sends the incident information to the transit system operator interface process;

(c) if no response is received from the output of the flow in (b), the output of the data flow in (b) shall be repeated at not more than 30 second intervals until a response is received;

(d) when a response is received to the data flow in (b), generate the other output data flows;

(e) continuously monitor the video input flow from the secure area environment and process it to determine if any incidents are taking place, and if so carry out similar actions to those in (b) through (d) above;

(f) if the response data flow in (d) is received without there being any input from the secure area environment, output the camera control data flow.

PROCESS SPECIFICATIONS

Page 244 of 475

PROCESS SPECIFICATIONS

Page 245 of 475

4.4.1.2 Manage Transit Emergencies

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

Description:

Overview: This process shall be responsible for supporting the management of emergencies that occur in the transit system by processing information about those on-board 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which is received as a result of output to another process:

(a)'transit_operator_request_acknowledge'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) if no response is received from the other process to the notification of the transit system operator, the output of the message to that process shall be repeated at not more than 30 second intervals until a response is received;
- (d) the data about the emergency sent to the transit system operator shall include the transit vehicle location which is derived from separate input to that for the emergency data.

PROCESS SPECIFICATIONS

Page 246 of 475

4.4.1.3 Provide Transit System Operator Security Interface

Input Flows

ftso_emergency_request_acknowledge
ftso_media_parameter_request
ftso_media_parameter_updates
ftso_security_action
ftso_video_camera_action_request
transit_media_interface_parameters
transit_operator_emergency_request
transit_operator_incident_information

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

Description:

Overview: This process shall be responsible for providing an interface for the transit system operator through which information potential security problems and emergencies can be output and acted upon. This information shall be provided by other processes through 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 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 in hardcopy, or as a display.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

- (a)'ftso-emergency_request_acknowledge', which is a result of output to the transit system operator;
- (b)'ftso-security_action'which is a result of output to the transit system operator;
- (c)'transit_media_interface_parameters', which is data written to or requested from the store of transit media interface parameters.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs providing information about emergencies are received, generate the output to the transit system operator identified above;
- (c) as a result of (b) monitor for input of a response from the transit system operator using the

- input defined above;
- (d) on receipt of the data in (c), generate the output to the process that provided the information about the emergency;
- (e) when the data flow requesting the media interface parameters is received, read the data from the store and output it to the transit system operator;
- (f) when the data flow with updates to the media interface parameters is received, load the data into the store of these parameters;
- (g) be responsible for the management of the data in the store of transit media interface parameters, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 247 of 475

PROCESS SPECIFICATIONS

Page 248 of 475

4.4.1.4 Provide Transit Media Interface for Emergencies

Input Flows

transit_media_emergency_information
transit_media_incident_information

Output Flows

tm_transit_emergency_information
tm_transit_incident_information

Description:

Overview: This process shall be responsible for providing the interface through which information about security problems and emergencies detected within the transit system are notified directly to the media. 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 shall be output by the process. The process shall provide its outputs in audible and visual forms, it being possible for the latter to be in hardcopy, or as a display.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'transit_media_emergency_information';
- (b)'transit_media_incident_information'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'tm-transit_emergency_information';
- (b)'tm-transit_incident_information'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) upon receipt of either of the inputs the process shall immediately generate the appropriate output, using the supplied parameters to determine what information, and the format in which it shall be supplied to the media.

PROCESS SPECIFICATIONS

Page 249 of 475

4.4.1.5 Provide Transit Driver Interface for Emergencies

Input Flows

ftd_emergency_request
transit_driver_emergency_acknowledge

Output Flows

transit_driver_emergency_request
ttd_emergency_information

Description:

Overview: This process shall be responsible for providing an interface in 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'ftd-emergency_request';
- (b)'transit_driver_emergency_acknowledge'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'transit_driver_emergency_request';
- (b)'ttd-emergency_information'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) upon receipt of either of the inputs, immediately generate the appropriate output shown above;
- (c) the output to the transit driver and the method of providing the input must be transparent to transit users and anyone in the vicinity of the transit vehicle..

PROCESS SPECIFICATIONS

Page 250 of 475

4.4.1.6 Collect Transit Vehicle Emergency Information

Input Flows

transit_emergency_details
transit_media_emergency_interface_parameters

Output Flows

transit_emergency_data
transit_media_emergency_information

Description:

Overview: This process shall be responsible for collecting data about emergencies that occur on-board transit vehicles for output to the media and the Manage Emergency Services function. It shall be possible for these emergencies to 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 store.

Data Flows: All input data flows are unsolicited and all output flows are solicited, with the exception of the following which contains data read from a data store:

- (a)'transit_media_emergency_interface_parameters'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above;
- (b) when the input is received, generate the two outputs identified above, adding the data from the media interface parameters store to that being sent to the media interface process.

PROCESS SPECIFICATIONS

Page 251 of 475

4.4.2 Coordinate Multiple Agency Responses to Incidents

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
tftm_coordination_request

Description:

Overview: This process shall be responsible for providing 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 process shall support inputs from the transit fleet 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.

Data Flows: Two of the input data flows are unsolicited and all output flows are solicited. The other flows are defined as follows with the exception of the following which contains data requested from a data store:

- (a)'transit_preplanned_responses_for_incidents', which contains data requested from a data store;
- (b)'ftfm-coordination_data', which is received as a result of a previous output to the transit fleet manager;
- (c)'transit_incident_coordination_data', which is received as a result of output being sent to processes in the Manage Emergency Services function.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when either of the information inputs are received, search the store of predefined responses for a match;
- (c) if (c) is successful, generate the output to the Manage Emergency Services function identified above;
- (d) if (c) is unsuccessful, output all the data received as input to the transit fleet manager and monitor for receipt of the solicited input flow from the manager;
- (e) when the solicited input flow in (d) is received, generate the output to the Manage Emergency Services function identified above;
- (f) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the store of predefined responses identified above.

PROCESS SPECIFICATIONS

Page 252 of 475

4.4.3 Generate Responses for Incidents

Input Flows

ftfm_request_response_parameter_output
ftfm_response_parameters
transit_preplanned_responses_for_incidents

Output Flows

tftfm_response_parameter_output

Description:

Overview: This process shall be responsible for providing 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 by the process in a form in which it can be used by another process to provide coordination data to the Manage Emergency Services function. The process shall support inputs from the transit fleet 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.

Data Flows: Both input data flows are unsolicited and the output flow is solicited with the exception of the following which contains data requested from or written to a data store:

- (a)'transit_preplanned_responses_for_incidents'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the input requesting output of the store contents is received, read the data from the store of transit predefined responses;
- (c) when (c) has been successfully completed, generate the output to the transit fleet manager identified above;
- (d) when the input in (a) contains new transit predefined responses data, update the store, overwriting any old data as necessary;
- (e) be responsible for the management of the data in the store of preplanned responses, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 253 of 475

4.5.1 Assess Transit Driver Performance

Input Flows

transit_driver_performance_considerations

Output Flows

transit_driver_performance
transit_driver_performance_data

Description:

Overview: This process shall be responsible for assessing 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 make use of the details of any moving violations or accidents, supervisor comments, government regulations, and company policies. The data shall be sent to the process by another process that provides the interface to a local data store, each time that the store is updated with driver performance data.

Data Flows: The input data flow is solicited as are both the output flows, and are used as follows:

- (a)'transit_driver_performance_considerations', contains data requested from a data store;
- (b)'transit_driver_performance', contains data written to a data store;
- (c)'transit_driver_performance_data', contains data sent to another process.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flow listed above;
- (b) when the input is received, analyze the data it contains and generate the two output flows identified above.

PROCESS SPECIFICATIONS

Page 254 of 475

4.5.2 Assess Transit Driver Availability

Input Flows

transit_driver_availability_considerations

Output Flows

transit_driver_availability

transit_driver_availability_data

Description:

Overview: This process shall be responsible for assessing 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 the process by another process that provides the interface to a local data store, each time that the store is updated with driver availability data.

Data Flows: The input data flow is solicited as are both the output flows, and are used as follows:

- (a)'transit_driver_availability_considerations', contains data requested from a data store;
- (b)'transit_driver_availability', contains data written to a data store;
- (c)'transit_driver_availability_data', contains data sent to another process.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flow listed above;
- (b) when the input is received, analyze the data it contains and generate the two output flows identified above.

PROCESS SPECIFICATIONS

Page 255 of 475

4.5.3 Access Transit Driver Cost Effectiveness

Input Flows

transit_driver_cost_effectiveness_considerations

Output Flows

transit_driver_cost_effectiveness

transit_driver_cost_effectiveness_data

Description:

Overview: This process shall be responsible for assessing the transit driver's cost effectiveness when carrying out previous work assignments. The process shall carry out 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 the process by another process that provides the interface to a local data store, each time that the store is updated with driver cost effectiveness data.

Data Flows: The input data flow is solicited as are both the output flows, and are used as follows:

- (a)'transit_driver_cost_effectiveness_considerations', contains data requested from a data store;
- (b)'transit_driver_cost_effectiveness', contains data written to a data store;

(c)'transit_driver_cost_effectiveness_data', contains data sent to another process.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flow listed above;
- (b) when the input is received, analyze the data it contains and generate the two output flows identified above.

PROCESS SPECIFICATIONS

Page 256 of 475

4.5.4 Assess Transit Driver Eligibility

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

Description:

Overview: This process shall be responsible for assessing 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 the process by another process that provides the interface to a local data store, each time that the store is updated with driver eligibility data, or produced as the result of analysis work carried out by other processes in the Manage Traffic function.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

- (a)'transit_driver_eligibility_considerations', which contains data requested from a data store;
- (b)'transit_driver_eligibility', which contains data written to a data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flows listed above;
- (b) when any of the input flows are received, generate the outputs identified above.

PROCESS SPECIFICATIONS

Page 257 of 475

4.5.5 Generate Transit Driver Route Assignments

Input Flows

paratransit_services_for_transit_drivers
transit_driver_eligibility_data
transit_driver_route_assignment_considerations
transit_driver_route_data
transit_services_for_transit_drivers
transit_vehicle_availability

Output Flows

ttd_route_assignments

Description:

Overview: This process shall be responsible for assigning transit drivers to transit schedules. It shall base its assignments upon the transit driver's eligibility, route assignment preferences, seniority and transit vehicle availability. The transit driver's eligibility, route preferences and 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 process shall provide its outputs to the transit driver in both audible and visual forms. It shall enable the visual output to be in either hardcopy, or as a display.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;

- (b) when either of the inputs containing eligibility or route assignment consideration data are received, generate the output to the transit driver identified above;
- (c) when any of the other inputs is received, load the data into a local data store for use in future route assignment calculations;
- (d) use the appropriate mechanism(s) such as RDBMS, to write data to and read data from the store identified above.

PROCESS SPECIFICATIONS

Page 258 of 475

4.5.6 Update Transit Driver Information

Input Flows

ftd_information_updates

Output Flows

transit_driver_consideration_inputs

Description:

Overview: This process shall be responsible for providing 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 process shall support inputs from the transit driver in both manual and audio forms.

Data Flows: The input data flows are unsolicited and the output flow is solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow from the transit driver listed above;
- (b) when the input is received, generate the output data flow identified above.

PROCESS SPECIFICATIONS

Page 259 of 475

4.5.7 Report Transit Driver Information

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
tftm_transit_driver_information

Description:

Overview: This process shall be responsible for providing 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 process shall support inputs from the transit fleet manager in both manual and audio form, and shall provide its outputs in both audible and visual forms. It shall enable the visual output to be in either hardcopy, or as a display.

Data Flows: The input data flows are unsolicited and the output flows are solicited with the exception of the following which is the result of data being sent to another process:

- (a)'transit_driver_information_output'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows from the transit fleet manager listed above;
- (b) when either of the inputs is received, output the appropriate data flow to the process that provides the interface to the store of transit driver information identified above;
- (c) if the input flow requested output of driver information, continuously monitor for receipt of the data flow sent in response identified above that shall contain the requested data;
- (d) when the input data flow in (c) is received, generate the appropriate output flow to the transit fleet manager, containing the requested data.

PROCESS SPECIFICATIONS

Page 260 of 475

4.5.8 Provide Transit Driver Information Store Interface

Input Flows

transit_driver_availability
transit_driver_consideration_inputs
transit_driver_consideration_updates
transit_driver_cost_effectiveness
transit_driver_eligibility
transit_driver_information
transit_driver_information_output_request
transit_driver_performance

Output Flows

transit_driver_availability_considerations
transit_driver_cost_effectiveness_considerations
transit_driver_eligibility_considerations
transit_driver_information_output
transit_driver_performance_considerations
transit_driver_route_assignment_considerations

Description:

Overview: This process shall be responsible for providing the read and write interface to the store of transit driver information. The interface provided by the process shall enable the contents of the store to be updated with inputs received from the transit driver and transit fleet manager via other processes and inputs resulting from analysis work on 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 inputs 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 are received.

Data Flows: The input data flows are unsolicited and the output flows is solicited with the exception of the following:

- (a)'transit_driver_information'- which contains data written to a data store;
- (b)'transit_driver_availability'- which contains data received as a result of output being sent to the availability analysis process;
- (c)'transit_driver_cost_effectiveness'- which contains data received as a result of output being sent to the cost effectiveness analysis process;
- (d)'transit_driver_eligibility'- which contains data received as a result of output being sent to the eligibility analysis process;
- (e)'transit_driver_performance'- which contains data received as a result of output being sent to the performance analysis process.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when either the input or update data flow is received, load the data into the store of transit driver information and send the data to the appropriate analysis process;
- (c) when the input containing the results of the availability, cost effectiveness, eligibility and performance analysis is received, again load in into the store of transit driver information and send the data to the driver route assignment process;
- (d) when the input containing the request for output of the current store data is received from the transit fleet manager interface process, read the data from the store and send it to the requesting process;
- (e) use the appropriate mechanism(s) such as RDBMS, to write data to and read data from the store identified above.

PROCESS SPECIFICATIONS

PROCESS SPECIFICATIONS

Page 261 of 475

Page 262 of 475

4.6.1 Detect Transit User on Vehicle

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

Description:

Overview: This process shall be responsible for the detection of embarking transit users on-board a transit vehicle and for reading data from the payment instrument that they are carrying. The process shall be able to provide an image of all transit users which shall be used for violation processing of those that do not have a payment instrument or whose transit fare transaction fails. It shall be capable of obtaining 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.

Data flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the transit user tag data input flow is received, generate the transit user tag identity output flow identified above;
- (c) when the flow requesting an image of the transit use is received, if necessary convert the video data in the flow from the transit user into a digital form, and output the digitized image in the transit user vehicle image data flow;
- (d) if the input flow in (c) is not received, discard the video image data in the flow from the transit user;
- (e) all input and output flows must be encrypted in such a way that it is not possible to determine the credit identity being transmitted using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 263 of 475

4.6.2 Determine Transit User Needs on Vehicle

Input Flows

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

Description:

Overview: This process shall be responsible for the determination of 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 that this shall be achieved, 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 another process for loading into a data store.

Data flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the transit user tag identity input flow is received, continuously monitor for receipt of the flow with the other transit user information;
- (c) when both the flows in (c) have been received, use the vehicle location and transit services inputs to generate the output flows identified above;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the transit user's payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 264 of 475

4.6.3 Determine Transit Fare on Vehicle

Input Flows

transit_fares_for_vehicle
transit_user_vehicle_ride

Output Flows

transit_user_vehicle_fare

Description:

Overview: This process shall be responsible for the calculation of 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 another process in a 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:
(a)'transit_fares_for_vehicle'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the transit user ride data input flow is received, generate the output data flow identified above, using the data in the store of transit fares for vehicles;
- (c) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the transit user's payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 265 of 475

4.6.4 Manage Transit Fare Billing on Vehicle

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

Description:

Overview: This process shall be responsible for managing the processing of transit user fare payments on-board a transit vehicle. The process shall be provided with information about the fare that is to be paid and the method of payment being 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 first shall be used by the process for individual transactions, such as those in demand responsive transit type operations where value/volume ratios are high. It shall be performed by the process interactively, on a individual transaction basis, sending transit user fare payment data to processes in the Provide Electronic Payment Services function for the 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 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 it has reached a convenient point on its route. Successful completion of the batch processing shall be notified to the transit vehicle driver. In either mode of operation, a record of the status of all transit fare processing shall be sent to another process for storage in a fare collection database.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) if the fare processing mode is not set, then request it as input from the transit vehicle driver;
- (c) when confirmation of the mode is received, and at the end of each service, request an update to the store of bad tag data, completely overwriting the existing data with the new data;
- (d) when the transit user fare input data is received, check the payment information against the store of bad tag data;
- (e) if a match is found in (d), output a transaction failed message to the transit user;
- (f) if no match is found in (d), generate the necessary outputs identified above that are consistent with the mode of processing being employed;
- (g) when confirmation of an interactive mode transaction is received, pass on the result to the transit user;
- (h) transmit the fare transaction data collected in batch mode when an instruction is received from the transit vehicle driver, the vehicle reaches the end of its current service, or the store of fare data becomes full;
- (i) when confirmation of successful completion of batch mode fare processing is received, clear the fare data from the store and inform the transit vehicle driver;

User Service Requirements:

- (j) if the batch mode fare transaction fails, then inform the transit vehicle driver;
- USR = 3.0; (k) all input and output flows must be encrypted in such a way that it is not possible to determine

USR = 3.1; the transit user's payment information being transmitted, using any form of digital or analog

USR = 3.1.2; techniques.

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;

PROCESS SPECIFICATIONS

PROCESS SPECIFICATIONS

Page 266 of 475

Page 267 of 475

4.6.5 Provide Transit User Interface on Vehicle

Input Flows

ftu_destination_on_vehicle
 ftu_other_services_vehicle_request
 other_services_vehicle_response
 transit_user_vehicle_credit_identity
 transit_user_vehicle_payment_response
 transit_vehicle_advanced_payment_response
 transit_vehicle_location

Output Flows

other_services_vehicle_request
 transit_user_advanced_payment_on_vehicle
 transit_user_vehicle_information
 transit_vehicle_advanced_payment_request
 ttu_other_services_vehicle_confirmed
 ttu_vehicle_payment_confirmed

Description:

Overview: This process shall be responsible for providing the interface for the transit user on-board a transit vehicle. The interface supported by the process shall enable the transit user to specify the required destination of a transit service ride and request other (yellow pages) type services. The process shall be capable of prompting the transit user for information necessary for the transaction that has not been provided. The successes 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 process shall support inputs from the transit user 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 a variety of printed or displayed formats to serve all types of transit user.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs from the transit user are received, generate the appropriate outputs identified above, prompting the user for any information that has not been supplied;
- (c) when either of the response flows is received, generate the appropriate output to the transit user to indicate the success or failure of the requested transaction;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the transit user's payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 268 of 475

4.6.6 Update Transit Vehicle Fare Data

Input Flows

transit_vehicle_fare_data

Output Flows

transit_fares_for_vehicle

Description:

Overview: This process shall be responsible for providing 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data written to a data store:

- (a)'transit_fares'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above;
- (b) when the input flow is received, generate the output flow identified above to update the contents of the store of transit fares;
- (c) be responsible for the management of the data in the store of transit fares, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 269 of 475

4.6.7 Provide Transit Vehicle Passenger Data

Input Flows

transit_user_vehicle_processed_fare_data

transit_user_vehicle_ride_data

transit_vehicle_fare_collection_data

Output Flows

transit_vehicle_passenger_data

Description:

Overview: This process shall be responsible for providing passenger loading and fare statistics data to other ITS functions. The process shall send the data automatically at regular periodic intervals using that which it has collected in the store of fare transaction data from data sent to it by another process (in the Manage Transit function) on-board a transit vehicle.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from and written to a data store:

- (a)'transit_vehicle_fare_collection_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, write the data into the store of collected transit fare data;
- (c) at periodic intervals of not less than once per day, read all the data from the store and send it to the other processes in the Manage Transit function using the passenger data flow identified above;
- (d) be responsible for the management of the data in the store of collected transit fare data, using the appropriate mechanism(s) such as RDBMS, for storing the data;
- (e) the data contained in the store shown above shall not contain any reference to the transit users' identities or credit payment information.

PROCESS SPECIFICATIONS

Page 270 of 475

4.6.8 Manage Transit Vehicle Advanced Payments

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

Description:

Overview: This process shall be responsible for acting as the interface for advanced payment of tolls and parking lot charges from the transit user. The process shall enable requests for these advanced payments to 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 are returned to the transit vehicle from which they were made.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the request input data flow is received from the transit user interface process, generate the request output data flow identified above;
- (c) as a result of (b), continuously monitor for receipt of the confirmation flow identified above;
- (d) when the flow in (c) is received, output the response flow identified above to the transit user interface process;
- (e) all input and output flows must be encrypted in such a way that it is not possible to determine the transit user's payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 271 of 475

4.7.1.1 Provide Transit User Roadside Data Interface

Input Flows

ftu_transit_information_request
transit_information_for_transit_users
transit_services_roadside_data

Output Flows

transit_information_transit_user_request
ttu_transit_information

Description:

Overview: This process is responsible for providing public transit information at roadside locations. The process shall support these locations being transit vehicle stops, although they could be at other locations to provide general public transit information. The process shall support inputs from the transit user 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 a variety of printed or displayed formats to serve all types of transit user.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;

- (b) when the input from the transit user is received, read the data in the local data store to see if the information is already present;
- (c) if the required information is not present, generate the appropriate output identified above;
- (d) as a result of (c) continuously monitor for the receipt of the data flow containing the information for transit users;
- (e) when the data flow in (d) is received, or if the data was read from the local data store, generate the output flow to the transit user with the requested data;
- (f) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the local data store identified above.

PROCESS SPECIFICATIONS

Page 272 of 475

4.7.1.2 Provide Transit User Roadside Communications Interface

Input Flows

transit_information_transit_user_request
transit_services_for_travelers
transit_vehicle_arrival_time

Output Flows

transit_information_for_transit_users
transit_services_roadside_data
transit_services_travelers_request

Description:

Overview: This process shall be responsible for providing the interface through which a roadside unit at a transit stop can communicate with the transit management center (TRM). The process shall enable the roadside unit to obtain information about 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 passed onto the transit user interface process and loaded into a local store for future use.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'transit_vehicle_arrival_time';
- (b)'transit_information_transit_user_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:

- (a)'transit_services_for_travelers'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'transit_information_for_transit_users';
- (b)'transit_services_roadside_data';
- (c)'transit_services_travelers_request'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) upon receipt of the first input data flow shown above, send the first of the output flows and also send the data to the local store using the second of the output flows;
- (c) be responsible for the management of the data in the store of transit services roadside data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 273 of 475

4.7.1.3 Provide Transit User Roadside Vehicle Data Interface

Input Flows

transit_services_roadside_data
transit_vehicle_user_data

Output Flows

ttu_transit_vehicle_information

Description:

Overview: This process shall be responsible for providing the roadside (transit stop) interface through which transit users receive information about an approaching or already arrived transit vehicle. 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 deleted it from the local store. The process shall provide its outputs to the transit user in both audible and visual forms. It shall enable the visual output to be in a variety of printed or displayed formats to serve all types of transit user.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'transit_vehicle_user_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:
(a)'transit_services_roadside_data

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'transit_services_roadside_data';
(b)'ttu-transit_vehicle_information'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input flow listed above;
(b) when the data input flow is first received, calculate the time before the transit vehicle will arrive using the difference between the transit vehicle arrival time and time now;
(c) output the data calculated in (b) to the transit user and store the data locally to provide a continuous source of data;
(d) once the data shows that the transit vehicle is leaving the stop, clear the output for that vehicle and delete its data from the local store;
(e) if the input ceases to be received from the transit vehicle, maintain the output using the locally stored data;
(f) be responsible for the management of the data in the store of the pollution data log, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 274 of 475

4.7.2.1 Detect Transit User at Roadside

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

Description:

Overview: This process shall be responsible for the detection of transit users embarking at a roadside transit stop and for reading data from the payment instrument that they are carrying. The process shall be able to provide an image of all transit users which shall be used for violation processing of those that do not have a payment instrument or whose transit fare transaction fails. It shall be capable of obtaining 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.

Data flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the input flows listed above;
(b) when the transit user tag data input flow is received, generate the transit user tag identity output flow identified above;
(c) when the flow requesting an image of the transit use is received, if necessary convert the video data in the flow from the transit user into a digital form, and output the digitized image in the transit user vehicle image data flow;
(d) if the input flow in (c) is not received, discard the video image data in the flow from the transit user;
(e) all input and output flows must be encrypted in such a way that it is not possible to determine the credit identity being transmitted using any form of digital or analog techniques.

4.7.2.2 Determine Transit User Needs at Roadside**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

Description:

Overview: This process shall be responsible for the determination of the transit user's travel routing based on the location of the roadside transit stop from which the route request is being made and the user's destination. 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 that this shall be achieved, the process shall have 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 another process for loading into a data store.

Data flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the transit user tag identity input flow is received, continuously monitor for receipt of the flow with the other transit user information;
- (c) when both the flows in (c) have been received, use the vehicle location and transit services inputs to generate the output flows identified above;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the transit user's payment information being transmitted, using any form of digital or analog techniques.

4.7.2.3 Determine Transit Fare at Roadside**Input Flows**

transit_fares_for_roadside
 transit_user_roadside_ride

Output Flows

transit_user_roadside_fare

Description:

Overview: This process shall be responsible for the calculation of 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 another process in a 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

- (a)'transit_fares_for_roadside'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the transit user ride data input flow is received, generate the output data flow identified above, using the data in the store of transit fares for vehicles;
- (c) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the transit user's payment information being transmitted, using any form of digital or analog techniques.

4.7.2.4 Manage Transit Fare Billing at Roadside

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

Description:

Overview: This process shall be responsible for generating the data necessary to enable the financial transaction between the transit user and the transit provider to be completed on at the roadside, i.e. a transit stop. The capability to accept and process current transit passenger fare collection information shall be provided by the process. The process shall always 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 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the transit user fare input is received, generate the necessary outputs identified above that are consistent with the type of processing being employed;
- (c) all input and output flows must be encrypted in such a way that it is not possible to determine the transit user's payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 278 of 475

4.7.2.5 Provide Transit User Roadside Fare Interface

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

Description:

Overview: This process shall be responsible for providing the interface for the transit user at the roadside, i.e. a transit stop. The interface supported by the process shall enable the transit user to specify the required destination of a transit service ride and request other (yellow pages) services. The process shall be capable of prompting the transit user for information necessary for the transaction that has not been provided. The successes 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 process shall support inputs from the transit user 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 a

variety of printed or displayed formats to serve all types of transit user.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs from the transit user are received, generate the appropriate outputs identified above, prompting the user for any information that has not been supplied;
- (c) when either of the response flows is received, generate the appropriate output to the transit user to indicate the success or failure of the requested transaction;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the transit user's payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 279 of 475

4.7.2.6 Update Roadside Transit Fare Data

Input Flows

transit_roadside_fare_data

Output Flows

transit_fares_for_roadside

Description:

Overview: This process shall be responsible for providing 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data written to a data store:

- (a)'transit_fares_for_roadside'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above;
- (b) when the input flow is received, generate the output flow identified above to update the contents of the store of transit fares;
- (c) be responsible for the management of the data in the store of transit fares, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 280 of 475

4.7.2.7 Provide Transit Roadside Passenger Data

Input Flows

transit_roadside_fare_collection_data

transit_user_roadside_processed_fare_data

transit_user_roadside_ride_data

Output Flows

transit_roadside_passenger_data

Description:

Overview: This process shall be responsible for providing passenger loading and fare statistics data to other ITS functions. The process shall send the data automatically at regular periodic intervals using that which it has collected in the store of fare transaction data from data sent to it by another process (in the Manage Transit function) at the roadside, i.e. a transit stop.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from and written to a data store:

- (a)'transit_fare_collection_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, write the data into the store of collected transit fare data;
- (c) at periodic intervals of not less than once per day, read all the data from the store and send it to the other processes in the Manage Transit function using the passenger data flow identified above;
- (d) be responsible for the management of the data in the store of collected transit fare data,

using the appropriate mechanism(s) such as RDBMS, for storing the data;
(e) the data contained in the store shown above shall not contain any reference to the transit users' identities or credit payment information.

PROCESS SPECIFICATIONS

Page 281 of 475

5.1.1 Identify Emergencies from Inputs

Input Flows

cf_hazmat_route_information
emergency_request_driver_details
emergency_request_kiosk_traveler_details
emergency_request_personal_traveler_details
emergency_request_vehicle_details
fets_emergency_telephone_service_identity
fets_incident_details
foec_emergency_telephone_service_identity
foec_incident_details
incident_alert
transit_emergency_data
transit_incident_details

Output Flows

identified_emergency

Description:

Overview: This process shall be responsible for providing the capability for existing emergency centers to receive the calls, determine response requirements, and route distress calls to predesignated responding agencies. This process shall provide the identified emergency information in a standard format as required. 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'cf_hazmat_route_information';
- (b)'emergency_request_driver_details';
- (c)'emergency_request_kiosk_traveler_details';
- (d)'emergency_request_personal_traveler_details';
- (e)'emergency_request_vehicle_details';
- (f)'fets-incident_details';
- (g)'fets-emergency_telephone_service_identity';
- (h)'foec-incident_details';
- (i)'foec-emergency_telephone_service_identity';
- (j)'incident_alert'
- (k)'transit_emergency_data';
- (l)'transit_incident_details'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'identified_emergency'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when any of the inputs are received, the process shall immediately perform an analysis of the data to produce the output as a set of data in a standard format;
- (c) the data format produced in (b) shall include a classification of the level of confidence that can be assigned to the probability that the data is accurate, i.e. it relates to a'real'emergency and the information contained is correct.

PROCESS SPECIFICATIONS

Page 282 of 475

PROCESS SPECIFICATIONS

Page 283 of 475

5.1.2 Determine Coordinated Response Plan

Input Flows

emergency_service_allocation_data
identified_emergency

Output Flows

emergency_response_data_for_communications
emergency_response_data_for_management
emergency_service_allocation_data_request
toec_emergency_telephone_service_identity
toec_incident_details

Description:

Overview: This process shall be responsible for providing the capability to receive the identified emergency message. It shall be capable of analyzing the data that the message contains to determine which emergency service response plans are to be sent to other processes for implementation. The process shall obtain the plans from the process that manages the store of emergency allocation criteria. The response plan and other data shall also be sent by the process to the other emergency centers (other EM) for information and possible action.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'identified_emergency'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:
(a)'emergency_service_allocation_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'emergency_response_data_for_communications';
- (b)'emergency_response_data_for_management';
- (c)'emergency_service_allocation_data_request';
- (d)'toec-incident_details';
- (e)'toec-emergency_telephone_service_identity'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the input is received, the process will determine the response from the data requested from the process that manages the store emergency service allocation criteria;
- (c) when (b) is complete, the data shall be sent to both the emergency management and communications processes.

PROCESS SPECIFICATIONS

Page 284 of 475

5.1.3 Communicate Emergency Status

Input Flows

detailed_emergency_status
emergency_response_data_for_communications
emergency_service_action_log
emergency_service_log_output_request
incident_details_request
incident_information_request

Output Flows

emergency_request_driver_acknowledge
emergency_request_kiosk_traveler_acknowledge
emergency_request_personal_traveler_acknowledge
emergency_request_vehicle_acknowledge
emergency_service_log_output
emergency_vehicle_operational_data
incident_details
incident_information
tets_incident_acknowledge
transit_incident_coordination_data

Description:

Overview: This process shall be responsible for receiving the emergency service response plans and the status of their implementation for dissemination to other ITS functions. It shall also be capable of reading data about emergency responses from the emergency services action log. All data

shall be communicated by the process to travelers, drivers and other ITS functions in proper and standard formats. In the case of in-vehicle, personal traveler and transit emergencies the data shall be sent as soon as new status or plan data is received. Emergencies that are notified by the emergency telephone system or E911 operators, shall be sent data only when the response plan data is first received. The process shall also be capable of extracting data from the emergency service action log on request from other processes in other ITS functions, and from the emergency services operator.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

(a)'emergency_service_action_log'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when either the input of emergency response plan data or detailed emergency status is received, read any associated data from the emergency service action log data store and generate the outputs to the vehicle, traveler and Manage Transit function identified above;
- (c) if the input in (b) is the first for an emergency, and it was notified by the emergency telephone or E911 operator, send the acknowledge message identified above;
- (d) when either of the input flows requesting information is received, read the data from the emergency services action log, and send it in the data flow of incident information identified above to the source of the request;
- (e) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the emergency service action log store identified above.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 285 of 475
Page 286 of 475

5.1.4 Manage Emergency Response

Input Flows

cf_hazmat_vehicle_information
emergency_response_data_for_management
emergency_service_allocation_override
emergency_vehicle_acknowledge
emergency_vehicle_dispatch_status
incident_response_clear
transit_coordination_data

Output Flows

cf_hazmat_request
detailed_emergency_status
emergency_service_action_log
emergency_service_allocations
emergency_vehicle_dispatch_failure
emergency_vehicle_dispatch_request
emergency_vehicle_incident_details
incident_response_status

Description:

Overview: This process shall be responsible for providing the capability for existing emergency centers to receive the calls, determine response requirements, and route distress calls 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

- (a)'emergency_service_action_log', which contains data written to a data store;
- (b)'cf_hazmat_vehicle_information', which is received as a result of output to a process in the Manage Commercial Vehicles function.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the input of emergency response plan data is received, generate the output data flows identified above, and create an initial entry in the emergency response action log data store;
- (c) when other inputs are received, update the data for the emergency to which they relate in the emergency service action log data store;
- (d) if the emergency vehicle dispatch status indicates a failure, send the data to the action log and to the emergency services operator interface process;
- (e) use the appropriate mechanism(s) such as RDBMS, to write data to the store identified above.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 287 of 475
Page 288 of 475

5.1.5 Manage Emergency Service Allocation Store

Input Flows

emergency_service_allocation_criteria
 emergency_service_allocation_data_output_request
 emergency_service_allocation_data_request
 emergency_service_allocation_data_updates

Output Flows

emergency_service_allocation_data
 emergency_service_allocation_data_output

Description:

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

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'emergency_service_allocation_data_output_request';
- (b)'emergency_service_allocation_data_request';
- (c)'emergency_service_allocation_data_updates'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval:

- (a)'emergency_service_allocation_criteria'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'emergency_service_allocation_criteria';
- (b)'emergency_service_allocation_data';
- (c)'emergency_service_allocation_data_output'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the inputs are received, generate the appropriate outputs identified above to read or write data from and to the data store of emergency services allocation criteria;
- (c) be responsible for the management of the data in the store of the emergency allocation criteria data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 289 of 475

5.2 Provide Operator Interface for Emergency Data

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

Description:

Overview: This process shall be responsible for providing 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 request output of the log of emergency service actions. It shall also enable the output of a message showing the failure of an emergency vehicle dispatch 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. It shall enable the visual output to be either in hardcopy, or as a display, and shall provide an audible output to accompany the emergency vehicle dispatch failure message.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows from the emergency services operator and the emergency vehicle dispatch failure flow listed above;
- (b) when the inputs from emergency services operator are received, generate the appropriate output data flows identified above to other processes;
- (c) when the responses to the flows generated in (b) are received, generate the appropriate outputs to the emergency services operator;
- (d) when the emergency vehicle dispatch failure flow is received, generate the appropriate output flow listed above, overriding all other output flows..

PROCESS SPECIFICATIONS

PROCESS SPECIFICATIONS

Page 290 of 475

Page 291 of 475

5.3.1 Select Response Mode

Input Flows

emergency_vehicle_dispatch_request
emergency_vehicle_incident_details
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

Description:

Overview: This process shall be responsible for selecting 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 obtain data about which vehicles are available by reading data from the 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).

Data Flows: Input data flows are unsolicited and output flows are solicited with the exception of the following which represent a data query response and updates command for a store (managed by another process):

- (a)'emergency_vehicle_status_data_for_responses'
- (b)'emergency_vehicle_status_data_update.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the two vehicular inputs identified above;
- (b) when the input of the emergency vehicle dispatch request data flow is received, request data from the process providing the interface for the store of emergency vehicle status;
- (c) use the data obtained in (b) to determine the vehicle(s) necessary to provide the appropriate response to the incident;
- (d) when the vehicle(s) have been determined in (c), generate the output identified above to dispatch each vehicle;
- (e) send changed emergency vehicle status data back to the store interface process to reflect the new vehicle status resulting from output of the dispatch data;
- (f) when all the required vehicle type(s) and numbers have been dispatched, send the emergency vehicle dispatch status data flow to the process responsible for managing emergency responses, showing that the dispatch was successful;
- (g) if no vehicles of the required type(s) are available, send the emergency vehicle dispatch status data flow to the process responsible for managing emergency responses, showing that the dispatch has failed for the particular vehicle type(s).

PROCESS SPECIFICATIONS

Page 292 of 475

5.3.2 Dispatch Vehicle

Input Flows

emergency_vehicle_dispatch_data
emergency_vehicle_driver_inputs
emergency_vehicle_route
emergency_vehicle_status_data_for_dispatch

Output Flows

emergency_vehicle_driver_outputs
emergency_vehicle_green_wave
emergency_vehicle_route_request

Description:

Overview: This process shall be responsible for providing selected emergency vehicles and drivers details of an emergency incident to which they must respond. The data about the incident shall be sent to the process by the process in the Manage Emergency Services function responsible for selecting the vehicles for the response. The process shall also be responsible for requesting emergency vehicle routing from the Provide Driver and Traveler Services function and sending details to the Manage Traffic function to enable a 'green wave' to be provided for the vehicle(s). The data for the emergency vehicle driver shall be sent to the driver interface process.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a process that manages the store of vehicle status data:

- (a)'emergency_vehicle_status_data_for_dispatch'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the dispatch and status data input flows listed above;
- (b) when the flow in (a) is received, generate the outputs identified above to request the emergency vehicle route and provide the driver with information about the incident, monitoring for the receipt of any reply data;
- (c) when the emergency vehicle route data is received, generate the emergency vehicle green wave data and send it to the Manage Traffic function;
- (d) use an appropriate mechanism(s) (e.g. SQL), to request data from the vehicle status store manager identified above.

PROCESS SPECIFICATIONS

Page 293 of 475

5.3.3 Track Vehicle

Input Flows

emergency_vehicle_status_data_for_priority
vehicle_location_for_emergency_services

Output Flows

emergency_vehicle_preemptions
emergency_vehicle_tracking_data

Description:

Overview: This process shall be responsible for providing means for processing information on the location of all emergency vehicles. The process shall send this data to the store of emergency vehicle status. 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 local priority at traffic signals.

Data Flows: The input data flow is unsolicited and both output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flows listed above;
- (b) when the location flow is received and is different to the previous value, generate the data flow to the emergency vehicle status store interface process, adding the current data and time to the received location data;
- (c) when the status flow is received, if it shows that the vehicle is on its way to an emergency incident, then simultaneously with (b) output the data flow requesting local vehicle priority to the indicator control processes in the Manage Traffic function.

PROCESS SPECIFICATIONS

Page 294 of 475

5.3.4 Assess Response Status

Input Flows

emergency_driver_status_update
emergency_vehicle_route
emergency_vehicle_status_data_for_assessment

Output Flows

emergency_vehicle_acknowledge
emergency_vehicle_status_data_needed
emergency_vehicle_status_data_update

Description:

Overview: This process shall be responsible for assessing 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 drivers received as inputs to their interface process. The process shall send the results of the assessment to the process responsible for managing emergency responses and update the store of vehicle status.

Data Flows: All input data flows are unsolicited and the output flow is solicited with the exception of the following which contains data requested from a process managing the data store:
(a)'emergency_vehicle_status_data_for_assessment';

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the input from the driver is received, use the data it contains to request the relevant emergency vehicle status data from the store interface process;
- (c) update the status using the input received from the driver and generate the flow to the process that manages emergency responses;
- (d) send the revised emergency vehicle status data to the store interface process.

PROCESS SPECIFICATIONS

Page 295 of 475

5.3.5 Provide Emergency Vehicle Driver Interface

Input Flows

emergency_vehicle_driver_outputs
fevd_emergency_dispatch_acknowledge
fevd_incident_status

Output Flows

emergency_driver_status_update
emergency_vehicle_driver_inputs
tevd_emergency_dispatch_order

Description:

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

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received from the emergency vehicle driver, generate the appropriate output flows identified above and send them to the dispatch and response status monitoring processes;
- (c) when input is received from the dispatch process, generate the corresponding output to the emergency vehicle driver.

PROCESS SPECIFICATIONS

Page 296 of 475

5.3.6 Maintain Vehicle Status

Input Flows

emergency_vehicle_status_data
emergency_vehicle_status_data_change
emergency_vehicle_status_data_needed
emergency_vehicle_status_data_request
emergency_vehicle_status_data_update
emergency_vehicle_tracking_data

Output Flows

emergency_vehicle_status_data_for_assessment
emergency_vehicle_status_data_for_dispatch
emergency_vehicle_status_data_for_priority
emergency_vehicle_status_data_for_responses

Description:

Overview: This process shall be responsible for maintaining a store of all emergency vehicles' current status. 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 a vehicle's status to the process responsible for vehicle tracking for as long as it is on its way to an incident, to enable local vehicle priority to be given at intersections.

Data Flows: The input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for the receipt of the input flows listed above;
- (b) when either the data request or data needed flows is received, read the requested data from the store and send it to the requesting process;
- (c) when either of the data flows containing updated or changed vehicle status data is received, load the new data into the data store, overwriting the existing data for the vehicle;
- (d) when the new status data shows that a vehicle is on its way to an incident, output the vehicle status data to the processes responsible for vehicle tracking and vehicle dispatch;
- (e) when the new status data shows that a vehicle is no longer on its way to an incident, output the status data to the process responsible for vehicle tracking, but do not send any further status updates until the condition in (d) is again satisfied;
- (f) respond to the data queries in (b) using an appropriate mechanism (e.g. SQL) to provide individual records for one or more emergency vehicles;
- (g) accept the record updates in (c) and use an appropriate mechanism(s) such as RDBMS, to maintain the data store identified above.

PROCESS SPECIFICATIONS

Page 297 of 475

5.4.1 Process TM Detected Violations

Input Flows

enforcement_data_for_TM
fdmv_traffic_violation_state_identity
fdmv_traffic_violation_vehicle_registration
hov_lane_violation
vehicle_pollution_alert

Output Flows

tdmv_traffic_violation_identity_code
tdmv_traffic_violation_vehicle_license
tea_traffic_violation_data

Description:

Overview: This process shall be responsible for processing the details of high occupancy vehicle (hov) lane use and pollution violations reported by the Manage Traffic function. The process shall use the parameters in the store of traffic management (TM) violation 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. The maintenance of the TM store shall also be the responsibility of this process.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'hov_lane_violation';
- (b)'vehicle_pollution_alert'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from a local data store:

- (a)'enforcement_data_for_TM'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to the DMV terminator:

- (a)'fdmv-traffic_violation_state_identity';
- (b)'fdmv-traffic_violation_vehicle_registration'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'enforcement_data_for_TM';
- (b)'tdmv-traffic_violation_identity_code';
- (c)'tdmv-traffic_violation_vehicle_license';
- (d)'tea-traffic_violation_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the first or second unsolicited inputs shown above are received, use the data that they contain to generate the third and fourth solicited outputs shown above;
- (c) when the solicited input flows from the DMV terminator shown above have been received, use the other data received in (b) and the contents of the TM store to generate the fourth solicited output identified above;
- (d) be responsible for the management of the data in the store of traffic management violation data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 298 of 475

PROCESS SPECIFICATIONS

Page 299 of 475

5.4.2 Process Violations for Tolls

Input Flows

enforcement_data_for_tolls
fdmv_toll_violation_state_identity
fdmv_toll_violation_vehicle_registration
toll_violation_information

Output Flows

tdmv_toll_violation_identity_code
tdmv_toll_violation_vehicle_license
tea_toll_violation_data

Description:

Overview: This process shall be responsible for processing 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 data 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. The maintenance of the toll payment violation data store shall also be the responsibility of this process.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'toll_violation_information'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:
(a)'enforcement_data_for_tolls'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to the DMV terminator:
(a)'fdmv-toll_violation_state_identity';
(b)'fdmv-toll_violation_vehicle_registration'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'enforcement_data_for_tolls';
(b)'tdmv-toll_violation_identity_code';
(c)'tdmv-toll_violation_vehicle_license';
(d)'tea-toll_violation_data'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input flows listed above;
(b) when the unsolicited input shown above is received, use the data that it contains to generate the second and third solicited outputs shown above;
(c) when the solicited input flows from the DMV terminator shown above have been received, use the other data received in (b) and the contents of the toll payment violation data store to generate the fourth solicited output identified above;
(d) be responsible for the management of the data in the store of toll payment violation data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 300 of 475
Page 301 of 475

5.4.3 Process Parking Lot Violations

Input Flows

enforcement_data_for_parking
fdmv_parking_lot_violation_state_identity
fdmv_parking_lot_violation_vehicle_registration
parking_lot_violation_information

Output Flows

tdmv_parking_lot_violation_identity_code
tdmv_parking_lot_violation_vehicle_license
tea_parking_violation_data

Description:

Overview: This process shall be responsible for processing 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 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. The maintenance of the store of parking lot violation data shall also be the responsibility of this process.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'parking_lot_violation_information'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:

(a)'enforcement_data_for_parking

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to the DMV terminator:

- (a)'fdmv-parking_lot_violation_state_identity';
- (b)'fdmv-parking_lot_violation_vehicle_registration'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'enforcement_data_for_parking';
- (b)'tdmv-parking_lot_violation_identity_code';
- (c)'tdmv-parking_lot_violation_vehicle_license';
- (d)'tea-parking_violation_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the unsolicited input shown above is received, use the data it contains to generate the second and third solicited outputs shown above;
- (c) when the solicited input flows from the DMV terminator shown above have been received, use the other data received in (b) and the contents of the parking lot payment violation data store to generate the fifth solicited output identified above;
- (d) be responsible for the management of the data in the store of parking lot payment violation data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 302 of 475

5.4.4 Process Fare Payment Violations

Input Flows

enforcement_data_for_fare_payment
fare_violation_information

Output Flows

tea_fare_payment_violation_data

Description:

Overview: This process shall be responsible for processing 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 data to process and send the data to the correct law enforcement agency. The maintenance of the store is also the responsibility of this process.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fare_violation_information'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:

- (a)'enforcement_data_for_fare_payment

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'tea-fare_payment_violation_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the unsolicited input shown above is received, use the data it contains and the contents of the store to generate the second output identified above;
- (c) be responsible for the management of the data in the store of the data about the enforcement of fare payment violations, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 303 of 475

5.4.5 Process Vehicle Fare Collection Violations

Input Flows

enforcement_data_for_vehicle_fare_collection
fare_collection_vehicle_violation_information

Output Flows

tea_fare_collection_vehicle_violation_data

Description:

Overview: This process shall be responsible for processing 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 fare payment violation data to process and send the data to the correct law enforcement agency. The maintenance of the store is also the responsibility of this process, using data that it receives from a central process common to all individual enforcement agency processes.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'fare_collection_vehicle_violation_information'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from a local data store:
(a)'enforcement_data_for_vehicle_fare_collection'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'tea-fare_collection_vehicle_violation_data'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input flows listed above;
(b) when the unsolicited input shown above is received, use the data it contains and the contents of the store to generate the second output identified above;
(c) be responsible for the management of the data in the store of vehicle fare collection violation data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 304 of 475

5.4.6 Process CV Violations

Input Flows

cv_violation_data
enforcement_data_for_cv
fdmv_cv_violation_state_identity
fdmv_cv_violation_vehicle_registration

Output Flows

tdmv_cv_violation_identity_code
tdmv_cv_violation_vehicle_license
tea_cv_violation_data

Description:

Overview: This process shall be responsible for processing 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 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. The maintenance of the store of commercial vehicle violation data shall also be the responsibility of this process.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'cv_violation_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from a local data store:
(a)'enforcement_data_for_cv

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to the DMV terminator:
(a)'fdmv-cv_violation_state_identity';
(b)'fdmv-cv_violation_vehicle_registration'.

Solicited Output Processing: This process shall provide the following output flows as a result of

the above inputs being received:

- (a)'enforcement_data_for_cv';
- (b)'tdmv-cv_violation_identity_code';
- (c)'tdmv-cv_violation_vehicle_license';
- (d)'tea-cv_violation_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the unsolicited input shown above is received, use the data it contains to generate the second and third solicited outputs shown above;
- (c) when the solicited input flows from the DMV terminator shown above have been received, use the other data received in (b) and the contents of the store of commercial vehicle (cv) violation data to generate the fourth solicited output identified above;
- (d) be responsible for the management of the data in the store of commercial vehicle violation data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 305 of 475

5.4.7 Process Roadside Fare Collection Violations

Input Flows

enforcement_data_for_roadside_fare_collection
fare_collection_roadside_violation_information

Output Flows

tea_fare_collection_roadside_violation_data

Description:

Overview: This process shall be responsible for processing the details of fare collection violations reported by the Manage Transit function that have taken place at the roadside, i.e. a transit stop. The process shall use the parameters in the store of fare payment violation data to process and send the data to the correct law enforcement agency. The maintenance of the store is also the responsibility of this process, using data that it receives from a central process common to all individual enforcement agency processes.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fare_collection_roadside_violation_information'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from a local data store:

- (a)'enforcement_data_for_roadside_fare_collection'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'tea-fare_collection_roadside_violation_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the unsolicited input shown above is received, use the data it contains and the contents of the store to generate the second output identified above;
- (c) be responsible for the management of the data in the store of roadside fare collection violation data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 306 of 475

5.5 Update Emergency Display Map Data

Input Flows

fmup_emergency_display_update
request_emergency_display_update

Output Flows

map_data_for_emergency_display
tmup_request_emergency_display_update

Description:

Overview: This process shall be responsible for providing 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 function.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'request_emergency_display_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to external functions:
(a)'fmup-emergency_display_update'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'tmup-request_emergency_display_update';
(b)'map_data_for_emergency_display'.

Functional Requirements: This process shall meet the following requirements:
(a) continuously monitor for the receipt of the unsolicited data flow shown above;
(b) when the data flow in (a) is received, generate the first solicited output data flow shown above and continuously monitor for receipt of the solicited input data flow shown above;
(c) when the flow in (b) is received, output the second solicited output data flow shown above;
(d) be capable of receiving the input data in a variety of formats and converting it into a single format suitable for use with the store of digitized map data;
(e) be responsible for the management of the data in the store of digitized map data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 307 of 475

6.1.1 Provide Trip Planning Information to Traveler

Input Flows

current_conditions
intermodal_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

intermodal_service_data_request
paratransit_route_response
paratransit_trip_request
parking_lot_data_request
request_prices
traveler_personal_trip_information
traveler_rideshare_request
traveler_trip_information
trip_information
trip_request

Description:

Overview: This process shall be responsible for obtaining 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 more than one mode 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 specialist 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, modal change points, etc. are known. The information provided to the traveler by the process shall be sufficient to enable the routing, modes and cost of the trip to be understood by the traveler, and to be used as the basis of a subsequent trip confirmation. This process shall exchange all input and output data from and to the traveler with the appropriate traveler interface process.

Data Flows: The traveler tip request input data flow is unsolicited. All output flows are solicited and will themselves generate input flows. The following input flows contain data requested from or written to data stores:

- (a)'trip_information', write only;
- (b)'trip_planning_parameters', read only.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the traveler trip request input flow listed above;
- (b) when the input flow in (a) is received, output data flows to other processes requesting various types of routes, rideshare information, demand responsive trip requests, according to the preferences and constraints in the traveler's trip request and the parameters governing trip selection contained in the read only data store identified above;
- (c) when the data has been returned, construct the trip and ensure that there are no breaks, i.e. where mode changes are involved, each segment begins and ends at a valid modal interchange point;
- (d) if any of the segments do not joint up, change the preferences and constraints and repeat (b) until a correct match is produced;
- (e) in parallel with (c) and (d) compute the total cost of the trip, including all tolls, parking lot charges, transit fares, and other costs;
- (f) when all calculations are complete, store the trip information in the local store for use if the traveler decides on confirmation and send the trip data to the process that provides the traveler

User Service Requirements:

defined above; interface using the traveler trip information output flow

- (g) use the appropriate mechanism(s) such as RDBMS, to retrieve data from and write data to the two data stores identified above.

USR = 1.0;
USR = 1.1;
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 = 1.1.3.2;
USR = 1.1.3.3;
USR = 1.1.3.3.1;
UAR = 1.1.3.3.2;
USR = 1.4;
USR = 1.4.0;
USR = 1.4.1;
USR = 1.4.1.3;
USR = 1.4.3;

USR = 1.4.3.1;

Output Flow Dynamics Assumptions: USR = 1.4.3.3;

USR = 2.0;

USR = 2.3; paratransit_route_response = paratransit_route_request;

USR = 2.3.1; paratransit_trip_request = $5/(60*60)*ITS_PTRANSIT_TRAVERS$;

USR = 2.3.1.2; parking_lot_data_request = $5/(60*60)*ITS_RS_TRAVERS$;

USR = 2.4; request_prices = $20/(60*60)*ITS_TRAVERS$;

USR = 2.4.3; intermodal_service_data_request = $5/(60*60)*ITS_TRAVERS$;

USR = 2.4.3.1; traveler_personal_trip_information = traveler_personal_trip_request;

traveler_rideshare_request = $5/(60*60)*ITS_RS_TRAVERS$;

traveler_trip_information = $20/(60*60)*ITS_TRAVERS$;

trip_information = $20/(60*60)*ITS_TRAVERS$;

trip_request = $5/(60*60)*ITS_TRAVERS$;

PROCESS SPECIFICATIONS

Page 308 of 475

PROCESS SPECIFICATIONS

Page 309 of 475

PROCESS SPECIFICATIONS

Page 310 of 475

6.1.2 Confirm Traveler's Trip Plan

Input Flows

intermodal_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

intermodal_service_confirm

paratransit_service_confirmation

parking_lot_reservation_request

traveler_payment_confirmation

traveler_payment_request

traveler_personal_payment_confirmation

traveler_personal_transaction_confirmation

traveler_rideshare_confirmation

traveler_transaction_confirmation

Description:

Overview: This process shall be responsible for the confirmation of a trip previously requested by a traveler and any financial transactions that this may require. The process shall base the trip confirmation upon information about the route, etc. created by the process responsible for trip planning and stored locally. The confirmation details shall be sent to specialist processes such as those responsible for demand responsive transit, ridesharing, etc. in order that reservations may be made for their services. The response to these reservation requests and to the payment transactions required by the route shall be sent to the traveler. This process shall exchange all input and output data from and to the traveler with the appropriate traveler interface process.

Data Flows: The traveler trip confirmation input data flow is unsolicited. All output flows are solicited and will themselves generate input flows. The input flows 'trip_information' and 'trip_planning_parameters' contain data requested data stores.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the traveler trip confirmation input flow listed above;
- (b) when the input flow in (a) is received, obtain the data about the confirmed trip from the local trip information data store;
- (c) output data flows to other processes requesting confirmation of various services and if required, advanced payment of tolls, parking lot charges, transit fares, and other costs;
- (d) when all reservation and payment confirmations have been received, confirm the traveler's trip

by sending the traveler trip confirmation information data flow identified above to the traveler interface process;
(e) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the two data stores identified above.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 311 of 475
Page 312 of 475

6.1.3 Manage Intermodal Service Supplier Interface

Input Flows

fitsp_air_services
fitsp_ferry_services
fitsp_intermodal_service_confirmation
fitsp_rail_services
intermodal_service_confirm
intermodal_service_data_read
intermodal_service_data_request
intermodal_service_data_write

Output Flows

intermodal_service_confirmation
intermodal_service_data_response
titsp_air_services_request
titsp_confirm_intermodal_service
titsp_ferry_services_request
titsp_rail_services_request

Description:

Overview: This process shall be responsible for collecting data about services that are available to travelers from intermodal 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 the data in such a way that it can be used as part of a traveler's multimodal trip, and shall also support the subsequent confirmation of the portion of the trip provided by the intermodal service provider.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'intermodal_service_confirm';
- (b)'intermodal_service_data_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to the intermodal transportation service provider:

- (a)'fitsp-air_services';
- (b)'fitsp-ferry_services';
- (c)'fitsp-intermodal_service_confirmation';
- (d)'fitsp-rail_services'.

Solicited Input/Output Processing: This process shall send and receive the following data flows as a means of reading data from and writing data to a data store:

- (a)'intermodal_service_data_read';
- (b)'intermodal_service_data_write'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'intermodal_service_confirmation';
- (b)'intermodal_service_data_response';
- (c)'titsp-confirm_alternate_mode_service';
- (d)'titsp-ferry_services_request';
- (e)'titsp-rail_services_request';
- (f)'titsp-air_services_request';
- (g)'intermodal_services_details'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;

- (b) when the first input is received, request confirmation for the services that were specified;
- (c) when a response to (b) is received, generate the first output flow identified above with the result of the confirmation request;
- (d) when the second input is received, read data from the store of intermodal transportation service details to see if the required data has already been provided;
- (e) if the data in (d) does not contain the required services or is more than a month old, generate the fourth to sixth solicited output data flows shown above that correspond to the modes in the input, including the required destination and arrival times;
- (f) when the corresponding solicited inputs (first, second and fourth in the list shown above) are received, load the data into the data store;
- (g) generate last solicited output data flow shown above using either the data read from the store

User Service Requirements:

or that provided by the intermodal transportation

service provider.

PROCESS SPECIFICATIONS

Page 313 of 475

PROCESS SPECIFICATIONS

Page 314 of 475

6.1.4 Provide ISP Operator Interface for Trip Planning Parameters

Input Flows

fispo_trip_planning_parameters_request
fispo_trip_planning_parameters_update
trip_planning_parameters

Output Flows

tispo_trip_planning_parameters

Description:

Overview: This process shall be responsible for the management of the store containing parameters used by the trip planning processes. These parameters shall govern the way in which intermodal trips are planned by these processes. The process shall support inputs from the ISP operator in both manual and audio form, and shall provide its outputs in audible and visual forms. It shall enable the visual output to either be in hardcopy, or as a display.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fispo-trip_planning_parameters_request'.
- (b)'fispo-trip_planning_parameters_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:

- (a)'trip_planning_parameters'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'tispo-trip_planning_parameters';
- (b)'trip_planning_parameters'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the first input is received, read the data from the store and generate the second output identified above;
- (c) when the second input is received, update the contents of the store with the new data;
- (d) be responsible for the management of the data in the store of trip planning parameters data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 315 of 475

6.2.1.1 Collect Traffic Data for Advisory Messages

Input Flows

predicted_incidents
prediction_data
traffic_data_for_advisory_output
traveler_traffic_information_data_output

Output Flows

traffic_data_advisory_request
traveler_traffic_information_data_update

Description:

Overview: This process shall be responsible for the collection and fusion of data received from the Manage Traffic function and loading it into the store of traveler traffic information. The data shall be provided to the process as both requested and unrequested data flows. The requested data flows shall be obtained by the process at periodic intervals. The process shall fuse all the received data into a coherent set which is loaded into a data store for access by other processes in the Provide Driver and Traveler Services function.

Data Flows: All input data flows are solicited with the exception of those listed above that contain prediction and predicted incidents data. All output flows are solicited. Read and write access to the local store into which the input data is loaded after fusion is provided through the 'traveler_traffic_information_data' data flow.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above containing prediction data and predicted incidents data;
- (b) periodically (at intervals of not less than once a minute between 7am and 7pm and not less than once every five (5) minutes at all other times), generate the data output flows to processes in this and other functions, as listed above;
- (c) collect the data returned as a result of (b) and load it with that received in (a) into the data store of traveler traffic information, fusing it with the data already present, deleting old data, e.g. that relating to incidents that are completed, etc.;
- (d) be responsible for the management of the data in the store of traveler information data, using the most appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 316 of 475

PROCESS SPECIFICATIONS

Page 317 of 475

6.2.1.2 Provide Traffic and Transit Advisory Messages

Input Flows

advisory_data_request
traveler_traffic_information_data
traveler_transit_information_data

Output Flows

advisory_data

Description:

Overview: This process shall be responsible for extracting data from the stores of traveler traffic and 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.

Data Flows: The input data flow requesting advisory information is unsolicited. All output flows and the other input flows are solicited with the exception of the following:

- (a) 'traveler_traffic_information_data' data flow - which contains data requested from a store;
- (b) 'traveler_transit_information_data' data flow - which contains data requested from a store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow requesting advisory data listed above;
- (b) when the input in (a) is received, read the requested data from the store identified above and generate the advisory data output flow identified above;
- (c) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 318 of 475

PROCESS SPECIFICATIONS

Page 319 of 475

6.2.1.3 Collect Transit Data for Advisory Messages

Input Flows

transit_running_data_for_advisory_output
transit_services_for_advisory_data
traveler_transit_information_data_output

Output Flows

transit_conditions_advisories_request

transit_services_advisories_request
traveler_transit_information_data_update

Description:

Overview: This process shall be responsible for the collection and fusion of data received from the Manage Transit function and loading it into the store of traveler transit information. The data shall be obtained by request from the process at periodic intervals. The process shall fuse all the received data into a coherent set which is loaded into a data store for access by other processes in the Provide Driver and Traveler Services function.

Data Flows: All input data flows are solicited as are all output flows. Read and write access to the local store into which the input data is loaded after fusion is provided through the 'traveler_transit_information_data' data flow.

Functional Requirements: This process shall meet the following functional requirements:

- (a) periodically (at intervals of not less than once a minute between 7am and 7pm and not less than once every five (5) minutes at all other times), generate the data output flows to processes in this and other functions, as listed above;
- (b) collect the data returned as a result of (a) and load it into the local data store, fusing it with the data already present, deleting old data, e.g. that relating to incidents that are completed, etc.;
- (c) be responsible for the management of the data in the store of traveler information data, using the most appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 320 of 475

6.2.1.4 Provide Traffic and Transit Broadcast Messages

Input Flows

broadcast_parameters_data
traveler_traffic_information_data
traveler_transit_information_data

Output Flows

broadcast_data

Description:

Overview: This process shall be responsible for extracting data from stores of traveler traffic and transit information at regular intervals and sending it out to the driver or transit user in a vehicle as a wide area broadcast message. This message shall be output by the process at regular intervals without the need for requests by either of the recipients or the ISP operator. The data read from the store shall be filtered using parameters set up in a store by the ISP operator.

Data Flows: The input data flow requesting advisory information is unsolicited. All output flows and the other input flows are solicited with the exception of the following:

- (a) 'traveler_traffic_information_data' data flow - which contains data requested from a store;
- (b) 'traveler_transit_information_data' data flow - which contains data requested from a store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) periodically read the data from the stores identified above and generate the broadcast data output flow identified above;
- (b) the data flow in (a) shall be generated using the filter parameters set up by the ISP operator and retained in a local data store;
- (c) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 321 of 475

6.2.1.5 Provide ISP Operator Broadcast Parameters Interface

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

Description:

Overview: This process shall be responsible for providing the interface through which the ISP operator can manipulate data in the store of broadcast parameters. The data in this store shall be used by the process that outputs wide area broadcast messages to vehicles. The process shall enable the ISP operator to request that the parameter data is output and/or update the store with new parameter values. The process shall support inputs from the ISP 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fispo-broadcast_data_parameters_request';
- (b)'fispo-broadcast_data_parameters_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores:

- (a)'broadcast_parameters_data_output'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'broadcast_parameters_data_update';
- (b)'tispo-broadcast_data_parameters_output'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the first input is received, read the data from the store of parameters using the solicited input flow identified above and generate the second solicited output flow identified above;
- (c) when the second input is received, update the data in the store of parameters using the first solicited output flow shown above;
- (d) be responsible for the management of the data in the store of route selection parameters, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 322 of 475

6.2.2 Prepare and Output In-vehicle Displays

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_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
yellow_pages_advisory_requests

Description:

Overview: This process shall be responsible for producing in-vehicle advisory and broadcast data displays for drivers and transit users. The displays shall be produced by the process either in response to requests for information from the driver, or the transit user in a vehicle, or be sent by processes in other ITS functions as unsolicited data for automatic output to the driver. The process shall obtain the data for requested output from other processes and shall include the vehicle's current location in the data request. The request shall be repeated by the process when

the vehicle changes location by a significant distance, e.g. one quarter of a mile. Prioritizing of the data sent for output to the driver shall be set up by the process so that safety and warning messages supersede advisory and broadcast messages, and both are output in conjunction with the visual data. The process shall also support the transfer of reservation requests from the transit user for other (yellow pages) services to other processes for action.

Data Flows: All input data flows are unsolicited with the exception of that requesting input of advisory data. All output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs other than those from the driver, transit user, or location data are received, generate the output of advisory or broadcast data to the driver identified above;
- (c) when the input from either the driver or the transit user is received, add the current vehicle location and send it as the advisory data request to another process;
- (d) the data returned as a result of (c) shall be output to the driver or transit user depending on the source of the request for advisory information;
- (e) when the broadcast data input is received, it shall be filtered to remove that which is not relevant to the wide area surrounding the vehicle before being output to the driver;
- (f) repeat (c) and (d) for as long as the input is present, every time the vehicle location changes by more than one quarter of a mile.

PROCESS SPECIFICATIONS

Page 323 of 475

PROCESS SPECIFICATIONS

Page 324 of 475

6.2.3 Provide Transit User Advisory Interface

Input Flows

ftu_request_advisory_information
transit_user_advisory_information
vehicle_display_definitions_data

Output Flows

transit_user_advisory_information_request
ttu_advisory_information

Description:

Overview: This process shall be responsible for providing 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. The process shall support inputs from the transit user 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 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.

Data Flows: The input data flow from the transit user is unsolicited. All output data flows are solicited as is the input data flow of traveler advisory information. The data flow 'vehicle_display_definitions_data' provides output template data from a local data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above that is a request from the transit user for advisory information;
- (b) when the input in (a) is received, generate the output information request output data flow identified above;
- (c) when the input data flow of advisory information is received in response to (b), generate the output to the transit user using the template data available from the local data store;
- (d) continue with (b) and (c) for as long as the input from the transit user is present;
- (e) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 325 of 475

PROCESS SPECIFICATIONS

Page 326 of 475

6.2.4 Collect Yellow Pages Data

Input Flows

transit_user_yellow_pages_output
yellow_pages_data

Output Flows

transit_user_yellow_pages_update
yellow_pages_data_request

Description:

Overview: This process shall be responsible for the collection and fusion of data about other (yellow pages) services received from other processes in the Provide Driver and Traveler Services function. The data shall be provided as a requested data flow and shall be obtained by the process at periodic intervals. The process shall fuse all the received data into a coherent set which is loaded into a store of yellow pages information for access by another process in response to requests from the transit user.

Data Flows: All input and output data flows are solicited. Read and write access to the local store is provided through the following data flows:

- (a)'transit_user_yellow_pages_output', contains data being read from the store;
- (b)'transit_user_yellow_pages_update', contains data being written to the store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) periodically (at intervals of not less than once a minute between 7am and 7pm and not less than once every five (5) minutes at all other times), generate the data output flow to request other (yellow pages) services data, as listed above;
- (b) collect the data returned as a result of (a) and load it into the local data store, fusing it with the data already present;
- (c) be responsible for the management of the data in the store of traveler information data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 327 of 475

6.2.5 Provide Driver Interface

Input Flows

driver_advisory_information
driver_broadcast_information
fd_activate_vehicle_control
fd_request_advisory_information
vehicle_display_definitions_data

Output Flows

driver_advisory_information_request
td_advisory_information
td_broadcast_information
vehicle_control_request

Description:

Overview: This process shall be responsible for providing an 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 and safety messages to the driver. The output of safety messages and vision enhancement data by the process shall take priority over all other types of output. When constructing all outputs the process shall use the data in the store of vehicle display definitions data. 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 automatic highway system (ahs) lanes. 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 either in hardcopy, or as a display. Both types of output shall be produced in such a way that they do not impair the driver's ability to control the vehicle in a manner that is safe to both its occupants, to other road and freeway users, and to pedestrians.

Data Flows: All input data flows are unsolicited. The output flow is solicited as is the input of driver advisory information when it is received in response to driver input. The data flow 'vehicle_display_definitions_data' provides output template data from a local data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow listed above that is a request from the driver for advisory information;
- (b) when the input in (a) is received, generate the output information request output data flow identified above;
- (c) when the input data flow of advisory information is received in response to (b), generate the output to the driver using the template data available from the local data store;
- (d) continue with (b) and (c) for as long as the input from the driver is present;
- (e) if the request from the driver is for a change in the mode of vehicle automatic control, send the data to another process in the Provide Vehicle Control and Monitoring function using the vehicle control request data flow;
- (f) if the driver advisory information data flow is received as unsolicited input, output the data that it contains immediately;
- (g) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 328 of 475
Page 329 of 475

6.2.6 Provide Yellow Pages Data and Reservations

Input Flows

transit_user_reservation_confirmation
transit_user_yellow_pages_information_data
yellow_pages_advisory_requests

Output Flows

transit_user_reservation_request
yellow_pages_advisory_data

Description:

Overview: This process shall be responsible for extracting data from the store of transit user yellow pages information upon request for advisory data from the driver or a transit user in a vehicle. The data read from the store shall be filtered by the process so that output only contains that which is relevant to the current location of the vehicle from which the request was made. The process shall also enable the transit user to make reservations for other (yellow pages) services from a vehicle.

Data Flows: The input data flow requesting advisory information is unsolicited. All output flows and the other input flows are solicited with the exception of the following:

- (a)'traveler_traffic_information_data'data flow - which contains data requested from a store;
- (b)'traveler_transit_information_data'data flow - which contains data requested from a store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow requesting advisory data listed above;
- (b) when the input in (a) is received, read the requested data from the store identified above and generate the advisory data output flow identified above;
- (c) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 330 of 475
Page 331 of 475

6.3.1 Get Traveler Request

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

Description:

Overview: This process shall be responsible for receiving input data from a traveler located at a kiosk. The process shall provide support for trip planning, traffic, transit and other (yellow pages) services information requests, as well as trip and yellow pages 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 the responses. The interface to the traveler shall also be provided through a separate process, from which the input flow to this process will originate.

Data Flows: The input data flow is unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the traveler trip planning input flow listed above;
- (b) when the flow in (a) is received, extract the data and send it to the appropriate processes in the Provide Driver and Traveler Services function;
- (c) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 332 of 475

PROCESS SPECIFICATIONS

Page 333 of 475

6.3.2 Inform Traveler

Input Flows

map_data_for_traveler_displays
traffic_data_for_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

Description:

Overview: This process shall be responsible for providing the traveler located at a kiosk with data about all requested trip, traffic, transit and other (yellow pages) services information plus 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 to be shown in a suitable format. This process shall obtain data from other ITS functions or be sent it as a result of requests from another process.

Data Flows: All output flows are solicited and all input data flows are unsolicited with the exception of the following:

- (a)'traffic_data_for_kiosks'- which is received as a result of output being sent to another process;
- (b)'transit_deviations_for_kiosks'- which is received as a result of output being sent to another process;
- (c)'transit_services_kiosk_request'- which is received as a result of output being sent to another process;
- (d)'map_data_for_traveler_displays'- which contains data requested from a data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above that are not details of transit services, traffic data and the display map data;
- (b) when any of the flows in (a) are received, retrieve the relevant digitized display map data from the local store and send the combined data to the traveler interface process;
- (c) when the flow received in (a) contains a request for transit or traffic data, send the request

- to the relevant process in the Manage Transit or Manage Traffic function;
- (d) the input data received as a result of (c) shall be combined with the relevant digitized display map data from the local store and sent to the traveler interface process;
- (e) use the most appropriate mechanism(s) such as RDBMS, to read data from the store of digitized map data identified above;
- (f) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 334 of 475

PROCESS SPECIFICATIONS

Page 335 of 475

6.3.3 Provide Traveler Kiosk Interface

Input Flows

emergency_request_kiosk_traveler_acknowledge
 ft_extra_trip_data
 ft_kiosk_emergency_request
 ft_trip_planning_requests
 traveler_regular_data
 traveler_roadside_credit_identity
 traveler_trip_planning_responses

Output Flows

emergency_request_kiosk_traveler_details
 traveler_roadside_trip_costs
 traveler_trip_planning_requests
 tt_extra_trip_data_request
 tt_kiosk_emergency_response
 tt_trip_planning_responses

Description:

Overview: This process shall be responsible for providing an interface at a kiosk through which travelers can plan and confirm trips, as well as obtain current traffic and transit information, and declare emergencies. The process shall support the inclusion in the trip planning and confirmation data of other (yellow pages) services such as lodging, restaurants, theaters, and other tourist activities. 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 data from the traveler 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 the purpose. 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 kiosk. These forms shall also include those that are suitable for travelers with physical disabilities. Output data shall be displayed by the process for a reasonable length of time before giving the traveler the option to renew it, or have it automatically deleted. 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 the data being displayed to only show that which is relevant to the local area in which the kiosk is located.

Data Flows: All input data flows are unsolicited and all output flows are solicited, with the exception of the 'traveler_regular_data' data flow which contains data requested from or written to a data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows from the traveler listed above;
- (b) when any of the inputs in (a) are received, check for content and if necessary utilize data from the local store identified above;
- (c) generate the output identified above and load the requested data into the local data store;
- (d) continually monitor the data in the local store and compare it with that being input by travelers, deleting any data from the store which is not frequently used;
- (e) be responsible for the management of the data in the store of regularly used data, using the appropriate mechanism(s) such as RDBMS, for storing the data;
- (f) when the emergency request input is received from the traveler, generation of the output to the Manage Emergency Services function must take priority over all other processing;
- (g) following the output of the message in (f), all other processing shall be suspended until the

acknowledgement data flow is received from the Manage Emergency Services function, and the output has been displayed;

(h) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted using any form of digital or analog techniques.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 336 of 475
Page 337 of 475

6.3.4 Update Traveler Display Map Data at Kiosk

Input Flows

fmup_traveler_display_update

Output Flows

map_data_for_traveler_displays

tmup_request_traveler_display_update

Description:

Overview: This process shall be responsible for providing 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 displays output to travelers at kiosks. The process shall obtain the new data from a specialist data supplier or some other appropriate data source, at periodic intervals.

Unsolicited Output Processing: This process shall provide the following output flows regardless of any inputs being received:

(a)'tmup-request_traveler_display_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to external functions:

(a)'fmup-traveler_display_update'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above solicited input being received:

(a)'map_data_for_traveler_displays'.

Functional Requirements: This process shall meet the following requirements:

(a) send out the request for new data from the specialized digital map data supplier at periodic intervals (e.g. once per month) automatically so as to provide an up to date map display using the unsolicited output data flow shown above;

(b) as a result of the output of the data flow in (a) continuously monitor for receipt of the solicited input data flow shown above;

(c) when the flow in (b) is received, output the second solicited output data flow shown above;

(d) be capable of receiving the input data in a variety of formats and converting it into a single format suitable for use with the store of digitized map data;

(e) be responsible for the management of the data in the store of digitized map data, using the most appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 338 of 475

6.4.1 Screen Rider Requests

Input Flows

rideshare_eligibility_data

traveler_rideshare_request

Output Flows

rideshare_ineligible_status_notification

rideshare_request_from_eligible_traveler

Description:

Overview: This process shall be responsible for accepting and screening traveler requests for ridesharing. 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 data store, to screen travelers before they are matched with other travelers to enable ridesharing for all or part of their proposed trips.

Data Flows: Both output flows are solicited. The input data flow from the traveler is unsolicited, whilst the other input flow'rideshare_eligibility_data'contains data read from the store of rideshare data.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow from the traveler listed above;
- (b) when the input in (a) is received, check the traveler's eligibility for ridesharing against the data in the local rideshare data store;
- (c) if the traveler is eligible for ridesharing, send the eligible ride share request data flow to another process for matching of the traveler's proposed trip with others;
- (d) if the traveler is ineligible for ridesharing, send the ineligible status data flow to the process that reports the results of the rideshare application to the process from which the rideshare request was sent;
- (e) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 339 of 475

6.4.2 Match Rider and Provider

Input Flows

rideshare_confirmation_data
rideshare_data
rideshare_request_from_eligible_traveler
traffic_data_for_ridesharing

Output Flows

rideshare_selection
traffic_data_ridesharing_request

Description:

Overview: This process shall be responsible for matching travelers for ridesharing trips. The process shall attempt to achieve a match by considering the origin and destination of the traveler's proposed trip, any routing constraints and preferences specified by the traveler, the compatibility of this with the rideshares confirmed by other travelers, the requesting traveler's eligibility data, plus traffic data obtained on request from the Manage Traffic function. Special algorithms shall be used by the process to find the rideshare that is best suited to the traveler's requirements without causing significant disbenefit to other travelers who will be part of the same rideshare. The process shall store data about selected rideshares in the store of rideshare data, and shall update the data when confirmation of acceptance of the rideshare is received from another process.

Data Flows: The input data flows are unsolicited and all output flows are solicited with the exception of the following:

- (a)'rideshare_data', which contains data requested from or written to a data store;
- (b)'traffic_data_ridesharing_request', which contains a request for traffic data from the Manage Traffic function;
- (c)'traffic_data_for_ridesharing', which contains the response to the request for traffic data from the Manage Traffic function.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow containing the eligible request data, as identified in the list above;
- (b) when the input in (a) is received, match the traveler's rideshare request by comparing its routing with data on existing rideshares in the local data store, and producing a new route for the requesting traveler and one or more existing rideshare travelers;
- (c) check that the route computed in (b) against traffic data requested from the Manage Traffic function to ensure that there will be no significant penalties from predicted occupancy levels and journey times;
- (d) when all the criteria for sharing a ride have been satisfied and other traveler's have been found whose trips will match all or part of that in the current request, send the trip details to another process for the reporting of the results to the process from which the traveler's request was received;
- (e) when the input is received indicating that the rideshare has been confirmed, update the data relating to the rideshare in the store of rideshare data;
- (f) be responsible for the management of the store of rideshare data, using the most appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 340 of 475

6.4.3 Report Ride Match Results to Requestor

Input Flows

rideshare_ineligible_status_notification
rideshare_selection
traveler_rideshare_request

Output Flows

rideshare_response

Description:

Overview: This process shall be responsible for reporting ridesharing 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'rideshare_ineligible_status_notification';
- (b)'rideshare_selection';
- (c)'traveler_rideshare_request'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'rideshare_response'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the input is received from the screening process indicating a failure, output this in the rideshare response data flow;
- (c) when the rideshare details input is received, check that the rideshare matches the traveler's rideshare request;
- (d) if the test in (c) is true, generate the rideshare response with the details of the rideshare;
- (e) if the test in (c) fails, generate the rideshare response with a failure indicated.

PROCESS SPECIFICATIONS

Page 341 of 475

6.4.4 Confirm Traveler Rideshare Request

Input Flows

rideshare_payment_confirmation
traveler_rideshare_confirmation

Output Flows

rideshare_confirmation
rideshare_confirmation_data
rideshare_payment_request

Description:

Overview: This process shall be responsible for confirming the traveler's rideshare request and initiating the necessary payment transaction. 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 rideshare confirmation. A successful payment transaction shall result in this process loading confirmation of the rideshare into the store of rideshare data.

Data Flows: The input data flow of traveler confirmation is unsolicited. All other input and output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow confirming the traveler's rideshare request, as shown in the list above;
- (b) when the input in (a) is received, generate the output to request payment for obtaining the rideshare, identified in the above list;
- (c) when as a result of (b) the payment confirmation data flow is received, generate the rideshare

confirmation data flow to the process that provided the rideshare confirmation and to the process that matches rider and provider as it also manages the store of rideshare data;
(d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 342 of 475

6.5.1 Collect and Update Traveler Information

Input Flows

fm_traveler_information
fws_current_weather
fws_predicted_weather
fypsp_yellow_pages_data
incident_information
tourist_information
yellow_pages_new_data_request
yellow_pages_service_provider_data
yellow_pages_update_request

Output Flows

current_conditions
incident_information_request
tm_traveler_information_request
typsp_yellow_pages_info_request

Description:

Overview: This process shall be responsible for the collection and updating of data about incidents, road construction, weather, events and other services (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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fws-current_weather';
- (b)'fws-predicted_weather';
- (c)'yellow_pages_new_data_request';
- (d)'yellow_pages_update_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:

- (a)'incident_information';
- (b)'yellow_pages_service_provider_data', which contains data requested from a data store;
- (c)'fm-traveler_information';
- (d)'fypsp-yellow_pages_data';
- (e)'tourist_information', which contains data requested from a data store.

Unsolicited Output Processing: This process shall provide the following output flows regardless of any inputs that are received:

- (a)'incident_information_request'
- (b)'tm-traveler_information_request';
- (c)'typsp-yellow_pages_info_request'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'tourist_information', which contains data written to a data store;
- (b)'typsp-yellow_pages_info_request'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when either of the weather service inputs are received, load the data into the store of tourist information using the solicited output flow shown above;
- (c) when either of the yellow pages data flows are received in (a) send the yellow pages data request shown above in the list of unsolicited output flows;
- (d) when the response to (c) is received in the solicited yellow pages input flow, load the data into

the store of tourist information using the solicited output flow shown above;

- (e) before loading data into the store of tourist information, read the current data from the store and amalgamate it with the new data;
- (f) be responsible for the management of the data in the store of tourist information, using the most appropriate mechanism(s) such as RDBMS, for storing the data;
- (g) use the most appropriate mechanism(s) such as RDBMS, to read data from the store of information and service provider data identified above.

PROCESS SPECIFICATIONS

Page 343 of 475

PROCESS SPECIFICATIONS

Page 344 of 475

6.5.2 Provide Traveler Yellow Pages Information and Reservations

Input Flows

fypsp_transaction_confirmation
transit_user_reservation_request
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
yellow_pages_data_request

Output Flows

transit_user_reservation_confirmation
traveler_other_services_payment_request
traveler_personal_yellow_pages_data
traveler_yellow_pages_data
typsp_transaction_request
yellow_pages_data
yellow_pages_update_request

Description:

Overview: This process shall be responsible for the provision of information and reservations for 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. If the process cannot find the required data in the store of tourist information, it shall output a request for new data to the process responsible for providing the data in the 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'transit_user_reservation_request';
- (b)'traveler_payment_information';
- (c)'traveler_transaction_request';
- (d)'traveler_personal_payment_information';
- (e)'traveler_personal_transaction_request';
- (f)'traveler_personal_yellow_pages_information_request';
- (g)'traveler_yellow_pages_information_request';
- (h)'yellow_pages_data_request';

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:

- (a)'fypsp-transaction_confirmation';
- (b)'traveler_other_services_payment_result';
- (c)'yellow_pages_data', which contains data requested from a data store.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'transit_user_reservation_confirmation';
- (b)'traveler_other_services_payment_request';

(c)'traveler_personal_yellow_pages_data';
(d)'traveler_yellow_pages_data';
(e)'typsp-transaction_request';
(f)'yellow_pages_update_request';
(g)'yellow_pages_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the inputs are received, generate the solicited outputs as described below;
- (c) unsolicited inputs (a) and (b) together generate solicited outputs (a) and (e), which in turn generate solicited inputs (b) and (a), which then generate solicited output (b);
- (d) unsolicited inputs (c) and (e) will each generate solicited outputs (a) and (f), which in turn

User Service Requirements:

generate solicited inputs (b) and (a), which then generate solicited output (b);
(e) unsolicited inputs (f) and (g) generate solicited input (c), then the solicited outputs (d)
USR = 1.0; and (e) respectively;
USR = 1.5; (f) unsolicited input (h) generates solicited input (c), then the solicited output (g);
USR = 1.5.0; (g) if in (e) or (f) the required data is not in the solicited input, then generate solicited output
USR = 1.5.1; (f) and repeat the read of the input data;
USR = 1.5.1.3; (h) all input and output flows must be encrypted in such a way that it is not possible to determine
USR = 1.5.1.5; the payment information being transmitted, using any form of digital or analog techniques;
USR = 1.5.2; (i) use the most appropriate mechanism(s) such as RDBMS, to read data from the tourist information
USR = 1.5.2.3; store identified above.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 345 of 475
Page 346 of 475

6.5.3 Register Yellow Pages Service Providers

Input Flows

fypsp_request_provider_registration
yellow_pages_service_provider_registration_response

Output Flows

typsp_provider_registration_confirm
yellow_pages_new_data_request
yellow_pages_service_provider_data
yellow_pages_service_provider_registration_request

Description:

Overview: This process shall be responsible for registering 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'fypsp-request_provider_registration'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:
(a)'yellow_pages_service_provider_registration_response'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'yellow_pages_service_provider_data', which is data output to a store;
(b)'yellow_pages_service_provider_registration_request';
(c)'typsp-provider_registration_confirm';

(d)'yellow_pages_new_data_request'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) when the input in (a) is received, generate the solicited output in (b) identified above;
- (c) when as a result of (b), the solicited input flow is received, generate the solicited outputs identified above (a), (c) and (d);
- (d) use the appropriate mechanism(s) such as RDBMS, to write data to the store identified above;
- (e) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 347 of 475

6.6.1 Provide Mutli-modal Route Selection

Input Flows

cf_route_request
cv_route_request
emergency_vehicle_route_request
From_Weather_Service
other_route
paratransit_route_response
predicted_incidents
transit_route
traveler_route_accepted
traveler_route_request
trip_request
vehicle_route

Output Flows

cf_route
cv_route
emergency_vehicle_route
get_other_route
get_transit_route
get_vehicle_route
paratransit_route_confirm
paratransit_route_request
supplied_route
traveler_guidance_route

Description:

Overview: This process shall be responsible for managing the provision of route guidance and route selection activities. It shall support on-line route guidance for travelers using personal portable devices, the selection of specialized vehicle based routes for other ITS functions, such as Manage Emergency Services and Manage Commercial Vehicles, and the 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, particularly with regard to security for single travelers and the access needs of those with disabilities, and shall minimize waiting time at modal interchange points.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'cv_route_request';
- (b)'emergency_vehicle_route_request';
- (c)'From_Weather_Service';
- (d)'predicted_incidents';
- (e)'traveler_route_request';
- (f)'trip_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes:

- (a)'other_route';
- (b)'paratransit_route_response';
- (c)'transit_route';
- (d)'traveler_route_accepted';

(d)'vehicle_route'.

Solicited Output Processing: This process shall generate the following output flows as a result of the above inputs being received:

- (a)'cf_route';
- (b)'cv_route';
- (c)'emergency_vehicle_route';
- (d)'get_other_route';
- (e)'get_transit_route';
- (f)'get_vehicle_route';
- (g)'paratransit_route_confirm';
- (h)'paratransit_route_request';

User Service Requirements:

(j)'traveler_route'.

USR = 1.0;

USR = 1.3.0;

requirements:

USR = 1.3.1;

USR = 1.3.1.2;

and shall take

USR = 1.3.2;

USR = 1.3.2.1;

to the

USR = 1.3.2.2;

(type of vehicle,

USR = 1.3.2.2.2;

specified in the input

USR = 1.3.3;

USR = 1.3.3.1;

and the

USR = 1.3.4;

Manage Traffic

USR = 1.3.4.2;

the emergency

USR = 1.3.4.2.1;

USR = 1.3.4.3;

capable of

USR = 1.4; automatically making several route requests of the specialized route selection facilities

until it

USR = 1.4.0;

constraints included

USR = 1.4.3;

USR = 1.4.3.3;

private car in so far

USR = 5.0; as this is allowed by the preferences and constraints in the trip or traveler guidance

request;

USR = 5.2; (g) the requirements of (e) and (f) above shall be superseded by the need to keep waiting

time at

USR = 5.2.2;

or other situations

USR = 5.2.2.1;

where personal security may be a problem;
(h) if the process finds that a route cannot be provided within the preferences and constraints specified in the input request, the process shall produce a response which shall contain details of where the route cannot be selected.

PROCESS SPECIFICATIONS

PROCESS SPECIFICATIONS

PROCESS SPECIFICATIONS

6.6.2.1 Calculate Vehicle Route

Input Flows

fws_current_weather

(i)'supplied_route';

Functional Requirements: This process shall meet the following functional

(a) continuously monitor for receipt of the unsolicited input flows listed above;

(b) priority shall be given to the response to requests for emergency vehicle routes

precedence over all other activities;

(c) the response to any input that is a route request shall be to pass on the request

appropriate Route Selection facility, bearing in mind the originator of the route

or traveler, trip plan or route request) plus the preferences and constraints

data;

(d) when an emergency vehicle route has been determined, data showing the links,

intersections along the route, including expected arrival times shall be sent to the

function to enable the current traffic management strategy to be modified to give

vehicles a green wave;

(e) when trip or traveler guidance requests are received, the process shall be

has determined the best multimodal route, bearing in mind the preferences and

in the original request;

(f) the use of multimodal routes shall be aimed at minimizing the use of the

Page 348 of 475

Page 349 of 475

Page 350 of 475

fws_predicted_weather
get_vehicle_route
map_data_for_route_selection
route_segment_details
route_segment_details_updated
route_selection_parameters
routes_for_vehicles_data
vehicle_guidance_route_accepted
vehicle_route_request

Output Flows

logged_hazmat_route
low_traffic_route
request_route_segment_data
vehicle_guidance_route
vehicle_route

Description:

Overview: This process shall be responsible for calculating the routes for all types of vehicles both for trip planning and in real-time, for dynamic guidance. The route data provided by the process in response to requests from vehicles using infrastructure based in-vehicle guidance shall only contain the data necessary for the vehicle to provide guidance. This shall be different to the data that the process provides for trip planning purposes. The process shall select the route according to the data included in the route request, using the preferences and constraints in the request, and other factors such as the predicted weather. If the process cannot find the data it needs in the store of route segment data, it shall request the process responsible for providing route calculation data to obtain it from the appropriate source. When a route is to be used in low traffic conditions, the process shall send the route data to the Manage Traffic function so that a 'green wave' can be provided for the requesting vehicle. Details of routes for commercial vehicles with hazardous or unusual loads shall be sent to the Manage Incidents facility in the Manage Traffic function for processing as a predicted incident.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'get_vehicle_route';
- (b)'vehicle_route_request';
- (c)'fws-current_weather';
- (d)'fws-predicted_weather'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:

- (a)'other_road_details';
- (b)'road_details';
- (c)'road_details_updated';
- (d)'routes_for_vehicles_data';
- (e)'route_selection_parameters';
- (f)'vehicle_guidance_route_accepted'.

Solicited Output Processing: This process shall provide the following output flows as a result of the processing required by receipt of the above inputs:

- (a)'logged_hazmat_route';
- (b)'low_traffic_route';
- (c)'request_route_segment_data';
- (d)'routes_for_vehicles_data';
- (e)'vehicle_guidance_route';
- (f)'vehicle_route'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) the data received from the Weather Service shall be loaded into a local data store for use during the route determination process;
- (c) the route selection process shall be performed using all the data provided in the request, and take account of the data received from the Weather Service and that in the store of road details;

(d) the route selection process shall only be performed when a request is received, or when the road details have been updated - see unsolicited input flows above;

User Service Requirements: (e) if during the course of the route selection process it is found that data for certain links does not exist in the local store, then the process shall check the store of other road details, and if

USR = 1.3.0; the data is still not found, set up a request to the Provide Vehicle Route Calculation Data process

USR = 1.3.1; for that data to be obtained;

USR = 1.3.1.2; (f) when the route has been selected, the time at which vehicles will use each route segment shall

USR = 1.3.1.2.1; be entered into the store of routes for vehicles data;

USR = 1.3.1.3; (g) a selected route shall not include segments for which the maximum number of allowed vehicles has

USR = 1.3.2; been exceeded, as specified in the store of road details - note that this is particularly important

USR = 1.3.2.2; for segments that are part of an Automatic Highway System (AHS);

USR = 1.3.2.2.1; (h) if the number of vehicles on route segments within a selected route fall below preset values in

USR = 1.3.3; the store of road details, that on completion of the route selection, send details of the links

USR = 1.3.3.1; involved and the times at which intersections will be reached to the Manage Traffic function so that

USR = 1.3.3.2; a form of 'green wave' can be produced, this minimizing unwanted stops and delays to vehicles

USR = 1.3.3.2.1; traveling at off-peak times;

USR = 1.3.3.2.2; (i) the process shall be responsible for the maintenance of the store of routes for vehicles data,

USR = 1.3.3.3; using the appropriate mechanism(s) such as RDBMS, for storing the data, and shall delete data

USR = 1.3.4; relating to vehicles that have now used the segments on their routes from that store;

USR = 1.3.4.2; (j) if the route for a commercial vehicle is constrained by its load, either because of size or its

USR = 1.3.4.3; contents, e.g. HAZMAT loads, send a list of the links and arrival times at intersections to the

USR = 1.3.4.3.1; Manage Traffic function (Manage Incidents facility);

(k) the decision on whether a selected route shall contain guidance information shall be based on the source of the request, and the time at which the route is to be used, i.e. those that are not to be used in the immediate future shall not contain guidance data.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 351 of 475
Page 352 of 475

6.6.2.2 Provide Vehicle Route Calculation Data

Input Flows

current_highway_network_state
current_road_network_state
link_data_for_guidance
other_route_segment_data
predicted_incidents
prediction_data
request_route_segment_data
route_segment_details
routes_for_vehicles_data
traffic_data_for_guidance
vehicle_probe_data_amalgamation

Output Flows

current_road_network_use
link_and_queue_data
link_data_store
request_other_route_segment_data

route_segment_details_updated
traffic_data_guidance_request

Description:

Overview: This process shall be responsible for updating the data stores containing information which is used by the another process to calculate vehicle routes. This process shall also provide data about link journey times and queues to vehicles providing autonomous guidance. The data shall be obtained by this process either from the that provided by other guided vehicles, or through data collected from vehicles equipped for electronic toll collection. The process shall obtain data for route segments either as requested data or as data received at periodic intervals from other ITS functions. The process shall respond to requests for data about route segments outside its own area by sending a data request to the process that provides the interface with the other ISP's.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:

- (a)'current_highway_network_state';
- (b)'current_road_network_state';
- (c)'link_data_for_guidance';
- (d)'predicted_incidents';
- (e)'prediction_data';
- (f)'request_route_segment_data';
- (g)'vehicle_probe_data_amalgamation'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:

- (a)'link_data_store';
- (b)'map_database';
- (c)'other_route_segment_data';
- (d)'route_segment_details';
- (e)'routes_for_vehicles_data';
- (f)'traffic_data_for_guidance'.

Unsolicited Output Processing: This process shall provide the following output flows regardless of any inputs that are received:

- (a)'current_road_network_use';
- (b)'link_and_queue_data';
- (c)'traffic_data_guidance_request'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'link_data_store';
- (b)'route_segment_details';
- (c)'request_other_rooute_segment_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) periodically send the first and third unsolicited output data flows listed above to the Manage Traffic function;

User Service Requirements:

- (c) every five (5) minutes broadcast the second unsolicited output data flow to the Provide Vehicle
- USR = 1.0; Guidance facility, i.e. the frequency will be independent of the number of vehicles;
- USR = 1.3.0; (d) when actual road data is received, load it into the appropriate data store for use by other
- USR = 1.3.1; processes within the Select Vehicle Route facility;
- USR = 1.3.1.2; (e) each update of the data in a store should be followed by the sending of the road details updated
- USR = 1.3.1.2.1; data flow to the process that calculates vehicle routes;
- USR = 1.3.1.3; (f) if a request for data for route segments outside those in the local area is received, the
- USR = 1.3.3; process shall determine the source of supply of that data using the contents of the store of link

USR = 1.3.3.1; data and send a request to the process within the Select Vehicle Route facility that has links to

USR = 1.3.3.2; other data suppliers;

USR = 1.3.3.2.1; (g) the process shall be responsible for the maintenance of the store of road details both for local

USR = 1.3.3.2.2; data and that obtained from other data suppliers, plus the store of data showing which links are not

USR = 1.3.3.3; local and the identity of the supplier holding the data, using the most appropriate mechanism(s)

USR = 1.3.4; such as RDBMS.

USR = 1.3.4.2;

USR = 1.3.4.3;

USR = 1.3.4.3.1;

PROCESS SPECIFICATIONS

Page 353 of 475

PROCESS SPECIFICATIONS

Page 354 of 475

6.6.2.3 Provide Route Segment Data for Other Areas

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

Description:

Overview: This process shall be responsible for providing communications with suppliers of data for road links that are outside the area served by the local supplier. The boundary and scope of the local area shall not affect the operations of the process and may be defined on a geographic or jurisdictional basis. The communications provided by the process shall enable data to be obtained for use in route selection calculations, where the destination or best route is outside the area served by the local supplier. The process shall support data requests from the local processes, or from processes in other areas, which have a route that goes into the local area of this process.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'foisp-request_data';
- (b)'request_other_data'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to a terminator and requests for data retrieval:

- (a)'foisp-data_supply';
- (b)'link_data_store';
- (c)'road_details'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'other_route_segement_data';
- (b)'toisp-data_supply';
- (c)'toisp-request_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) use the most appropriate communications protocols to handle the data traffic on the link to the terminator, bearing in mind the transmission medium plus the size and frequency of the data flow.

PROCESS SPECIFICATIONS

Page 355 of 475

6.6.2.4 Update Vehicle Route Selection Map Data

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

Description:

Overview: This process shall be responsible for providing the interface to a map update provider through which fresh 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 data store by the process in such a way that it can be easily used by the route selection process in determining vehicle routes and for on-line vehicle guidance and trip planning.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'request_route_selection_map_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to the map provider terminator:
(a)'fmup-route_selection_map_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'map_data_for_route_selection';
(b)'tmup-request_route_selection_map_update'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input data flow listed above;
(b) when the input is received, generate the second output data flow identified above;
(c) when the new data is received from the map update provider terminator, update the data store using the first output data flow identified above;
(d) be responsible for the management of the data in the store of map data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 356 of 475

6.6.2.5 Provide ISP Operator Route Parameters Interface

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

route_selection_parameters

Output Flows

request_other_routes_map_update

request_route_selection_map_update

tispo_route_selection_parameters

Description:

Overview: This process shall be responsible for providing the interface through which the ISP operator can manipulate map data used by the Provide Driver and Traveler Services function. The operator shall be able to use the process request that the digitized map data used for the selection of both vehicle based and non-vehicle based routes is updated by obtaining a fresh copy from the supplier, or request output of the parameters used to control the trip planning and route selection processes, or update these parameters with new values. The process shall support inputs from the ISP 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'fispo-request_other_routes_selection_map_data_update';
(b)'fispo-request_route_selection_map_data_update';
(c)'fispo-route_selection_parameters_request';
(d)'fispo-route_selection_parameters_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:
(a)'route_selection_parameters'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'request_route_selection_map_update';
(b)'request_other_routes_map_update';
(c)'tispo_route_selection_parameters'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input data flows listed above;
(b) when either of the first two inputs are received, generate the appropriate of the first two output data flows identified above;
(c) when the third input is received, read the data from the route selection parameters data store and output it in the third output data flow shown above;
(d) when the fourth input is received, load the data into the store of route selection parameters;
(e) be responsible for the management of the data in the store of route selection parameters, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 357 of 475

6.6.2.6 Calculate Vehicle Probe Data for Guidance

Input Flows

vehicle_guidance_probe_data
vehicle_toll_probe_data

Output Flows

vehicle_probe_data_amalgamation

Description:

Overview: This process shall be responsible for calculating route segment journey 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 journey times. The journey times for those route segments for which no data is available shall be set by the process to zero.

Unsolicited Input Processing: This process shall receive the following input unsolicited data flows:
(a)'vehicle_guidance_probe_data';
(b)'vehicle_toll_probe_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'vehicle_probe_data_amalgamation'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the unsolicited input data flows listed above;
(b) when either of the input data flows is received, recalculate the route segment journey time for which the data applies, using an appropriate smoothing factor to remove any sudden fluctuations;
(c) periodically (not less than every five (5) minutes) send the amalgamated route segment journey times calculated in (b) to the process that provides vehicle route calculation data.

PROCESS SPECIFICATIONS

Page 358 of 475

6.6.3 Update Other Routes Selection Map Data

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

Description:

Overview: This process shall be responsible for providing the interface to a map update provider through which fresh 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 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 and for on-line traveler guidance and trip planning.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'request_other_routes_map_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to external functions:
(a)'fmup-other_routes_map_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:
(a)'tmup-request_other_routes_map_update';
(b)'map_data_for_general_use'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for the receipt of the unsolicited data flow shown above;
(b) when the data flow in (a) is received, generate the first solicited output data flow shown above and continuously monitor for receipt of the solicited input data flow shown above;
(c) when the flow in (b) is received, output the second solicited output data flow shown above;
(d) be capable of receiving the input data in a variety of formats and converting it into a single format suitable for use with the store of digitized map data;
(e) be responsible for the management of the data in the store of the pollution data log, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 359 of 475

6.6.4 Select Transit Route

Input Flows

get_transit_route
transit_mode_routes
transit_route_details
transit_running_data_for_guidance
transit_services_for_guidance

Output Flows

current_transit_routes_use
transit_conditions_guidance_request
transit_route
transit_services_guidance_request

Description:

Overview: This process shall be responsible for determining routes that are based on regular transit services. Routes shall be provided by the process for travelers in response to both trip planning, and on-line personal guidance requests. The data provided by the process shall be different for the two types of request since the data for trip planning will not need the detailed guidance data. The process shall base its routes on the current state of the regular transit services using data obtained from processes in the Manage Transit function and loaded into a local data store. It shall also take account of any preferences and constraints, such as those for travelers with special needs, that have been 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 facility in the Manage Transit function to aid its demand management.

Data Flows: The input data flow requesting a route is unsolicited. All other input flows and all output flows are solicited with the exception of the following which contains data requested from and written to data stores:

(a)'transit_mode_routes';
(b)'transit_route_details'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the input flow requesting a transit route shown in the list of input data flows above;

- (b) when the input in (a) is received, select a route that meets the requirements of the data in the constraints and preferences supplied as part of the request, using the data in the store of transit route details;
- (c) as a result of (b) send the route details back to the requesting process in the transit route output data flow identified above and load the route details into the store of transit mode routes;
- (d) periodically obtain new transit services data from the manage Transit function and load it into the store of transit route details;
- (e) when the estimated arrival time of the last segment of each route expires, delete the entire route from the store of transit mode route details;
- (f) periodically send a copy of the store of transit mode route details to the Manage Demand facility of the Manage Traffic function in the current transit routes use data flow;
- (g) be responsible for the management of the data in the stores of transit route details and transit mode route details, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 360 of 475

PROCESS SPECIFICATIONS

Page 361 of 475

6.6.5 Select Other Routes

Input Flows

get_other_route
map_data_for_general_use
other_modes_routes
predicted_incidents

Output Flows

current_other_routes_use
other_route

Description:

Overview: This process shall be responsible for determining routes that are not based on the use of vehicles or regular transit services. Routes shall be provided by the process for travelers in response to both trip planning, and on-line personal guidance requests. The data provided by the process shall be different for the two types of request since the data for trip planning will not need the detailed guidance data. The process shall base its routes on digitized map data obtained by another processes from a map supplier. It shall make use of the alternative modes, such as 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 the current use of routes in on-line guidance shall be provided by the process to the Manage Demand facility in the Manage Transit function to aid its demand management.

Data Flows: The input data flow requesting a route is unsolicited. All other input flows and all output flows are solicited with the exception of the following which contains data requested from and written to data stores:

- (a)'other_modes_routes', read and write;
- (b)'map_data_for_general_use', read only.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flow requesting an other modes route, shown in the list of input data flows above;
- (b) when the input in (a) is received, select a route that meets the requirements specified in the constraints and preferences supplied as part of the request, using the data in store of map data;
- (c) as a result of (b) send the route details back to the requesting process in the other route output data flow identified above and load the route details into the store of other mode routes;
- (d) when the estimated arrival time of the last segment of each route expires, delete the entire route from the store of other mode route details;
- (e) periodically send a copy of the store of other mode route details to the Manage Demand facility of the Manage Traffic function in the current other routes use data flow;
- (f) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the store of digitized map data identified above;
- (g) be responsible for the management of the data in the store of other mode route details, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 362 of 475

PROCESS SPECIFICATIONS

Page 363 of 475

6.7.1.1 Build Driver Personal Security Message

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

Description:

Overview: This process shall be responsible for responding 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'fd-emergency_request';
- (b)'driver_location_for_emergencies'
- (c)'vehicle_status_details_for_driver_security'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'vehicle_identity_store'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'driver_personal_emergency_request'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor arrival of input data flow from the driver;
- (b) when input from the driver is received obtain the current vehicle location and status and send this in a message with the vehicle identity and status to the communications process;
- (c) if some or all of the data in (b) is missing, e.g. there is no current location, the message must be sent anyway, and repeated when a location becomes available.

PROCESS SPECIFICATIONS

Page 364 of 475

6.7.1.2 Provide Driver In-vehicle Communications Function

Input Flows

driver_personal_emergency_request
emergency_request_driver_acknowledge

Output Flows

emergency_message_driver_output
emergency_request_driver_details

Description:

Overview: This process shall be responsible for preparing and sending 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'driver_personal_emergency_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'emergency_request_driver_acknowledge'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'emergency_request_driver_details';
- (b)'emergency_message_driver_output'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) transmission of the output message must be as near to instantaneous as possible following the receipt of data from the driver security message preparation process;
- (b) the current data and time must be added to the data received from the driver security message preparation process;
- (c) when the acknowledgment message is received it should be sent immediately to the interface process for driver advisory data;
- (d) transmission of the output message must be repeated until an acknowledgment message is received updating the date and time as they change;
- (e) initially, the message sent for output to the driver must show that data has been sent, and only be changed when an acknowledgment is received.

PROCESS SPECIFICATIONS

Page 365 of 475

6.7.2.1.1 Determine In-vehicle Guidance Method

Input Flows

autonomous_vehicle_guidance_data
driver_guidance_accepted
driver_guidance_data
driver_guidance_request
dynamic_vehicle_guidance_data
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

Description:

Overview: This process shall be responsible for acting 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 methods shall be available to the process, comprising dynamic using centralized data, autonomous with link queue and journey times obtained from a central source, and autonomous using local data only. When dynamic guidance is selected then the vehicle's journey time for each link shall be provided by the process back to the 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'driver_guidance_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:

- (a)'autonomous_vehicle_guidance_data';
- (b)'driver_guidance_accepted';
- (c)'driver_guidance_data';
- (d)'dynamic_vehicle_guidance_data';
- (e)'retained_vehicle_guidance_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'autonomous_vehicle_guidance_accepted';

- (b)'autonomous_vehicle_guidance_data_request';
- (c)'driving_guidance_instructions';
- (d)'driver_input_request';
- (e)'dynamic_vehicle_guidance_data_request';
- (f)'retained_vehicle_guidance_data';
- (g)'vehicle_guidance_route_accepted'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) select the appropriate process for the guidance data request dependent on whether the data input by the driver specifies the use of dynamic guidance;
- (c) if dynamic guidance is not available and was requested, then the autonomous guidance process must be used;
- (d) if dynamic guidance is used but becomes unavailable then the autonomous guidance process must be used from the point at which the dynamic guidance was lost;
- (e) if dynamic guidance having been lost is regained, the first dynamic guidance request must use the vehicle's current location as the origin for the route request;
- (f) the process shall be responsible for the maintenance of the store of data used in guidance requests using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 366 of 475

PROCESS SPECIFICATIONS

Page 367 of 475

6.7.2.1.2 Provide Dynamic In-vehicle Guidance

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

Description:

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

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'ahs_route_request';
- (b)'dynamic_vehicle_guidance_data_request';
- (c)'vehicle_location_for_dynamic_guidance'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes:

- (a)'vehicle_guidance_route'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'ahs_route';
- (b)'dynamic_vehicle_guidance_data';
- (c)'vehicle_guidance_probe_data';
- (d)'vehicle_route_request'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;

- (b) when the second input in (a) is received send the driver route request data flow;
- (c) if no origin is specified in the guidance data request, then the current vehicle location data obtained from the unsolicited input flow shall be used as the origin of the flow in (b);
- (d) when the solicited input flow is received, load the data into the flow of dynamic driver guidance data and sent it to the process that determines the driver guidance method;
- (e) every time a route request is made compute the time since the last request and send it with the vehicle's current position in the flow of vehicle probe data;
- (f) when the first unsolicited input flow is received, send a request for a route based on using automatic highway system (ahs) lanes instead of the request in (b);
- (g) the route data resulting from (f) should be sent to the process requesting the ahs route and no guidance data shall be output as in (d) above.

PROCESS SPECIFICATIONS

Page 368 of 475

PROCESS SPECIFICATIONS

Page 369 of 475

6.7.2.1.3 Provide Autonomous In-vehicle Guidance

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

Description:

Overview: This process shall be responsible for providing 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 journey 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 journey time data is being used, the process shall provide guidance for the best route for the current traffic conditions, within the preferences and constraints specified by the driver in the guidance request.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'autonomous_vehicle_guidance_data_request';
- (b)'link_and_queue_data';
- (c)'vehicle_location_for_autonomous_guidance'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval and data being sent to another process:

- (a)'vehicle_map_database';
- (b)'autonomous_vehicle_guidance_accepted'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'autonomous_vehicle_guidance_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input data flows listed above;
- (b) when the first input data flow in (a) is received, determine the shortest route using the data in the store of navigable map data;
- (c) when requested by the driver and available, use the link queue and journey time data to modify the route obtained from using the in-vehicle map database data to produce the best route for the current traffic conditions;
- (d) use the vehicle location data in the second input data flow in (a) to determine the origin for the route unless the driver has specified an origin in the guidance request;
- (e) when the route has been determined, read the associated guidance instructions from the store and output them to the process requesting guidance.

PROCESS SPECIFICATIONS

Page 370 of 475

PROCESS SPECIFICATIONS

Page 371 of 475

6.7.2.2 Process Vehicle Location Data

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_transit

Description:

Overview: This process shall be responsible for providing 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 and guidance systems, as well as for any emergency notification systems shall be provided by the process in a manner that is as precise as is practical within cost and technology constraints, while location data intended for transit vehicles and driver advisories may be less precise.

Data Flows: The input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) continuously compute the vehicle's most probable current location using the data in the input flows, refinement and/or filtering algorithms (e.g. dead reckoning, map-following, etc).
- (c) Provide the vehicle location to other processes in the Provide Driver and Traveler Services, Manage Commercial Vehicle, Manage Transit, and Manage Emergency Services functions using output flows as identified above;
- (d) it shall be possible for the process to compute the location from as many sources of data as are simultaneously available to it, and to apply filtering and/or map matching algorithms as may be appropriate to consolidate or to choose among locations calculated from the various sources of data;
- (e) vehicle location determination for transit and driver advisories may be of lesser precision than locations intended for navigation and route guidance processes.

PROCESS SPECIFICATIONS

Page 372 of 475

PROCESS SPECIFICATIONS

Page 373 of 475

6.7.2.3 Provide Driver Guidance Interface

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

Description:

Overview: This process shall be responsible for providing an interface for the vehicle's driver

through which route guidance can be obtained. The three types of route guidance provided by other processes shall be supported by this process. 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 as and when needed. 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 either in hardcopy, or as a display. Both types of output shall be produced in such a way that they do not impair the driver's ability to control the vehicle in a manner that is safe to both its occupants, to other road and freeway users, and to pedestrians.

Data Flows: All input data flows are unsolicited and all output flows are solicited

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows from the driver listed above;
- (b) when any inputs are received, generate the appropriate outputs identified above;
- (c) implementation of guidance will generate a succession of output data which must be passed on to the driver without the need for further input;
- (d) the output in (c) must be retained until the next set of guidance data arrives for output;
- (e) the advanced payment for map data flow is only generated when the driver credit identity data flow contains a stored credit value and not a credit identity;
- (f) all input and output flows must be encrypted in such a way that it is not possible to determine the credit identity or stored credit being transmitted using any form of digital or analogue technique.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 374 of 475
Page 375 of 475

6.7.2.4 Update Vehicle Navigable Map Database

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

Description:

Overview: This process shall be responsible for updating the vehicle's navigable database based on digitized data obtained from a specialized supplier. The update shall be initiated by the driver through another process, and shall not be carried out unless the required financial transaction is first 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 store of map data 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.

Data Flows: The driver update request input data flow is unsolicited. All other input flows and the output flows are solicited with the exception of the following which contains the new navigable map data to be written to its data store: 'vehicle_map_database'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the driver update request input flow listed above;
- (b) when the input in (a) is received, send the data flow to the map update provider requesting the cost of the update;
- (c) when a response to (b) is received and the credit identity not the stored credit was provided as part of the data in (a), generate the payment request;
- (d) when a successful response is received to (c), generate the request to the map supplier for a new navigable map database;

- (e) when a response to (b) is received and the stored credit not the credit identity was provided as part of the data in (a), compare it with the stored credit value and if greater send the update response data flow to the driver interface process showing an unsuccessful update;
- (f) if the check in (e) shows that there is sufficient stored credit, generate the request to the map supplier for a new navigable map database;
- (g) when the response to the map database requests in (d) or (f) is received, load the new navigable map data into the data store, and send the update response data flow to the driver interface process showing a successful update;
- (h) if the map update process fails, send the update response data flow to the driver interface process showing an unsuccessful update;
- (i) use the appropriate mechanism(s) such as RDBMS, to write data to the store identified above;
- (j) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 376 of 475

PROCESS SPECIFICATIONS

Page 377 of 475

6.8.1.1.1 Determine Traveler Guidance Method

Input Flows

autonomous_traveler_guidance_data
dynamic_traveler_guidance_data
retained_traveler_guidance_data
traveler_guidance_accepted
traveler_guidance_data
traveler_guidance_request

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

Description:

Overview: This process shall be responsible for acting 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 using centralized data, and autonomous using local data only. 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:
(a)'traveler_guidance_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval from local data stores:

- (a)'autonomous_traveler_guidance_data';
- (b)'dynamic_traveler_guidance_data';
- (c)'retained_traveler_guidance_data';
- (d)'traveler_guidance_accepted';.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'autonomous_traveler_guidance_accepted';
- (b)'autonomous_traveler_guidance_data_request';
- (c)'dynamic_traveler_guidance_data_request';
- (d)'retained_traveler_guidance_data';
- (e)'traveler_input_request';
- (f)'traveler_guidance_instructions';
- (g)'traveler_route_accepted'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) select the appropriate process for the guidance data request dependent on whether the data input by the traveler specifies the use of dynamic guidance;
- (c) if dynamic guidance is not available and was requested, then the autonomous guidance process must be used;
- (d) if dynamic guidance is used but becomes unavailable then the autonomous guidance process must be used from the point at which the dynamic guidance was lost;
- (e) if dynamic guidance having been lost is regained, the first dynamic guidance request must use the traveler's current location as the origin for the route request;
- (f) the process shall be responsible for the maintenance of the store of data used in guidance requests using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 378 of 475

PROCESS SPECIFICATIONS

Page 379 of 475

6.8.1.1.2 Provide Dynamic Traveler Guidance

Input Flows

dynamic_traveler_guidance_data_request
traveler_guidance_route
traveler_location_for_dynamic_guidance

Output Flows

dynamic_traveler_guidance_data
traveler_route_request

Description:

Overview: This process shall be responsible for providing the interface to processes in other parts of the Provide Driver and Traveler Services function. The interface provided by the process shall enable dynamic traveler guidance data to be obtained. 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'dynamic_traveler_guidance_data_request';
- (b)'traveler_location_for_dynamic_guidance'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes:

- (a)'traveler_guidance_route'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'dynamic_traveler_guidance_data';
- (b)'traveler_route_request'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input data flow listed above;
- (b) when the first input data flow in (a) is received send the traveler route request data flow;
- (c) if no origin is specified in the guidance data request, then the current traveler location data obtained from the unsolicited input flow shall be used as the origin of the flow in (b);
- (d) when the solicited input flow is received, load the data into the flow of dynamic traveler guidance data and sent it to the process that determines the traveler guidance method.

PROCESS SPECIFICATIONS

Page 380 of 475

PROCESS SPECIFICATIONS

Page 381 of 475

6.8.1.1.3 Provide Autonomous Traveler Guidance

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

Description:

Overview: This process shall be responsible for providing 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'autonomous_traveler_guidance_data_request';
- (b)'traveler_location_for_autonomous_guidance'.

Solicited Input Processing: This process shall receive the following data flows as a result of requests for data retrieval from local data stores and data being sent to other processes:

- (a)'autonomous_traveler_guidance_accepted';
- (b)'traveler_map_database'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'autonomous_traveler_guidance_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input data flows listed above;
- (b) when the first input data flow in (a) is received, determine the shortest route using the data in the store of navigable map data;
- (c) use the traveler location data in the second input data flow in (a) to determine the origin for the route unless one is specified in the traveler's request;
- (d) when the route has been determined, read the associated guidance instructions from the store and output them to the process requesting guidance.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 382 of 475
Page 383 of 475

6.8.1.2 Provide Traveler Guidance Interface

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

Description:

Overview: This process shall be responsible for providing an interface for the traveler through which personal guidance can be obtained. The process shall enable the traveler to input data from which it can send to other processes for the determination of a suitable route. The two types of route guidance provided by other processes shall be supported by this process. The process shall also act as the interface through which the output of on-line guidance for the requested route is provided. Multimodal routes shall be supported by the process to give the greatest flexibility of travel for the traveler. 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 as and when needed. The process shall support inputs from the traveler 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. 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows from the traveler listed above;
- (b) when any inputs are received, generate the appropriate outputs identified above;
- (c) implementation of guidance will generate a succession of output data which must be passed on to the traveler without the need for further input;
- (d) the output in (c) must be retained until the next set of guidance data arrives for output;
- (e) the advanced payment for map data flow is only generated when the traveler credit identity data flow contains a stored credit value and not a credit identity;
- (f) all input and output flows must be encrypted in such a way that it is not possible to determine the credit identity or stored credit being transmitted using any form of digital or analogue technique.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 384 of 475
Page 385 of 475

6.8.1.3 Process Traveler Location Data

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

Description:

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

Data Flows: The input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) continuously compute the vehicle's current location using the data in the input flows and send it to other processes in the Provide Driver and Traveler Services and Manage Emergency Services functions using the output flows identified above;
- (c) it shall be possible for the process to compute the location from as many sources of data as are simultaneously available to it, using the following priority order where more than one is available: differential GPS, GPS, map matching, magnetic flux and dead reckoning;
- (d) the location determined by the source with the highest priority shall be used at all times, except that sources determined to be unreliable may be temporarily bypassed (e.g. GPS signals with low signal quality).

PROCESS SPECIFICATIONS

Page 386 of 475

6.8.1.4 Update Traveler Navigable Map Database

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

Description:

Overview: This process shall be responsible for updating the traveler's navigable database based on digitized data obtained from a specialized supplier. The update shall be initiated by the traveler through another process, and will not be carried out unless the required financial transaction is first 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 store of map data 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.

Data Flows: The traveler update request input data flow is unsolicited. All other input flows and the output flows are solicited with the exception of the following which contains the new navigable map data to be written to its data store: 'traveler_map_database'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the traveler update request input flow listed above;
- (b) when the input in (a) is received, send the data flow to the map update provider requesting the cost of the update;
- (c) when a response to (b) is received and the credit identity not the stored credit was provided as part of the data in (a), generate the payment request;
- (d) when a successful response is received to (c), generate the request to the map supplier for a new navigable map database;
- (e) when a response to (b) is received and the stored credit not the credit identity was provided as part of the data in (a), compare it with the stored credit value and if greater send the update response data flow to the traveler interface process showing an unsuccessful update;
- (f) if the check in (e) shows that there is sufficient stored credit, generate the request to the map supplier for a new navigable map database;
- (g) when the response to the map database requests in (d) or (f) is received, load the new navigable map data into the data store, and send the update response data flow to the traveler interface process showing a successful update;
- (h) if the map update process fails, send the update response data flow to the traveler interface process showing an unsuccessful update;
- (i) use the appropriate mechanism(s) such as RDBMS, to write data to the store identified above;
- (j) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 387 of 475

6.8.1.5 Provide Traveler Emergency Message Interface

Input Flows

emergency_message_traveler_output
traveler_location_for_information

Output Flows

tt_emergency_message

Description:

Overview: This process shall be responsible for providing 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.

Data Flows: All input data flows with the exception of that for traveler personal information and traveler location are unsolicited, but all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the input in (a) is received, generate the traveler emergency message output flow identified above.

6.8.2.1 Build Traveler Personal Security Message**Input Flows**

ft_personal_emergency_request
 traveler_identity_store
 traveler_location_for_emergencies

Output Flows

traveler_personal_emergency_request

Description:

Overview: This process shall be responsible for responding 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'ft-emergency_request';
- (b)'traveler_location_for_emergencies'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'traveler_identity_store'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

- (a)'traveler_personal_emergency_request'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor arrival of input from the traveler;
- (b) when input from the traveler is received obtain the current location and traveler identity and send this in a message to the communications process;
- (c) if some or all of the data in (b) is missing, e.g. there is no current location, the message must be sent anyway, and repeated when a location becomes available.

6.8.2.2 Provide Traveler Emergency Communications Function**Input Flows**

emergency_request_personal_traveler_acknowledge
 traveler_personal_emergency_request

Output Flows

emergency_message_traveler_output
 emergency_request_personal_traveler_details

Description:

Overview: This process shall be responsible for preparing and sending 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.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows:

- (a)'traveler_personal_emergency_request'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to other processes and requests for data retrieval:

- (a)'emergency_request_personal_traveler_acknowledge'.

Solicited Output Processing: This process shall provide the following output flows as a result of

the above inputs being received:

- (a)'emergency_request_personal_traveler_details';
- (b)'emergency_message_traveler_output'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) transmission of the output message must be as near to instantaneous as possible following the receipt of data from the security message process;
- (b) the current data and time must be added to the data received from the security message process;
- (c) when the acknowledgment message is received it should be sent immediately to the interface process for the personal traveler guidance facility;
- (d) transmission of the output message must be repeated until an acknowledgment message is received updating the date and time as they change;
- (e) initially, the message sent to the traveler must show that data has been sent, and only changed when an acknowledgment is received.

PROCESS SPECIFICATIONS

Page 390 of 475

6.8.3.1 Get Traveler Personal Request

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

Description:

Overview: This process shall be responsible for receiving traveler requests from a personal portable device. The process shall provide support for trip planning, traffic, transit and other (yellow pages) services information, plus trip and 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 the responses. The interface to the traveler shall also be provided through a separate process, from which the input flow to this process will originate.

Data Flows: The input data flow is unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the traveler trip planning input flow listed above;
- (b) when the flow in (a) is received, extract the data and send it to the appropriate processes in the Provide Driver and Traveler Services function;
- (c) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 391 of 475

PROCESS SPECIFICATIONS

Page 392 of 475

6.8.3.2 Provide Traveler with Personal Travel Information

Input Flows

map_data_for_traveler_personal_displays
traffic_data_for_portables
transit_deviations_for_portables
transit_services_for_portables
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_portables_request
transit_deviations_portables_request
transit_services_portables_request
traveler_personal_trip_planning_responses

Description:

Overview: This process shall be responsible for providing the traveler using a personal portable device with data about all requested trip, traffic, transit and other (yellow pages) services information plus 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 obtain its data from other ITS functions or be sent it as a result of requests from another process.

Data Flows: All output flows are solicited and all input data flows are unsolicited with the exception of the following:

- (a)'traffic_data_for_portables'- which is received as a result of output being sent to another process;
- (b)'transit_deviations_for_portables'- which is received as a result of output being sent to another process;
- (c)'transit_services_portables_request'- which is received as a result of output being sent to another process;
- (d)'map_data_for_traveler_personal_displays'- which contains data requested from a data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above that are not details of transit services, traffic data and the display map data;
- (b) when any of the flows in (a) are received, retrieve the relevant digitized display map data from the local store and send the combined data to the traveler interface process;
- (c) when the flow received in (a) contains a request for transit or traffic data, send the request to the relevant process in the Manage Transit or Manage Traffic function;
- (d) the input data received as a result of (c) shall be combined with the relevant digitized display map data from the local store and sent to the traveler interface process;
- (e) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above;
- (f) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 393 of 475

PROCESS SPECIFICATIONS

Page 394 of 475

6.8.3.3 Provide Traveler Personal Interface

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_regular_data
traveler_personal_trip_planning_responses

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

Description:

Overview: This process shall be responsible for providing an interface in a personal portable device through which travelers can plan and confirm trips, as well as obtain current traffic and transit information. The process shall support the inclusion in the trip planning and confirmation

data of other (yellow pages) services such as lodging, restaurants, theaters, and other tourist activities. The process shall be able to store frequently used data such as traveler identity (the owner of the personal portable device) plus a range of other data such as home and work locations, etc. to 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 before passing the data to other processes. The traveler's payment information and location shall be obtained by this process from other processes specially designed for the purpose. 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 portable device. These forms shall also include those that are suitable for travelers with physical disabilities. Output data shall be displayed for as long as is required by the traveler and the process must enable viewing of previously output data. Where it is appropriate, the process shall use the traveler's current location to filter the data being displayed to that relevant to the local area in which the traveler is currently located.

Data Flows: All input data flows are unsolicited and all output flows are solicited, with the exception of the 'traveler_personal_regular_data' data flow which contains data requested from or written to a data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows from the traveler listed above;
- (b) when any of the inputs in (a) are received, check for content, accuracy, consistency and out of range values, utilizing data from the local store identified above if necessary;
- (c) generate the output identified above and load the requested data into the local data store;
- (d) continually monitor the data in the local store and compare it with that being input by travelers, deleting any data from the store which is not frequently used;
- (e) be responsible for the management of the data in the store of regularly used data, using the appropriate mechanism(s) such as RDBMS, for storing the data;
- (f) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 395 of 475

PROCESS SPECIFICATIONS

Page 396 of 475

6.8.3.4 Update Traveler Personal Display Map Data

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

Description:

Overview: This process shall be responsible for providing updates to the digitized map data used as the background for displays on travelers' personal portable devices. These displays will include details of traffic, trip and travel information for use by travelers. The process shall obtain the new map data from a specialist data supplier or some other appropriate data source on request from the traveler via the traveler interface process. The data will be designed for use with the types of display that will be found on personal portable devices.

Unsolicited Input Processing: This process shall receive the following unsolicited input data flows: (a)'request_traveler_personal_display_map_update'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to an external function (terminator):

- (a)'fmup-traveler_display_update';
- (b)'fmup-traveler_personal_display_update_cost'.

Solicited Input Processing: This process shall receive the following data flows as a result of output being sent to another process:

(a)'traveler_personal_display_update_payment_response'.

Solicited Input Processing: This process shall receive the following data flows as a result of a request for data from a store:

(a)'traveler_personal_regular_data'.

Solicited Output Processing: This process shall provide the following output flows as a result of the above inputs being received:

(a)'map_data_for_traveler_personal_displays';
(b)'tmup-request_traveler_personal_display_update';
(c)'tmup-request_traveler_personal_display_update_cost';
(d)'traveler_personal_display_update_payment_request';
(e)'traveler_personal_display_map_update_response'.

Functional Requirements: This process shall meet the following requirements:

(a) monitor for the receipt of the unsolicited input data flow shown above;
(b) when the flow in (a) is received, send out the request for new data from the specialized digital map data supplier;
(c) be capable of receiving the input data in a variety of formats and converting it into a single format suitable for use with the store of digitized map data;
(d) use the appropriate mechanism(s) such as RDBMS to retrieve data from the store identified above;
(e) be responsible for the management of the data in the store of digitized map data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 397 of 475
Page 398 of 475

7.1.1.1 Read Tag Data for Tolls

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_pull_in_message
vehicle_tag_for_tolls
vehicle_type_for_tolls

Description:

Overview: This process shall be responsible for requesting the data in the toll tag being carried on-board the vehicle and used as the payment instrument be 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 a closed toll system is being used, correctly read data shall be checked by the process to see if it contains a toll segment number and if not present, that for the point at which the tag is currently being read (the toll plaza at which the process is located) shall be re-written back to the tag so that it can be used as the mechanism for charging for the use of the toll road. 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 send it to another process.

Data flows: All input and output data flows are solicited with the exception of the following which is used to trigger the process:

(a)'vehicle_toll_characteristic_data'- which is received from another process that detects a vehicle's presence.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the input flows listed above;
(b) when the unsolicited input identified above is received, generate the outputs identified above;
(c) all input and output flows must be encrypted in such a way that it is not possible to determine

the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 399 of 475

7.1.1.10 Determine Advanced Toll Bill

Input Flows

advanced_toll_needed

toll_prices

Output Flows

advanced_toll_billing

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

(a)'toll_prices'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques;
- (d) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 400 of 475

7.1.1.2 Calculate Vehicle Toll

Input Flows

toll_prices

vehicle_tag_for_tolls

vehicle_type_for_tolls

Output Flows

toll_charge

Description:

Overview: This process shall be responsible for calculating the toll 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 toll segment(s) traveled by the vehicle by reading data from a store that contains the standard prices for toll segments.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

(a)'toll_prices'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 401 of 475

7.1.1.3 Manage Bad Toll Payment Data

Input Flows

bad_toll_payment_list

ffi_bad_toll_payment_updates

toll_bad_payment_check_request

toll_payment_violator_data

Output Flows

tftoll_payment_violator_data
toll_bad_payment_check_response

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 and commercial vehicle carrier numbers provided for checking by the billing process. This checking 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 possible invalid credit identities or carrier numbers 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data written to or requested from a data store:

(a)'bad_toll_payment_list'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) be responsible for the management of the data in the store of bad driver Payment Instrument data, using the appropriate mechanism(s) such as RDBMS, for storing the data.
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 402 of 475

7.1.1.4 Check for Advanced Tolls Payment

Input Flows

advanced_toll_billing
advanced_toll_payment_list
toll_charge

Output Flows

advanced_toll_transactions
billing_for_tolls_needed

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 advance 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

(a)'advanced_toll_payment_list'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) be responsible for the management of the data in the store of advanced tolls payment data, using the most appropriate mechanism(s) such as RDBMS, for storing the data;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques;
- (e) remove the payment information received from the driver from all data that is sent to another process for loading into the store of toll payment transactions.

PROCESS SPECIFICATIONS

Page 403 of 475

7.1.1.5 Bill Driver for Tolls

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

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 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 last of bad payers, or the stored credit is not less than 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 the 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

- (a)'toll_bad_payment_check_response', which contains data from another process;
- (b)'advanced_toll_payment_list', which contains data written to a data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) when the unsolicited flow requesting billing for tolls is received, check to see if it contains a credit identity, carrier identity or a stored credit value;
- (d) if a credit identity or carrier identity is found in (c) send it to the process managing the store of bad payees for a check that it is not on the list of bad toll payments;
- (e) if a match is found in (d), get an image of the violator and output the data flow that requests the vehicle to pull in;
- (f) if a stored credit value is found in (c) and it is less than the toll cost, get an image of the violator and output the data flow that requests the vehicle to pull in;
- (g) if the stored credit value found in (c) is greater than or equal to the toll cost, send the output flow to the payment instrument requesting that the toll cost be deducted from the credit being stored by the instrument;
- (h) if a negative response is received to (g), get an image of the violator and output the data flow that requests the vehicle to pull in;
- (i) when the toll transaction is complete always output details of the transaction in the flow of current toll transactions and send out the data flow that clears the toll tag data store;
- (j) if the payment is identified as being for an advanced toll, and a match is found in (d) or the stored credit value is less than the toll cost then set the output flow of advanced toll payment to false and take no further action;
- (k) if the tests in (j) are clear then set the output flow of advanced toll payment to true and enter the credit and vehicle identities into the store containing the advanced payment list;
- (l) use the appropriate mechanism(s) such as RDBMS, to retrieve and write data to the store identified above;
- (m) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

7.1.1.6 Collect Probe Data From Toll Transactions

Input Flows

toll_transactions_for_probe_data

Output Flows

probe_data_for_traffic

toll_transactions_for_probe_data_request

vehicle_toll_probe_data

Description:

Overview: This process shall be responsible for calculating the time taken for vehicles to travel between successive toll plazas and sending 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

(a)'toll_transaction_records'.

Functional Requirements: This process shall meet the following functional requirements:

(a) periodically generate the toll probe data request data flow shown above;

(b) when the toll probe data flow is received, in response to (a), calculate the average journey times between toll plazas where there is sufficient data to enable an average to be sensibly computed;

(c) when (b) is completed, generate the two outputs of probe data identified above;

(d) remove any credit and/or vehicle identity plus other payment information from the data obtained in the input data flow identified above.

7.1.1.7 Update Toll Price Data

Input Flows

fto_local_toll_price_variations

ftsp_toll_price_changes_response

ftsp_toll_price_data

toll_price_changes_request

toll_price_data_request

toll_prices

Output Flows

toll_price_changes_response

toll_price_data

ttsp_toll_price_changes_request

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 service provider to enable changes to be made to the stored data. The process shall support inputs from the toll service provider 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

(a)'toll_prices'.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the input flows listed above;

(b) when any of the inputs from the toll service provider are received, generate the appropriate output identified above;

- (c) when the input for a toll prices change request is received from the Manage Transit function, generate the change request output to the toll service provider;
- (d) when a request for toll price data is received, generate the data flow containing the copy of the store of price data;
- (e) be responsible for the management of the data in the store of toll cost data, using the most appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 407 of 475

7.1.1.8 Register for Advanced Toll Payment

Input Flows

advanced_other_tolls_request
 advanced_traveler_tolls_request
 confirm_advanced_tolls_payment
 ftsp_confirm_advanced_toll

Output Flows

advanced_other_tolls_confirm
 advanced_toll_needed
 advanced_traveler_tolls_confirm
 ttsp_credit_identity
 ttsp_toll_segments
 ttsp_vehicle_identity

Description:

Overview: This process shall be responsible for responding to requests for tolls to be paid in advance. It shall provide the toll service provider with the opportunity to deny the request for an advanced toll. 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs requesting advanced tolls are received, generate the outputs to the toll service provider identified above;
- (c) if no response is received to the flows output in (b), assume that the advanced tolls have been accepted and send the data to the advanced toll bill determination process;
- (d) if a negative response is received to the flows in (b), then output the advanced toll response data flows with the confirmation data set to fail;
- (e) when the confirm data flow is received from the bill driver for tolls process, then output the advanced toll response data flows with the confirmation data set to true;
- (f) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 408 of 475

7.1.1.9 Manage Toll Financial Processing

Input Flows

advanced_toll_transactions
 current_toll_transactions
 ffi_confirm_toll_payment
 toll_transaction_records
 toll_transactions_for_probe_data_request

Output Flows

tfi_request_toll_payment
 toll_operational_data
 toll_transactions_for_probe_data
 tto_transaction_reports
 ttsp_transaction_reports

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 facility. At periodic intervals the process shall output the accumulated records to the toll service provider, the toll operator and the Plan System Deployment function. It shall also output the data on request to the process that calculates

probe data from the average journey 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 first two outputs shall be produced by the process in visual form, which may be either hardcopy or a visual display. 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

(a) 'toll_operational_data' - which is an unsolicited data flow periodically sent to the Plan System Deployment function.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs of transaction records are received, load the data in the data store served by the 'inout' flow shown in the above list;
- (c) when requested by the appropriate input, generate the outputs identified above;
- (d) be responsible for the management of the data in the store of toll transaction records, using the most appropriate mechanism(s) such as RDBMS, for storing the data;
- (e) remove all driver identities from the data before it is used in any output data flows;
- (f) ensure that all output flows are encrypted in such a way that it is not possible to determine the financial data being transmitted, using any form of digital or analog techniques;
- (g) periodically (not less than once per day) send the toll operational data flow to the Plan System Deployment function calculating the number of users for each toll segment from the collected toll costs.

PROCESS SPECIFICATIONS

Page 409 of 475

7.1.2 Produce Roadside Displays

Input Flows

toll_payment_pull_in_message
toll_tag_pull_in_message

Output Flows

td_toll_payment_confirmed
td_toll_payment_invalid

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 process shall provide its outputs in audible and visual forms, with the latter using 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flows listed above;
- (b) when either of the inputs is received, generate the appropriate output identified above.

PROCESS SPECIFICATIONS

Page 410 of 475

7.1.3 Obtain Toll Violator Image

Input Flows

From_Vehicle_Characteristics
get_toll_payment_violator_image
get_toll_tag_violator_image

Output Flows

toll_violation_information

Description:

Overview: This process shall be responsible for obtaining an image of a violator for use by other processes. The form which the image data is obtained by this process shall be so accurate 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 any punitive action that 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flows listed above;
- (b) when the input data flows requesting a violator image be obtained are received, read the vehicle characteristics data being received;
- (c) use the data in (b) to generate a highly accurate image of the vehicle and/or its driver;
- (d) output the image in the violation information flow identified above.

PROCESS SPECIFICATIONS

Page 411 of 475

7.1.4 Provide Driver Toll Payment Interface

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

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 for which advanced payment is supported by the process shall include parking lot charges and transit fares. The process shall interrogate 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 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 either in hardcopy, or as a display. Both types of output shall be produced in such a way that they do not impair the driver's ability to control the vehicle in a manner that is safe to both its occupants, to other toll plaza users, and to staff operating the plaza.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the driver, the basic vehicle and the driver's payment credit identity input flows are received, check the content of the input data to ensure that there is sufficient data to enable the advanced parking lot charge and/or transit fare to be determined;
- (c) if the check in (b) is not confirmed, output an insufficient data to complete transaction message to the driver using the appropriate output flow and highlighting the missing data;
- (d) the data required in (b) about the advanced booking shall include such things as the identity of the parking lot plus the date and time for which a space is required, and/or the origin and destination of the transit route;
- (e) the data in (b) must in addition to that for the advanced booking request, also include the vehicle identity and either the driver's credit identity or the credit, stored on the payment instrument being used by the driver;
- (f) if the check in (f) is not confirmed, output an insufficient financial data (credit identity or stored credit) to complete transaction message to the driver using the appropriate output flow;
- (g) if advanced parking lot booking is required output the appropriate flow to generate any required charges only, using the new value of the stored credit if advanced tolls were required;
- (h) if advanced parking lot booking was required in (g), when a response is received again reduce the value of the stored credit;
- (i) if an advanced transit fare is to be paid, output the appropriate flow to generate any advanced transit fares, using the new value of the stored credit generated by any advanced costs already

produced from the above;

(j) when output and response to all the advanced booking requests are complete, total up all the costs and check that they are less than the original value of the stored credit and if not output an insufficient credit message to the driver using the appropriate output flow;

(k) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 412 of 475

7.1.5 Detect Vehicle for Tolls

Input Flows

fbv_vehicle_characteristics

Output Flows

vehicle_toll_characteristic_data

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.

Data Flows: Input data flow is unsolicited and output flow is solicited.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the input data flow listed above;

(b) when the input is received, generate the output data flow identified above.

PROCESS SPECIFICATIONS

Page 413 of 475

7.1.6 Distribute Advanced Charges and Fares

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

Description:

Overview: This process shall be responsible for receiving requests for the advanced payment of tolls from the parking lot charge and 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 details of the success or failure of the transaction to the process from which the request was received. 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 payment process.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the input flows listed above;

(b) when the inputs from other processes in the Provide Electronic Payment Services function are received, generate the advanced toll payment request output identified above;

(c) when a response is received to the flow in (b), return it in the flow to the requesting process in the Provide Electronic Payment Services function;

(d) when the inputs requesting advanced parking lot charge or transit fare payment are received from the toll payment interface process, generate the appropriate transfer flows to other processes in the Provide Electronic Payment Services function;

(e) when a response is received to the flows in (d), return it in the flow to the requesting process;

(f) all input and output flows must be encrypted in such a way that it is not possible to determine

7.1.7 Provide Payment Instrument Interface for Tolls

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_read
toll_tag_data_request
toll_tag_data_update

Output Flows

driver_toll_payment_credit_identity
toll_payment_confirmation
toll_tag_data_collect
toll_tag_data_input
toll_tag_data_write
tpi_debited_payment_at_toll_plaza
tpi_request_payment_at_toll_plaza

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.

Data Flows: All input and output data flows are solicited with the exception of the following which are unsolicited input flows which activate the process, or are used to read and write data from a local data store:

- (a)'toll_tag_data_request'- which is a data flow received from the toll tag data collection process;
- (b)'toll_tag_data_needed'- which is a data flow received from the Manage Traffic function that contains a request for toll tag data;
- (c)'fpi-toll_tag_data'- which is a data flow received from the payment instrument terminator;
- (d)'toll_tag_data_read'- which reads data from a local data store;
- (e)'toll_tag_data_write'- which writes data to a local data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the first unsolicited input data flow in (a) is received, generate the output data flow containing the toll tag data;
- (c) as a result of (b) await receipt of the data flows that either request toll payment from the credit stored on the payment instrument, or confirm that payment will be deducted from the credit identity supplied by the payment instrument, or contain an updated version of the toll tag data with the identity of the toll segment at which the vehicle entered the toll road;
- (d) when either of the two flows in (c) is received, send the appropriate toll payment request or payment debited flow to the payment instrument;
- (e) when the third flow in (c) is received, update the local store of toll tag data;
- (f) when the second unsolicited data flow in (a) is received, send the data flow containing tag data to the Manage Traffic function;
- (g) when the third unsolicited input flow in (a) is received, write the data to the local data store replacing any data that already exists in the store;

- (h) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques;
- (i) use the most appropriate mechanism(s) such as RDBMS, to manage the data in the store identified above.

PROCESS SPECIFICATIONS

Page 415 of 475

PROCESS SPECIFICATIONS

Page 416 of 475

7.2.1.1 Read Parking Lot Tag Data

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

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 be 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 and 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.

Data Flows: All input and output data flows are solicited with the exception of the following which is used to trigger the process:

- (a)'vehicle_parking_lot_characteristic_data'- which is received from another process that detects a vehicle's presence.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 417 of 475

7.2.1.10 Determine Advanced Charges

Input Flows

advanced_charges_needed
parking_lot_prices

Output Flows

advanced_parking_lot_billing

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

- (a)'parking_lot_prices'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) all input and output flows must be encrypted in such a way that it is not possible to determine

the payment information being transmitted, using any form of digital or analog techniques;
(d) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 418 of 475

7.2.1.2 Calculate Vehicle Parking Lot Charges

Input Flows

parking_lot_prices
vehicle_tag_for_charges
vehicle_type_for_charges

Output Flows

parking_lot_charge

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

(a)'parking_lot_prices'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 419 of 475

7.2.1.3 Collect Bad Charge Payment Data

Input Flows

charge_payment_violator_data
ffi_bad_charges_payment_updates

Output Flows

bad_charge_payment_list
tfi_parking_lot_payment_violator_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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data written to a data store:

(a)'bad_charge_payment_list'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) be responsible for the management of the data in the store of bad driver payment instrument data, using the most appropriate mechanism(s) such as RDBMS, for storing the data.
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 420 of 475

7.2.1.4 Check for Advanced Parking Lot Payment

Input Flows

advanced_charges_payment_list
advanced_parking_lot_billing
parking_lot_charge

Output Flows

advanced_charge_transactions
billing_for_charges_needed

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 advance 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from and written to a data store:
(a)'advanced_charges_payment_list'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) be responsible for the management of the data in the store of advanced charges payment data, using the most appropriate mechanism(s) such as RDBMS, for storing the data;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques;
- (e) remove the payment information received from the driver from all data that is sent to another process for loading into the store of parking lot charge payment transactions.

PROCESS SPECIFICATIONS

Page 421 of 475

7.2.1.5 Bill Driver for Parking Lot Charges

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

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 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 last of bad payers, or the stored credit is not less than 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

- (a)'bad_charge_payment_list', which contains data requested from a data store;
- (b)'advanced_charges_payment_list', which contains data written to a data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) when the unsolicited flow requesting billing for parking lot charges is received, check to see if it contains a credit identity or a stored credit value;
- (d) if a credit identity is found in (c) that check it against those held in the store containing a list of bad charge payments;
- (e) if a match is found in (d), get an image of the violator and output the data flow that requests the vehicle to pull in;
- (f) if a stored credit value is found in (c) and it is less than the parking lot charge, get an image of the violator and output the data flow that requests the vehicle to pull in;
- (g) if the stored credit value found in (c) is greater than or equal to the parking lot charge, send the output flow to the payment instrument requesting that the charge be deducted from the credit being stored by the instrument;
- (h) if a negative response is received to (g), get an image of the violator and output the data flow that requests the vehicle to pull in;
- (i) when the parking lot charge transaction is complete always output details of the transaction in the flow of current charge transactions and send out the data flow that clears the parking lot tag data store;
- (j) if the payment is identified as being for an advanced charge, and a match is found in (d) or the stored credit value is less than the charge, then set the output flow of advanced parking lot charge payment to false and take no further action;
- (k) if the tests if (j) are clear then set the output flow of advanced parking lot charge payment to true and enter the credit and vehicle identities into the store containing the advanced payment list;
- (l) use the appropriate mechanism(s) such as RDBMS, to retrieve from and write data from to the store identified above;
- (m) all input and output flows must be encrypted in such a way that it is not possible to determine

User Service Requirements: the payment information being transmitted, using any

form of digital or analog techniques.

USR = 3.0;

USR = 3.1;

USR = 3.1.3;

USR = 3.1.3.1;

PROCESS SPECIFICATIONS

PROCESS SPECIFICATIONS

Page 422 of 475

Page 423 of 475

7.2.1.6 Manage Parking Lot Financial Processing

Input Flows

advanced_charge_transactions
current_charge_transactions
ffi_confirm_charges_payment
fppsp_transaction_reports_request
parking_lot_transaction_records

Output Flows

tfi_request_charges_payment
tppsp_transaction_reports

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 service provider, either in hardcopy form, or as a visual display. Both audible and manual inputs from the parking service provider shall be supported by the process. 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data written to and requested from a data store:
(a)'parking_lot_transaction_records'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs of transaction records are received, load the data in the store of transaction records;
- (c) when requested by the appropriate input, generate the outputs identified above;
- (d) be responsible for the management of the data in the store of parking lot charge transaction records, using the most appropriate mechanism(s) such as RDBMS, for storing the data;
- (e) remove all driver identities from the data before it is loaded into the data store;
- (f) ensure that all output flows are encrypted in such a way that it is not possible to determine the financial data being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 424 of 475

7.2.1.7 Update Parking Lot Data

Input Flows

fpsp_parking_lot_charge_change_response
fpsp_parking_lot_data
parking_lot_capacity_update_confirm
parking_lot_charge_change_request
parking_lot_price_data_request

Output Flows

parking_lot_capacity_update
parking_lot_charge_change_response
parking_lot_price_data
parking_lot_prices
tspsp_parking_lot_charge_change_request

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 process shall support inputs from the parking service provider 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:
(a)'parking_lot_prices'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when any of the inputs from the parking service provider are received, generate the appropriate output identified above;
- (c) when the input for a parking lot prices change request is received from the Manage Transit function, generate the change request output to the parking service provider;
- (d) when a request for parking lot price data is received, generate the data flow containing the copy of the store of price data;
- (e) be responsible for the management of the data in the store of parking lot price data, using the appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 425 of 475

7.2.1.8 Register for Advanced Parking Lot Payment

Input Flows

advanced_other_charges_request
advanced_parking_lot_charges_request
advanced_traveler_charges_request
confirm_advanced_charges_payment
fppsp_confirm_advanced_parking_payment
parking_lot_bookings_confirm

Output Flows

advanced_charges_needed
advanced_other_charges_confirm
advanced_parking_lot_charges_confirm
advanced_traveler_charges_confirm
parking_lot_bookings_request
tppsp_request_advanced_parking_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 service provider 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs requesting advanced parking lot charges are received, generate the outputs to the parking service provider identified above;
- (c) if no response is received to the flows output in (b), assume that the advanced parking lot charges have been accepted and send the data to the advanced charge determination process;
- (d) if a negative response is received to the flows in (b), then output the advanced parking lot charge response data flows with the confirmation data set to fail;
- (e) when the confirm data flow is received from the bill driver for parking lot charges process, then output the advanced parking lot charge response data flows with the confirmation data set to true;
- (f) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 426 of 475

7.2.1.9 Manage Parking Lot Reservations

Input Flows

parking_lot_bookings_request
parking_lot_capacity_update
parking_lot_data
parking_lot_data_request
parking_lot_reservation_request

Output Flows

parking_lot_availability
parking_lot_bookings_confirm
parking_lot_capacity_update_confirm
parking_lot_reservation_confirm

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the

exception of the following which contains data requested from a data store:
(a)'parking_lot_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) be responsible for the management of the data in the store of parking lot data, using the appropriate mechanism(s) such as RDBMS, for storing the data.
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 427 of 475

7.2.2 Produce Parking Lot Displays

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

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 process shall provide its outputs in audible and visual forms, with the latter using 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above.

PROCESS SPECIFICATIONS

Page 428 of 475

7.2.3 Obtain Parking Lot Violator Image

Input Flows

From_Vehicle_Characteristics
get_charge_payment_violator_image
get_parking_lot_tag_violator_image

Output Flows

parking_lot_violation_information

Description:

Overview: This process shall be responsible for obtaining an image of a violator for use by other processes. The form which the image data is obtained by this process shall be so accurate 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 any punitive action that 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the input data flows requesting a violator image be obtained are received, read the vehicle characteristics data being received;
- (c) use the data in (b) to generate a highly accurate image of the vehicle and/or its driver;
- (d) output the image in the violation information flow identified above.

PROCESS SPECIFICATIONS

Page 429 of 475

7.2.4 Provide Driver Parking Lot Payment Interface

Input Flows

advanced_parking_lot_charges_confirm
advanced_tolls_and_fares_response
driver_parking_payment_credit_identity
fbv_vehicle_identity
fd_other_services_parking_request

Output Flows

advanced_parking_lot_charges_request
advanced_tolls_and_fares_request
driver_advanced_payment_at_lot
td_other_services_parking_response

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. These services for which advanced payment is supported by the process shall include advanced parking lot payment, as well as advanced payment for tolls and transit fares. The process shall interrogate 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 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 either in hardcopy, or as a display. Both types of output shall be produced in such a way that they do not impair the driver's ability to control the vehicle in a manner that is safe to both its occupants, to other parking lot users, and staff operating the parking lot.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the driver, the basic vehicle and the driver's payment credit identity input flows are received, check the content of the input data to ensure that there is sufficient data to enable the advanced parking lot charge, toll and/or transit fare to be determined;
- (c) if the check in (b) is confirmed, then output the appropriate flow to generate any required advanced parking lot charges only;
- (d) if the check in (b) is not confirmed, output an insufficient data to complete transaction message to the driver using the appropriate output flow, and highlighting the missing data;
- (e) the data required in (b) about the advanced booking shall include such things as the identity of the parking lot plus the date and time for which a space is required, and/or the toll segments to be traversed, and/or the origin and destination of the transit route;
- (f) the data in (b) must in addition to that for the advanced booking request, also include the vehicle identity and either the driver's credit identity or the credit, stored on the payment instrument being used by the driver;
- (g) if the check in (f) is not confirmed, output an insufficient financial data (credit identity or stored credit) to complete transaction message to the driver using the appropriate output flow;
- (h) if advanced parking lot charges were required in (c), when a response is received reduce the value of the stored credit;
- (i) if advanced toll payment is required output the appropriate flow to generate any required tolls only, using the new value of the stored credit if advanced parking lot charges were required;
- (j) if advanced toll payment was required in (i), when a response is received again reduce the value of the stored credit;
- (k) if an advanced transit fare is to be paid, output the appropriate flow to generate any advanced transit fares, using the new value of the stored credit generated by any advanced costs already produced from the above;
- (l) when output and response to all the advanced booking requests are complete, total up all the costs and check that they are less than the original value of the stored credit and if not output an insufficient credit message to the driver using the appropriate output flow;
- (m) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 430 of 475

7.2.5 Detect Vehicle for Parking Lot Payment

Input Flows

fbv_vehicle_characteristics

Output Flows

vehicle_parking_lot_characteristic_data

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above.

PROCESS SPECIFICATIONS

Page 431 of 475

7.2.6 Distribute Advanced Tolls and Fares

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

Description:

Overview: This process shall be responsible for receiving requests for the advanced payment of parking lot charges from the toll and 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 details of the success or failure of the transaction to the process from which the request was received. 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 payment process.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs from other processes in the Provide Electronic Payment Services function are received, generate the advanced parking lot charge request output identified above;
- (c) when a response is received to the flow in (b), return it in the flow to the requesting process in the Provide Electronic Payment Services function;
- (d) when the inputs requesting advanced toll or transit fare payment are received from the parking lot payment interface process, generate the appropriate transfer flows to other processes in the Provide Electronic Payment Services function;
- (e) when a response is received to the flows in (d), return it in the flow to the requesting process;
- (f) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 432 of 475

7.2.7 Provide Payment Instrument Interface for Parking

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

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.

Data Flows: All input and output data flows are solicited with the exception of the following which are unsolicited input flows which activates the process, or are used to read and write data from a local data store:

- (a)'parking_lot_tag_data_request'- which is a data flow received from the parking lot tag data collection process;
- (b)'parking_lot_tag_data_needed'- which is a data flow received from the Manage Traffic function that contains a request for toll tag data;
- (c)'fpi-parking_tag_data'- which is a data flow received from the payment instrument terminator;
- (d)'parking_lot_tag_data_read'- which reads data from a local data store;
- (e)'parking_lot_tag_data_write'- which writes data to a local data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) when the first unsolicited input data flow in (s) is received, generate the output data flow containing parking lot tag data;
- (c) as a result of (b) await receipt of the data flows that either request parking lot charge payment from the credit stored on the payment instrument, or confirm that payment will be deducted from the credit identity supplied by the payment instrument, or contain an updated version of the parking lot tag data with the time at which the vehicle entered the parking lot and the lot entry lane number;
- (d) when either of the first two flows in (c) is received, send the appropriate parking lot charge payment request or payment debited flow to the payment instrument;
- (e) when the third flow in (c) is received, update the local store of parking lot tag data;
- (f) when the second unsolicited data flow in (a) is received, send the data flow containing tag data to the Manage Traffic function;
- (g) when the third unsolicited input flow in (a) is received, write the data to the local data store replacing any data that already exists in the store;
- (h) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques;
- (i) use the most appropriate mechanism(s) such as RDBMS, to manage the data in the store identified above.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 433 of 475
Page 434 of 475

7.3.1.1 Register for Advanced Transit Fare Payment

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 435 of 475

7.3.1.2 Determine Advanced Transit Fares

Input Flows

advanced_fares_needed
transit_fares_for_advanced_payments
transit_services_for_advanced_fares

Output Flows

advanced_fare_billing

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

- (a)'transit_fares_for_advanced_payments'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques;
- (d) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 436 of 475

7.3.1.3 Manage Transit Fare Financial Processing

Input Flows

advanced_fare_transactions
current_fare_transactions
ffi_confirm_fare_payment
transit_fare_transaction_records

Output Flows

tfi_request_fare_payment
ttfm_transaction_reports
ttso_transaction_reports

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 first two outputs shall be produced by the process in visual form, which may be either hardcopy or a visual display. 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data written to and requested from a data store:
(a)'transit_fare_transaction_records'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, load the data in the store of transaction records;
- (c) at periodic intervals, e.g. daily, generate the outputs identified above;
- (d) be responsible for the management of the data in the store of transit fare transaction records, using the appropriate mechanism(s) such as RDBMS, for storing the data;
- (e) remove all transit user identities from the data before it is loaded into the data store;
- (f) ensure that all output flows are encrypted in such a way that it is not possible to determine the financial data being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 437 of 475

7.3.1.4 Check for Advanced Transit Fare Payment

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:
(a)'advanced_fare_payment_list'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques;
- (e) remove the credit identity of the transit user from all data that is sent to another process for loading into the store of transit fare payment transactions.

PROCESS SPECIFICATIONS

Page 438 of 475

7.3.1.5 Bill Transit User for Transit Fare

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

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 farepayments 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

- (a)'bad_fare_payment_list', which contains data requested from a data store;
- (b)'advanced_fare_payment_list', which contains data written to a data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above that contain data about fare billing that is needed, or is a request for a copy of the bad payers list;
- (b) when the data about fare billing is received, check any credit identity that it contains against the list of bad payers;
- (c) if a match is found in (b) then output the data flow requesting a violator image and return a fare payment failed message to the source of the billing information;
- (d) if no match is found in (b) then either request that the tag's stored credit be reduced by the amount of the fare, or send the payment information to another process for debiting via the financial institution;
- (e) as a result of (d), send a fare payment successful message to the source of the billing information;
- (f) if the deduction of the fare from the tag's stored credit in (d) fails then output the data flow requesting a violator image and return a fare payment failed message to the source of the billing information;
- (g) if in (d) the payment information is sent to another process for output to the financial institution, then output a fare payment debited message to the source of the billing information;
- (h) if the billing information was for the advanced payment of a transit fare, then on successful completion of the payment transaction in the previous steps, store the data about the source of the payment and the fare to which it applies in the store of advanced payments;
- (i) when the data flow received in (a) is a request for a copy of the bad payers list, read the current list from the data store and send it to the requesting process;
- (j) use the most appropriate mechanism(s) such as RDBMS, to retrieve and write data from and to the

User Service Requirements:

stores identified above;

USR = 3.0; (k) all input and output flows must be encrypted in such a way that it is not possible to determine

USR = 3.1; the payment information being transmitted, using any form of digital or analog techniques.

USR = 3.1.2;

USR = 3.1.2.1;

USR = 3.1.2.3;

USR = 3.1.2.8;

PROCESS SPECIFICATIONS

Page 439 of 475

PROCESS SPECIFICATIONS

Page 440 of 475

7.3.1.6 Collect Bad Transit Fare Payment Data

Input Flows

fare_payment_violator_data

ffi_bad_fare_payment_updates

Output Flows

bad_fare_payment_list

tfi_fare_payment_violator_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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data written to a data store:

(a)'bad_fare_payment_list'.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the input flows listed above;

(b) when the inputs are received, generate the outputs identified above;

(c) be responsible for the management of the data in the store of bad transit user payment instrument data, using the most appropriate mechanism(s) such as RDBMS, for storing the data.

(d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 441 of 475

7.3.1.7 Update Transit Fare Data

Input Flows

ftso_fare_updates

ftso_request_fare_output

transit_fare_data_request

Output Flows

transit_fare_data

transit_fares_for_advanced_payments

transit_roadside_fare_data

transit_vehicle_fare_data

ttso_transit_fare_output

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the

exception of the following which contains data written to a data store:

(a)'transit_fares_for_advanced_payments'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flows from the transit system operator listed above;
- (b) when the input requesting output of the current data is received, generate the output of fares to the operator using the data flow identified above;
- (c) when the input containing new fare data is received, load the new data into the data store, and generate the other output data flows identified above.
- (d) when a request for transit fare price data is received, generate the data flow containing the copy of the store of price data;
- (e) use the most appropriate mechanism(s) such as RDBMS, to write data to the store of transit fares identified above.

PROCESS SPECIFICATIONS

Page 442 of 475

7.3.2 Distribute Advanced Tolls and Parking Lot Charges

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

Description:

Overview: This process shall be responsible for receiving requests for the 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 details of the success or failure of the transaction to the process from which the request was received. The process shall also receive requests for the advanced payment of tolls and parking lot charges from transit vehicle and roadside (transit stop) fare collection facilities within the Manage Transit function. It shall send these requests to other processes in the Provide Electronic Payment Services function and when received, return the results to the process in the Manage Transit function from which the request was received.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flows listed above;
- (b) when the inputs from other processes in the Provide Electronic Payment Services function are received, generate the advanced fare request output identified above;
- (c) when a response is received to the flow in (b), return it in the flow to the requesting process in the Provide Electronic Payment Services function;
- (d) when the inputs requesting advanced toll or parking lot charge payment are received, generate the appropriate transfer flows to other processes in the Provide Electronic Payment Services function;
- (e) when a response is received to the flows in (d), return it in the flow to the requesting process in the transit vehicle or roadside;
- (f) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 443 of 475

7.3.3 Get Transit User Image for Violation

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

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 processes in the Manage Emergency Services function for forwarding to the appropriate enforcement agency.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the input requesting a violator image is received, check to see if the roadside or vehicle identity is set, i.e. non-zero;
- (c) generate the output to the vehicle or roadside using the data flows identified above depending on which identity is set;
- (d) when the input corresponding to the output in (b) is received, generate the fare violation information output identified above to the process in the Manage Emergency Services function;
- (e) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 444 of 475

7.3.4 Provide Remote Terminal Payment Instrument Interface

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

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 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, 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.

Data Flows: All input and output data flows are solicited with the exception of the following which is an unsolicited input flow and which activates the process:

- (a)'fpi-transit_roadside_tag_data'- which is a flow received from the payment instrument terminator.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) when the input is received, generate the output data flow containing transit roadside tag data;
- (c) as a result of (b) await receipt of the data flows that either request transit fare payment from the credit stored on the payment instrument or confirm that payment will be deducted from the credit

identity supplied by the payment instrument;

(d) when either of the flows in (c) is received, send the appropriate transit fare payment request or payment debited flow to the payment instrument;

(e) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog technique.

PROCESS SPECIFICATIONS

Page 445 of 475

7.3.5 Provide Transit Vehicle Payment Instrument Interface

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

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.

Data Flows: All input and output data flows are solicited with the exception of the following which is an unsolicited input flow and which activates the process:

(a)'fpi-transit_vehicle_tag_data'- which is a flow received from the payment instrument terminator.

Functional Requirements: This process shall meet the following functional requirements:

(a) continuously monitor for receipt of the unsolicited input flow listed above;

(b) when the input is received, generate the output data flow containing transit vehicle tag data;

(c) as a result of (b) await receipt of the data flows that either request transit fare payment from the credit stored on the payment instrument or confirm that payment will be deducted from the credit identity supplied by the payment instrument;

(d) when either of the flows in (c) is received, send the appropriate transit fare payment request or payment debited flow to the payment instrument;

(e) all input and output flows must be encrypted in such a way that it is not possible to determine the credit identity or stored credit value being transmitted using any form of digital or analogue technique.

PROCESS SPECIFICATIONS

Page 446 of 475

7.4.1.1 Process Commercial Vehicle Payments

Input Flows

ffi_cv_payment_confirm

financial_request

Output Flows

financial_response

tfi_cv_payment_request

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flows listed above;
- (b) when the input is received from the Manage Commercial Vehicles function, generate the output to the financial institution identified above;
- (c) as a result of (b) await the response from the financial institution and when received, use it to generate the reply to the requesting process;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 447 of 475

7.4.1.2 Process Yellow Pages Services Provider Payments

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

Description:

Overview: This process shall be responsible for transacting payments for the registration of other (yellow pages) service providers. 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. 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input data flows listed above;
- (b) when the input is received from the Provide Driver and Traveler Services function, generate the output to the financial institution identified above;
- (c) as a result of (b) await the response from the financial institution and when received, use it to generate the reply to the requesting process;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 448 of 475

7.4.1.3 Process Driver Map Update Payments

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

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 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the input is received from the Provide Driver and Traveler Services function, generate the output to the financial institution identified above;

- (c) as a result of (b) await the response from the financial institution and when received, use it to generate the reply to the requesting process;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 449 of 475

7.4.1.4 Process Traveler Map Update Payments

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

Description:

Overview: This process shall be responsible for transacting payments from the traveler for updates to the navigable map database carried in the personal portable device. 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. 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the input is received from the Provide Driver and Traveler Services function, generate the output to the financial institution identified above;
- (c) as a result of (b) await the response from the financial institution and when received, use it to generate the reply to the requesting process;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 450 of 475

7.4.1.5 Process Transit User Other Services Payments

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

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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the input is received from the Manage Transit function, generate the output to the

financial institution identified above;

(c) as a result of (b) await the response from the financial institution and when received, use it to generate the reply to the requesting process;

(d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 451 of 475

7.4.1.6 Process Traveler Trip and Other Services Payments

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

Description:

Overview: This process shall be responsible for transacting advance 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. 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the input is received from the Provide Driver and Traveler Services function, generate the output to the financial institution identified above;
- (c) as a result of (b) await the response from the financial institution and when received, use it to generate the reply to the requesting process;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 452 of 475

7.4.1.7 Collect Payment Transaction Records

Input Flows

driver_map_update_payments_transactions
payment_transaction_records
transit_user_payments_transactions
traveler_map_update_payments_transactions
traveler_rideshare_payments_transactions
traveler_trip_payments_transactions
yellow_pages_provder_payments_transactions

Output Flows

payment_transaction_reports

Description:

Overview: This process shall be responsible for the collection and maintenance of a store that contains transaction records for payments made to enable various services to be provided. The process shall load into the store payment transaction records for services comprising electronic credential and tax filing purchase for commercial vehicle operations, update of map databases for drivers and travelers, registration of other (yellow pages) services providers so that information about what they have to offer is available to travelers and transit users, plus the advanced payment of tolls, parking lot charges, transit fares and for 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 source of payment, i.e. driver, traveler and commercial vehicle fleet manager, plus any other payment information removed. A complete record of all transactions shall be periodically loaded into a store of financial and ridership data by this process.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data store:

(a)'payment_transaction_records'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the input is received, generate the outputs identified above
- (c) be responsible for the management of the data in the store of payment transactions, using the most appropriate mechanism(s) such as RDBMS, for storing the data;
- (d) ensure that all references to the identity of those making the payments to which the transaction records relate are removed from the data before it is loaded into the store identified above;
- (e) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 453 of 475

7.4.1.8 Process Traveler Rideshare Payments

Input Flows

ffi_traveler_rideshare_payment_confirm
rideshare_payment_request

Output Flows

rideshare_payment_confirmation
tfi_traveler_rideshare_payment_request
traveler_rideshare_payments_transactions

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 receive the transaction 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flows listed above;
- (b) when the input is received from the Provide Driver and Traveler Services function, generate the output to the financial institution identified above;
- (c) as a result of (b) await the response from the financial institution and when received, use it to generate the reply to the requesting process;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 454 of 475

7.4.2 Collect Price Data for ITS Use

Input Flows

parking_lot_charge_request
parking_lot_price_data
price_data_for_services
request_prices
toll_price_data
toll_price_request
transit_fare_data
transit_fare_request

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

Description:

Overview: This process shall be responsible for collecting financial data on the prices being charged for tolls, parking lots and transit fares. This data shall be sent to this process by other processes in the parts of the Provide Electronic Payment Services function responsible for tolls, parking lot charges and transit fares. This shall be because either they have updated their data, or they have received a request for data from this process. When received by the process, the data shall be loaded into a store from which some or all of it can be read on request from processes in other ITS functions.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data written to and requested from a data store: (a)'financial_and_ridership_data'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flows listed above;
- (b) when inputs of data are received, load their contents into the data store identified above;
- (c) when the inputs requesting data are received, read the required data from the store and generate the output data flows identified above;
- (d) periodically generate the data flows listed above requesting the current toll and parking lot prices plus transit fare data;
- (e) be responsible for the management of the data in the store of the prices for services, using the most appropriate mechanism(s) such as RDBMS, for storing the data;
- (f) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS
PROCESS SPECIFICATIONS

Page 455 of 475
Page 456 of 475

7.4.3 Route Traveler Advanced Payments

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

Description:

Overview: This process shall be responsible for receiving a traveler's request for advanced payment 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 processes.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the traveler advanced payments request flow listed above;
- (b) when the flow in (a) is received, interrogate the data and generate the appropriate advanced fare, toll or charge request flow identified above;
- (c) when all of the resulting responses have been received, generate the traveler advanced payments response flow listed above;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 457 of 475

7.5.1 Provide Vehicle Payment Instrument Interface

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

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 vehicle from which data collection is supported by the process shall comprise 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) when the input is received, generate the appropriate one of the outputs identified above;
- (c) when either of the input flows requesting a reduction in stored credit is received, generate the appropriate flow to reduce the credit currently stored on-board the payment instrument;
- (d) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 458 of 475

7.5.2 Provide Transit User Roadside Payment Instrument Interface

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

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 shall be a transit stop. Payments by the transit user for fares and if needed, other services shall be supported by the process, as well as for the payment of advanced tolls, and/or parking lot charges. It shall also provide an interface through which the stored credit held by the tag can be debited for the same types of payment.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) when the input is received, generate the appropriate one of the outputs identified above;
- (c) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 459 of 475

7.5.3 Provide Personal Payment Instrument Interface

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

Description:

Overview: This process shall be responsible for providing the interface through which the credit identity or stored credit may be collected from the tag being used by a traveler with a personal portable 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, and/or parking lot charges, and/or transit fares. The process shall also enable the stored credit value on the tag to be used for the same purposes.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) when the input is received, generate the appropriate one of the outputs identified above;
- (c) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 460 of 475

7.5.4 Provide Commercial Fleet Payment Instrument Interface

Input Flows

cf_manager_enrollment_cost
fpi_commercial_manager_input_credit_identity

Output Flows

cf_manager_credit_identity
tpi_debited_commercial_manager_payment

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 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) when the input is received, generate the appropriate one of the outputs identified above;
- (c) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 461 of 475

7.5.5 Provide Traveler Kiosk Payment Instrument Interface

Input Flows

fpi_traveler_roadside_input_credit_identity
traveler_roadside_trip_costs

Output Flows

tpi_debited_traveler_payment_at_roadside
traveler_roadside_credit_identity

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 travelers. The process shall support the collection of this data at the roadside which in this instance shall be a kiosk and the use of this data for the payments needed to confirm a traveler's trip. Payments

supported by the process shall include those for advanced tolls, and/or parking lot charges, and/or transit fares, plus 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the unsolicited input flow listed above;
- (b) when the input is received, generate the appropriate one of the outputs identified above;
- (c) all input and output flows must be encrypted in such a way that it is not possible to determine the payment information being transmitted, using any form of digital or analog techniques.

PROCESS SPECIFICATIONS

Page 462 of 475

8.1 Import Static and Historical Data

Input Flows

ahs_operational_data
current_incident_static_data
current_other_routes_use
current_road_network_use
current_traffic_static_data
cv_operational_data
emergency_vehicle_operational_data
import_request
imported_map_data
parking_lot_operational_data
payment_transaction_reports
pollution_operational_data
toll_operational_data
traffic_data_for_deployment
transit_passenger_operational_data
transit_services_for_deployment

Output Flows

request_map_data_import
static_data_request
traffic_data_deployment_request
transit_raw_map_data
update_historic_data_store
update_static_data_store

Description:

Overview: This process shall be responsible for collecting static and historical data from other functions within ITS, as well as digitized map data from a specialist supplier. The process shall collect some of the data on request as a result of input received from transportation planners via an interface process, and the rest of the data in the form of regular unsolicited inputs. When received, this process shall send the data to another process for loading into stores that are used by other processes in the function as a source of data for simulation and evaluation of the ITS network.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

- (1) flows received as a result of outputs to processes in this function and in other ITS functions:
 - (a)'current_incident_static_data';
 - (b)'current_traffic_static_data';
 - (c)'imported_map_data';
 - (d)'traffic_data_for_deployment';
- (2) unsolicited flows which are received regularly:
 - (a)'ahs_operational_data';
 - (b)'cv_operational_data';
 - (c)'emergency_vehicle_operational_data';
 - (d)'payment_transaction_reports';
 - (e)'parking_lot_operational_data';
 - (f)'pollution_operational_data';

(g)'toll_operational_data';
(h)'transit_passenger_operational_data';
(i)'transit_services_for_deployment'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the import request flow is received, generate the data request flows identified above;
- (c) when the data resulting from the flows in (b) is received, generate the flows listed above that will send the data to the store data loading processes;
- (d) when any unsolicited data inputs are received, again generate the flows listed above that will send the data to the store data loading processes;
- (e) when the transit services data flow is received, output the data flow containing transit raw map data;

User Service Requirements:

(f) when generating the flows to the store loading processes, apply any necessary data transforms that may be needed to ensure that all the data is sent in a consistent format.

USR = 2.1.2;

USR = 2.1.2.1;

USR = 2.1.2.1.1;

PROCESS SPECIFICATIONS

Page 463 of 475

PROCESS SPECIFICATIONS

Page 464 of 475

8.2.1 Update Data Stores

Input Flows

update_historic_data_store

update_static_data_store

Output Flows

map_data_for_documentation

map_data_for_tp_displays

network_historical_data

network_static_data_store

Description:

Overview: This process shall be responsible for updating the four data stores used by processes in the Plan System Deployment function. The data for loading into the stores shall be provided by the data import process.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above that will load the received data into the store in which it resides;
- (c) be responsible for the management of the data in the stores of digitized map data, historical performance data for traffic, commercial vehicles, transit and emissions monitoring facilities, and the current network static data, using the appropriate mechanism(s) such as RDBMS, for storing the data that they each contain.

PROCESS SPECIFICATIONS

Page 465 of 475

8.2.2 Communicate with Transportation Planner

Input Flows

documentation_output

map_data_for_tp_displays

network_evaluation_results

network_historical_data

network_parameters

network_simulation_data

network_static_data_store

transportation_planner_inputs

Output Flows

documentation_request

network_evaluation_parameters

network_generation_parameters
network_simulation_parameters
transportation_planner_outputs

Description:

Overview: This process shall be responsible for acting as the interface between the transportation planner and other processes in this function. It shall receive transportation planner inputs via an interface process, and shall generate the required outputs, or send data to other processes in response to the inputs. Data for output shall be sent to the transportation planner interface process.

Data Flows: All input data flows are unsolicited and all output flows are solicited.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the inputs are received, generate the outputs identified above;
- (c) be responsible for the management of the data in the store of network planning data, using the appropriate mechanism(s) such as RDBMS, for storing the data;
- (d) use the appropriate mechanism(s) such as RDBMS, to retrieve data from the stores of historical performance data for traffic, commercial vehicles, transit and emissions monitoring facilities, network planning data, and current network static data identified above.

PROCESS SPECIFICATIONS

Page 466 of 475

8.2.3 Generate Static Data

Input Flows

network_generation_parameters
network_historical_data
network_static_data_store

Output Flows

network_static_data

Description:

Overview: This process shall be responsible for providing the facility to generate a new set of static data for use by the Manage Traffic function. The process shall produce a new set of static data on receipt of a set of network generation parameters from the transportation planner, via the interface processes. It shall use the current set of network static data and historical data to produce the new static data.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contain data requested from a data stores:

- (a)'network_historical_data';
- (b)'network_static_data_store'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flows listed above;
- (b) when the input containing the network generation parameters is received, generate the new set of static data by reading the data from the stores, as dictated by the generation parameters;
- (c) when (c) is complete sent the data to the Manage Traffic function, using the output data flow identified above;
- (d) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the stores identified above.

PROCESS SPECIFICATIONS

Page 467 of 475

8.2.4 Evaluate System

Input Flows

network_evaluation_parameters
network_static_data_store

Output Flows

network_evaluation_results

Description:

Overview: This process shall be responsible for the evaluation of the performance of the network of roads and freeways served by the local ITS function. This evaluation shall take into account any

factors that may affect performance such as, weather, travel demand, the desire to reduce emissions, transit services, proportion of guided and non-guided vehicles, historic performance, etc. From the results produced by the process it shall be possible to deduce the effects that any changes to the network have had on its performance in terms of its ability to move people and goods and the efficiency with which this is achieved. The process shall be driven by the input of evaluation parameters received from the transportation planner via the interface processes. The same processes shall be supplied with the results of the evaluation.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from store of network static data:
(a)'network_static_data_store'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the input data flows listed above;
(b) when the input of evaluation parameters is received, perform the network evaluation and generate the evaluation results output data flow identified above;
(c) use one or more specialized algorithms to evaluate the performance of the road and freeway network served by the local ITS function when requested;
(d) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 468 of 475

8.2.5 Simulate System

Input Flows

network_historical_data
network_simulation_parameters
network_static_data_store

Output Flows

network_simulation_data

Description:

Overview: This process shall be responsible for providing a simulation of the way in which the road and freeway network served by the local ITS functions will operate in terms of the traffic volumes, delays, etc. experienced. The simulation performed by the process shall take into account factors which may affect system performance such as weather, travel demand, the desire to reduce emissions, transit services, proportion of guided and non-guided vehicles, historic performance, etc. The process shall be driven by the input of simulation parameters received from the transportation planner via the interface processes. The same processes shall be supplied with the results of the simulation.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contains data requested from a data stores:
(a)'network_historical_data';
(b)'network_static_data_store'.

Functional Requirements: This process shall meet the following functional requirements:
(a) continuously monitor for receipt of the input data flows listed above;
(b) when the input of simulation parameters is received, perform the simulation and generate the simulation results output data flow identified above;
(c) use one or more specialized algorithms to simulate the performance of the road and freeway network served by the local ITS function when requested;
(d) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the stores identified above.

PROCESS SPECIFICATIONS

Page 469 of 475

8.2.6 Document System

Input Flows

documentation_request
map_data_for_documentation
network_historical_data
network_planning_data
network_static_data_store

Output Flows

documentation_output

Description:

Overview: This process shall be responsible for providing the transportation planner with the facility to create proper documentation of the network served by the ITS functions. The process shall provide output that is capable of being used as an aid in communicating to others how the ITS network will work and what improvements are proposed to improve the efficiency with which it moves people and goods and/or reduce the emissions which this movement generates. The process shall be driven by the input of a documentation request received from the transportation planner via the interface processes. The same processes shall be supplied with the output of the generated documentation in electronic form, which it shall be their responsibility to convert into the required output format.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following which contain data requested from a data stores:

- (a)'map_data_for_documentation';
- (b)'network_historical_data';
- (c)'network_planning_data';
- (d)'network_static_data_store'.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flows listed above;
- (b) when the input containing the documentation request is received, generate the documentation output identified above, using data obtained from the historic, static, planning and documentation map data stores;
- (c) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the stores identified above.

PROCESS SPECIFICATIONS

Page 470 of 475

8.3 Export Static Data

Input Flows

link_database
network_static_data

Output Flows

link_data
static_data_for_maps
supply_incident_static_data
supply_traffic_static_data

Description:

Overview: This process shall be responsible for sending a new data to other ITS functions. The process send a copy of the static data to processes in the Manage Traffic function, and a new copy of the database than maps link ownership to a process in the Provide Driver and Traveler Services function. This output shall only be produced by the process on receipt of a new set of static data from the process responsible for its generation.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

- (a)'link_database'which contains data requested from a data store.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flows listed above;
- (b) when the input of new static data is received, generate the output data flows identified above;
- (c) use the most appropriate mechanism(s) such as RDBMS, to retrieve data from the store identified above.

PROCESS SPECIFICATIONS

Page 471 of 475

8.4 Provide Transportation Planner Interface

Input Flows

ftp_evaluation_request
ftp_export_request
ftp_generation_request

ftp_import_request
ftp_output_request
ftp_parameters
ftp_request_documentation
ftp_request_link_data
ftp_simulation_request
ftp_static_data
ftp_traffic_data_request
ftp_update_link_data
link_database
transportation_planner_outputs

Output Flows

import_request
transportation_planner_inputs
ttp_evaluation_results
ttp_output_data_store
ttp_output_documentation
ttp_output_link_data
ttp_simulation_data

Description:

Overview: This process shall be responsible for the interface between the transportation planner and the other processes in the Plan System Deployment function. It shall enable the transportation planner to control the processes in the function, input data for use in the simulation, evaluation and generation of improvements to the ITS road and freeway network, obtain the results of the simulation and evaluation processes, plus network documentation, and import and export data. The process shall support inputs from the transportation planner 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.

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

- (a)'link_database'which contains data requested from a data store;
- (b)'transportation_planner_outputs'which is received as a result of output to another process.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input data flows listed above;
- (b) when inputs are received from the transportation planner, generate the appropriate outputs identified above;
- (c) send any data received as a result of (b) to the transportation planner using the output data flows identified above;
- (d) be responsible for the management of the data in the store of link mapping data, using the most appropriate mechanism(s) such as RDBMS, for storing the data.

PROCESS SPECIFICATIONS

Page 472 of 475

8.5 Provide Map Update Provider Deployment Interface

Input Flows

fmup_deployment_map_update
request_map_data_import
static_data_for_maps
transit_raw_map_data

Output Flows

imported_map_data
tmup_deployment_map_update_request
tmup_map_static_data

Description:

Overview: This process shall be responsible for providing the interface through which the Plan System Deployment function can exchange data with the map update provider. This exchange of data shall enable the process to send data about the road and highway network to the supplier for inclusion in the digitized map data that is provided to other function within ITS. The supplier shall provide the process with digitized map data for use in the evaluation and simulation of the

traffic conditions in the road and freeway network so that it can be continually updated to reflect the changing pattern of use..

Data Flows: All input data flows are unsolicited and all output flows are solicited with the exception of the following:

(a)'fmup-deployment_map_update', which is received as a result of output being sent to the map supplier terminator.

Functional Requirements: This process shall meet the following functional requirements:

- (a) continuously monitor for receipt of the input flows listed above;
- (b) when the map data import request input data flow is received, generate the map update request output identified above;
- (c) when the data is received from the map update supplier as a result of (b) send it to the process originating the request in the imported map data flow identified above;
- (d) when either the static data for maps or transit raw map input data flows is received, send the data they contain to the map update supplier in the static data output data flow identified above.

PROCESS SPECIFICATIONS

Page 473 of 475

9 Satisfy Implementation Requirements

Output Flows

Description:

Overview: This process represents the physical implementation of functions and communications links that are required by the architecture. It has no data flows or logical functions but is needed to meet the User Service Requirements (USR's).

Data Flows:

Functional Requirements: This process shall meet the following functional requirements: