SDAV: SmartData Model

1. High-level Object Representation

Objects in the vehicular scenario can be represented at a high-level with descriptive and motion data. The SmartData model described here focuses on the motion data described as motion vector. Below we describe the different motion models that can be used with the available data. On the following formulas the X and Y location refer to latitude and longitude, respectively, where the model of vehicle used in these formulas assume Z axis is omitted (assumed to be 0). We base our location representation in latitude, longitude, and altitude based on WSG84 coordinate system, commonly used by GPS devices embedded into autonomous vehicles. In this SmartData model, these coordinates are communicated using their respective SI Quantity, Radians. The usage of WSG84 is widespread throughout several Autonomous Vehicle simulators (e.g., CARLA, SUMO, and Artery) and standards (e.g., ETSI). Moreover, a common practice in several algorithms and models used in Autonomous vehicles omit the Z axis. Therefore, using coordinate systems such as ECEF when omitting Z axis (altitude=0) will impact X and Y calculations.

- **Constant Velocity**: This model describes motion through the xy plane, the norm of the velocity vector and the heading angle. Below are the state space and state transition equation:

$$
\begin{aligned}
x_{cv} =& (p_x \ p_y \ v \ \Psi)^T \\
x_{k+1} =& x_k + \begin{pmatrix} v\cos(\Psi)T_k \\ v\sin(\Psi)T_k \\ 0 \\ 0 \end{pmatrix}
\end{aligned}
$$

- **Constant Acceleration**: This model describes motion through the xy plane, the norm of the velocity vector, the heading angle and the longitudinal acceleration. Below are the state space and state transition equation:

$$
\begin{aligned}
x_{ca} =& (p_x \ p_y \ v \ \Psi \ a)^T \\
x_{k+1} =& x_k + \begin{pmatrix} (v_k T_k + \frac{a}{2}T_k^2)\cos(\Psi) \\ (v_k T_k + \frac{a}{2}T_k^2)\sin(\Psi) \\ aT_k \\ 0 \\ 0 \end{pmatrix}
\end{aligned}
$$

- **Constant Turn Rate and Velocity**: This model describes motion through the xy plane, the norm of the velocity vector, the heading angle and the yaw rate. Below are the state space and state transition equation:

$$
\begin{aligned}
x_{ctrv} =& (p_x \ p_y \ v \ \Psi \ w)^T \\
x_{k+1} =& x_k + \begin{pmatrix} \frac{v}{w}[\sin(\Psi_k + wT_k) - \sin(\Psi_k)] \\ \frac{v}{w}[\cos(\Psi_k) - \cos(\Psi_k + wT_k)] \\ 0 \\ wT_k \\ 0 \end{pmatrix}
\end{aligned}
$$

- **Constant Turn Rate and Acceleration**: This model describes motion through the xy plane, the

norm of the velocity vector, the heading angle, the yaw rate and the longitudinal acceleration. Below are the state space and state transition equation:

$$
\begin{aligned}
x_{ctra} =& (p_x \ p_y \ v \ \Psi \ w \ a)^T \\
x_{k+1} =& x_k + \begin{pmatrix} g_x(x_k, T_k) \\ g_y(x_k, T_k) \\ aT_k \\ wT_k \\ 0 \\ 0 \end{pmatrix} \\
g_x =& \frac{a[\cos(\Psi_k + wT_k) - \cos(\Psi_k)]}{w^2} \\
& + \frac{(v_k + aT_k)\sin(\Psi_k + wT_k) - v_k \sin(\Psi_k)}{w} \\
g_y =& \frac{a[\sin(\Psi_k + wT_k) - \sin(\Psi_k)]}{w^2} \\
& + \frac{(v_k + aT_k)\cos(\Psi_k + wT_k) - v_k \cos(\Psi_k)}{w}
\end{aligned}
$$

2. Mobile SmartData

The baseline feature for the Motion Vector is the Mobile version of SmartData, which comprises the id of the mobile objects (and static objects interacting with mobile ones, e.g., Road Side Units). For internal usage, the ID corresponds to a 64-bit ID given by object recognition and tracking modules. For V2X communication, the ID represents a 64-bit hash of the public certificate of the mobile object from which the data originated.

3. Dynamics Perception

A GPS sensor is modeled as a Multi-Value SmartData named. A Multi-Value SmartData represents a sensor that provides the interface for acquiring multiple data within the same space with the same unit. What differentiates a sample from another is either the time (e.g., a time series) or the disambiguation parameter. The Multi-Value SmartData for a GPS sensor comprises longitude, latitude, altitude coordinates in meters (m SI UNIT).

The IMU sensor is represented by the IMU Dynamics Multi-Unit SmartData composed of three units. A Multi-Unit SmartData represents a sensor that provides the interface for the collection of multiple data with different Units at the same time and space. It comprises data coming from accelerometers for longitudinal, lateral, and vertical accelerations (m/s2 SI UNIT), gyroscopes for yaw, pitch, and roll rate (velocity, m/s SI UNIT), and the orientation in the yaw, pitch, and roll axis (radians rad SI UNIT). For the sake of simplicity, By default, IMU Dynamics Multi-Unit SmartData comprises longitudinal and lateral accelerations, yaw rate, and yaw angle, derived from the gyroscope and accelerometers readings. However, the remaining information can be included in a customized version of the IMU Dynamics during setup of the simulation.

Other dynamics are combined as data gathered by the ECU, for instance, data from power inverters, battery accumulators, protection mechanisms, and engine sensors, wheel speed sensors. These dynamics are not set as a Multi-Unit SmartData. Instead, the ECU implements individual handlers for each other data meaningful for the current system not yet represented by the previously presented SmartData Sensors. Therefore, other SmartData are able to declare Interests to individual data handled by the ECU service.

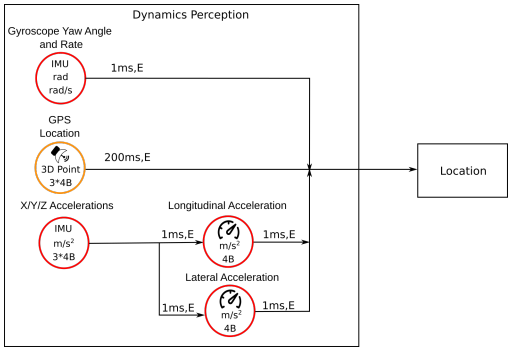

Figure 8: Dynamics Perception

Description	Unit	Desc	Semantics
IMU Acceleration	0xC4962924 (F32)	m/s²	Accelerometers' readings of the AV, longitudinal (dev=0), lateral (dev=1), vertical (dev=2).
IMU Rotation	0xC4B24924 (F32)	rad	Roll, Yaw, Pitch angle in radians (dev=0, 1, 2).
IMU Angular Velocity	0xC4B23924 (F32)	rad/s	Gyroscopes Roll, Yaw, Pitch rate in rad/s (dev=0, 1, 2).
GPS Location	0xC4B24924 (F32)	rad	The physical location of the object in meters decomposed in longitude (dev=3), latitude (dev=4)
GPS Location Alt	0xC4964924 (F32)	m	The altitude of the physical location of the object in meters (dev=0)

4. Location

Due to issues regarding the GPS update rate, and the known lack of precision of GPS in specific scenarios (e.g., tunnels), we include a localization step that is responsible for combining the previous State of the system with V2X, IMU, and ECU information to improve the Location accuracy (e.g., using a Kalman Filter).

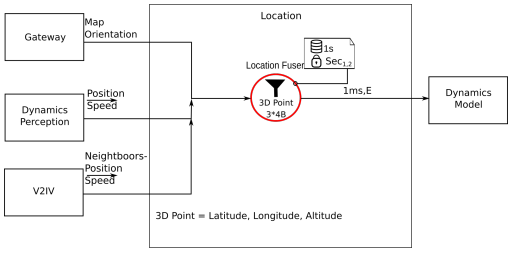

Figure 9: Location Estimation

Description	Unit	Desc	Semantics
Location Lon Lat	0xC4B24924 (F32)	rad	The estimated location of the AV comprising GPS, previous State, and IMU information, longitude (dev=5), latitude (dev=6)
Location Alt	0xC4964924 (F32)	m	The altitude of the physical location of the object in meters (dev=1)

5. Vision

The presented design comprehends the vision perception and comprises at least LiDARs, RADARs, and Cameras.

LiDARS and RADARS are sensors for detecting objects based on their effects on the reflection of radio or laser beams. RADARS are sensors that emit radio waves and measure the time it takes to bounce back from an object and the shift in the frequency of returned waves. LiDARS emit laser beams with a wavelength near-infrared range and measure the time it takes to bounce back to estimate distance. A RADAR has a fixed location, while a LiDAR constantly rotates to promote a 360º view of its surroundings. Data is organized as a three-axis grid of points called a point cloud. The data at every point is expected to correspond to the measured distance to the object (if any) in meters (m SI UNIT). Therefore, a Point Cloud SmartData is the points comprising the beam's grid (x, y, z coordinates) and the respective measured distance for the specific beam. The number of points in a Point Cloud is given by the sensor's field of view (FOV, from 1º to 360º), the acquisition frequency step (the distance between each horizontal data acquisition), and the number of vertical beans. For instance, multiple LiDARS and RADARS can be set in the vehicle to increase robustness.

The Camera sensor provides images from the vehicle's surroundings. The size of the images is relative to the number of pixels (height and width). Images can be compressed if the sensor supports compression. There can be any amount of cameras in the vehicle (e.g., collecting data from different angles). The SmartData representing the output of a camera is defined as an Image SmartData with a Digital UNIT encoding the data representation (e.g., JPEG or H261 according to RFC2435, and RFC4587, respectively). The Transformers SmartData interested in Image SmartData must issue Interest to all the Image UNITs it supports in order to establish a transparent integration.

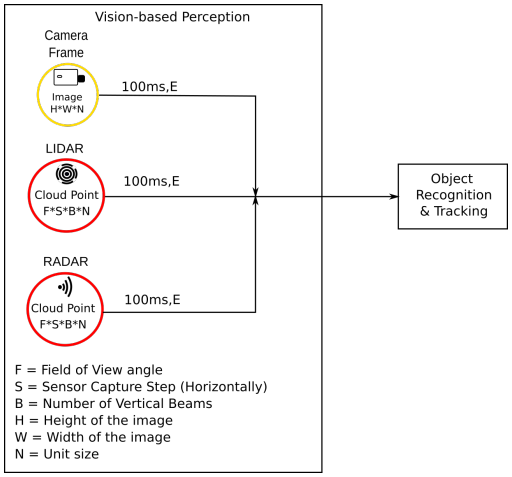

Figure 7: Vision

Description	Unit	Desc	Semantics
Point Cloud LIDAR	0x03010000+l	type=3, subtype=x, l=length of configuration	Device is given as a sequential counter of LIDARs in the vehicle
Point Cloud RADAR	0x03010000+l	type=3, subtype=x, l=length of configuration	Device is given as a sequential counter of RADARs in the vehicle.
Point Cloud LIDAR XYZ.RGB	0x03010000+l	type=3, subtype=1, l=length of configuration	Point Cloud File with X, Y, Z coordinates and intensity given as an RGB

Description	Unit	Desc	Semantics
Point Cloud LIDAR XYZ.I	0x03010000+l	type=3, subtype=2, l=length of configuration	Point Cloud File with X, Y, Z coordinates and intensity given as a single float between 0 and 1.
Image	0x02XX0000+l	type=2, subtype=Compressor type, l=maximum compressed image	Device is given as a sequential counter of Cameras in the vehicle.
Image PNG	0x02XX0000+l	type=2, subtype=36, l=1 (<1MB)	Device is given as a sequential counter of Cameras in the vehicle.
Image PNG	0x02XX0000+l	type=2, subtype=36, l=2 (1MB	Device is given as a sequential counter of Cameras in the vehicle.
Image PNG	0x02XX0000+l	type=2, subtype=36, l=3 (4MB	Device is given as a sequential counter of Cameras in the vehicle.

For example, If the LIDAR or RADAR is outputting a list of points with distance (e.g., given by 2 or 3 axis and one float for distance), one can derive max size from the LIDAR/RADAR characteristics such as Field of View * Step Size * Number of Beams * (Number of Axis in List * size of axis + 1 (distance) * size of distance).

6. MultiSmartData

The SmartData model presented below is based on MultiSmartData, a collection of SmartData that share common characteristics and is merged into a single structure to avoid the replication of metadata. For instance, merging data with the same origin (Multi-Unit SmartData) or same Unit (Multi-Value or Multi-Device SmartData). They are represented as Digital Units that meet the semantics behind the collection of multiple SmartData, given by the field `type` of a Digital Unit as follows:

Bit	31	30	16	0
	0	multi	type	length

7. Motion Vector SmartData Model

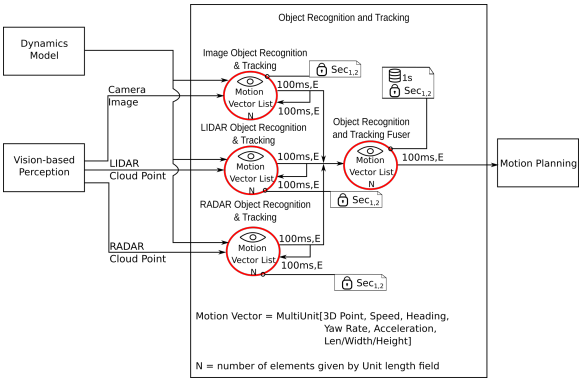

Figure 1: Object Recognition and Tracking Module

The output of Object Recognition algorithms and V2X messages of cooperative awareness are standardized to represent a Motion Vector. A Motion Vector is a Multi-Unit SmartData inspired by the Motion Vector defined by the European Telecommunications Standards Institute (ETSI) Cooperative Awareness Message (CAM) and Cooperative Perception Message (CPM). As it has been done previously for several Digital Units, we first split the `type` field of the Digital Unit into `type` and `subtype` in order to organize groups of similar Digital Units as subtypes of a general type:

Bit	**31 (SI)**	**30 (multi)**	**24**	**16**	**0**
	0	1	type	subtype	length

For the specific scenario of Motion vector, we reserve the subtype field (bits 16 to 23) for representing the classes of object we are describing. Furthermore, we reserve the types 0 and 1 of Multi SmartData for the Global and Local, versions of the Motion Vector (see section Global and Local Formats). Below we present the baseline units for Motion Vector, where N corresponds to the configuration type based on the number of elements inside the unit (N=1: a single Motion Vector, N>1: a list of Motion vector with N elements).

Bit	**31 (SI)**	**30 (multi)**	**24 (type)**	**16 (subtype)**	**(length)**
	0	1	0 = Motion_Vector_Global	class	N
	0	1	1 = Motion_Vector_Local	class	N

The class field corresponds to the class of the detected object/vehicle/RSU that sent the message. The class should be taken as a sequence of subclasses defined by ETSI SubClassTypes (page 108). The order should follow VehicleSubclassType, PersonSubclassType, AnimalSubclassType, and OtherSubClassType.

Finally, the confidence of such classification is given as the inverse of the uncertainty attribute of the SmartData, and the ID of the object described by the Motion Vector is given according to the origin ID in Mobile SmartData.

The Motion Vector Multi-Unit SmartData is composed of the motion data of the vehicle, including speed (m/s SI UNIT), heading (rad SI UNIT), Yaw rate (rad/s SI UNIT), longitudinal accelerations (m/s² SI UNIT). The semantics regarding location and heading are given in the next subsection. Note that, ID, time, and uncertainty are naturally tied to all Mobile SmartData. In the sense, in a Motion Vector MultiUnit SmartData, the uncertainty refers to the class associated with the object (defined at the unit). **The approximated dimension and mass are tied to the class associated to the object.**

7.1. Global and Local Formats

We split the Motion vector into two types to differentiate the local handling of data and data communicated via V2X and for Persistency. They differ in terms of reference points. On the one hand, the vehicle itself is assumed to be aware of its geographical location, for instance, given by a GPS according to WSG84 coordinate system. Therefore, its location is described by latitude, longitude, and altitude. In the Motion vector defined here, the location given by the GPS sensor shall be corrected to match the

center of its bounding box. Moreover, its heading is obtained as the angle in relation to the north coordinate defined by WSG84 (see Figure 2).

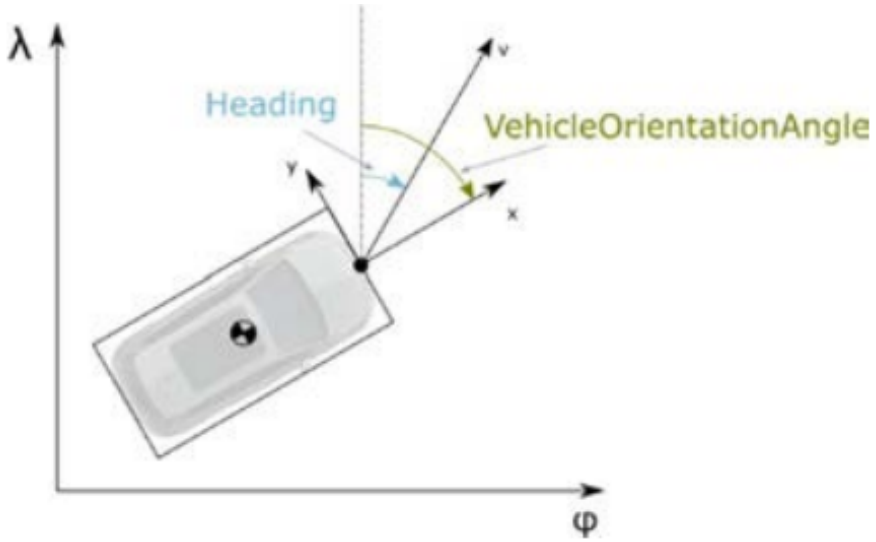

Figure 37: Vehicle Orientation Angle

Figure 2. Heading based on North from WSG84 coordinate system — Source: ETSI Documentation of CPM

On the other hand, objects detected through vision perception are commonly described locally in relation to the observer's Motion Vector. This format implies projecting the information to the location of the vehicle. In this sense, their ID corresponds to the Tracking system ID, and every other information in the Motion Vector (except dimensions) is given in relation to the vehicle's current dynamics (see the figure below).

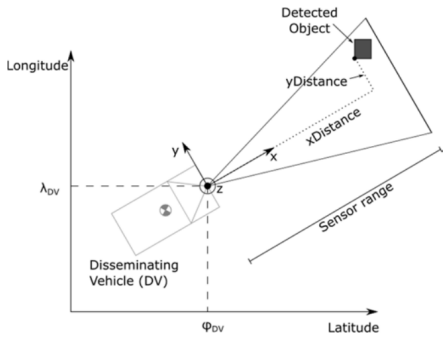

Figure 45: Coordinate System for detected object for vehicle as disseminating ITS-S

Figure 3: Perceived object from the perspective of the ego vehicle — Source: ETSI Documentation of CPM

7.2. Global Motion Vector

The Global format consists of data being communicated with other vehicles or cloud, and always carries absolute data. Here, location is represented by the SmartData attributes `x`, `y`, and `z`, in the respective ECEF format.

Description	Unit	Desc	Semantics
Speed	0xC4963924 (F32)	m/s	The speed of the object in meters per second (dev=0)
Heading	0xC4B24924 (F32)	rad	The heading of the speed vector of the object in relation to the true north (WGS84) in radians (values in degrees are expected to range between 0 and 2PI) (dev=7)

Description	Unit	Desc	Semantics
Yaw Rate	0xC4B23924 (F32)	rad/s	The yaw rate of the object in radians per second (dev=3)
Acceleration	0xC4962924 (F32)	m/s²	The acceleration of the object in meters per second squared (dev=3)

7.3. Local Motion Vector

The local format consists of data being generated by vision perception inside the vehicle. In this scenario, all data is derived as relative measures from the observers perspective. For instance, location is given as the distance to the observer's current location in meters. The notions of dimension and mass are tied to the class associated to the object.

More accurate information from the object can be received from trusted parties in V2X. The Motion Vector obtained via V2X is considered exact when a trusted vehicle sends information about itself, and is expected to be converted into a local motion vector before being submitted to the tracking sub-system. In this scenario, the ID is updated to match the ID of the vehicle (according to Mobile SmartData), as well as the samples of the other SmartData inside the Multi-Unit structure. Location in a Local Motion Vector is provide by the SmartData attributes for location, namely `x`, `y`, and `z`. Nonetheless, they carry the notion location of the object in meters relative to the observer's plane (e.g., 0,0,0 at position of current ego vehicle motion vector) decomposed in x, y, and z.

Description	Unit	Desc	Semantics
Speed	0xC4963924 (F32)	m/s	The relative speed of the object in meters per second (dev=1)
Heading	0xC4B24924 (F32)	rad	The relative heading of the object in radians (dev=8)
Yaw Rate	0xC4B23924 (F32)	rad/s	The relative yaw rate of the object in radians per second (dev=4)
Acceleration	0xC4962924 (F32)	m/s²	The relative acceleration of the object in meters per second squared (dev=4)

7.4. Deriving extra information

- From the speed and the difference between the heading of the vehicle and the heading of the road, one can derive longitudinal speed as "speed x cos(heading_diff)", and lateral speed as "speed x sin(heading)".
- Acceleration (m/s² SI UNIT) and yaw rate (when not present in the motion vector) can be derived by tracking the object over time.
- The Transport Theorem can be used to change an object perspective from the point of view of other vehicles (e.g., Motion Vectors being received via V2X) to the ego vehicle and vice-versa.

8. AV State

State Estimation (Figure 4) comprises the SmartData service responsible for gathering and deriving data that compose the notion of the State of the vehicle with information relevant to Path Planning, Motion

Planning, and Control, as an extension of the Motion Vector of the Vehicle. The data is expected to be improved in terms of accuracy during State Estimation by combining information from several dynamics in the system, similar to the Localization service. Data that is unavailable on the perception layer and can be derived from the remaining information, for instance, deriving speed and acceleration from last and current location, shall be done at the State Estimation service. The output of the State estimation is a Local MV Multi-Unit SmartData gathering information from IMU, ECU, and GPS. Its Unit is defined as:

During local processing, the AV State acts as the Motion Vector of the vehicle. Therefore, Local Motion Vector's present data relative to the current state of the vehicle.

Bit	31 (SI)	30 (multi)	24 (type)	16 (subtype)	(length)
	0	1	1 = Motion_Vector_Local	reference	1
	0	1	1 = Motion_Vector_Local	ego	1

And the Multi-Unit SmartData for the "reference" class comprises the following information:

Description	Unit	Desc	Semantics
Speed	0xC4963924 (F32)	m/s	The speed of the object in meters per second (dev=1)
Heading	0xC4B24924 (F32)	rad	The heading of the speed vector of the object in relation to the true north (WGS84) in radians (dev=8)
Yaw Rate	0xC4B23924 (F32)	rad/s	The yaw rate of the object in radians per second (dev=4)
Acceleration	0xC4962924 (F32)	m/s²	The acceleration of the object in meters per second squared (dev=4)

Location in a Local Motion Vector is provide by the SmartData attributes for location, namely `x`, `y`, and `z`. Nonetheless, for the "reference" class (ego reference position), x, y, and z shall be carried out as Longitude and Latitude in radians, and Altitude in meters (LLA). Once x, y, and z attributes are integers, LLA coordinates are treated as *float_in_radians*1000000000*.

A second motion vector, using "ego" as the class is expected to be made available. It follows the Local Motion Vector with all entries as 0 (e.g., 0 difference from reference). Nonetheless, the difference between the Heading (heading of the speed vector) and the vehicle orientation in regards to the WGS84 North is expected to be represented in Local Motion Vector with the class ego, thus, enabling the detection of skid.

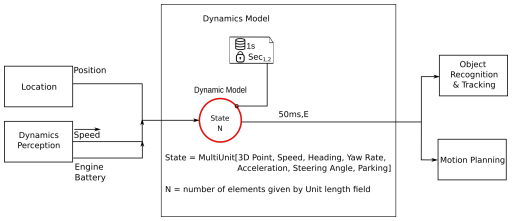

Figure 4: Dynamic Model

The Path Planning SmartData service declares Interest in Cached Map and Goal Destination, V2X, the updated list of Motion Vector from Objects Detected and Tracked, and the updated Vehicle Motion Vector (output of State Estimation). The path planning algorithm is expected to provide a valid path (e.g., without collisions) from the current position to the goal. This step implies behavior selection (e.g., selecting between an overtake, keeping on the lane, or changing lanes). Other steps may be included inside the path planning stage, such as predicting the movement of other non-stationary objects. A partial path (a.k.a., a segment) may be preferred to accommodate processing time. Therefore, the path is a variable-length list of future Waypoints (e.g., a list of future Local Motion Vector with the class "ego"), ignoring motion features. Motion Planning is the following SmartData service and, thus, is interested in the output of the Path Planning. Motion Planning is responsible for filling the motion information in the planned path, such as the respective set-points for actuation at each step and the time gap between each step. The granularity of the steps may be adapted to fit the complexity of the planned maneuver (e.g., removing equal commands in a straight path). Updates up to the motion planning are expected to be periodic. This decision envisions keeping the last motion plan output in check with reality, given the dynamicity of the driving task. Nevertheless, the temporal distance between the first element in the list of future States (motion plan output) and the last element dictates the maximum time gap to the next re-sampling. Thus, its expiry is equal to the segment length in time once all data shall be consumed until this point in time. Figure 5 presents the SmartData Design for the Planning module.

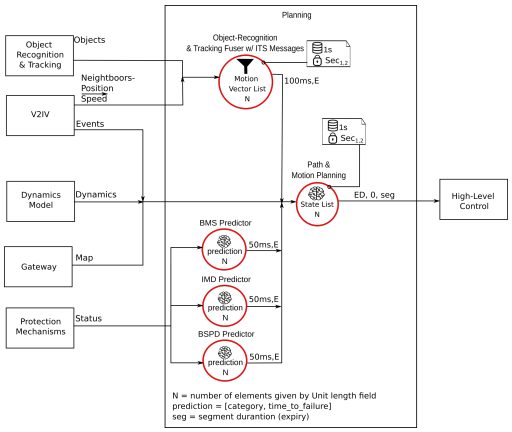

Figure 5: Planning

Therefore, the Multi-Unit SmartData comprising the future states of the vehicle is defined as a list of states with length=N>1:

Bit	31 (SI)	30 (multi)	24 (type)	16 (subtype)	(length)
	0	1	1 = Motion_Vector_Local	ego	N

9. High-Level Control

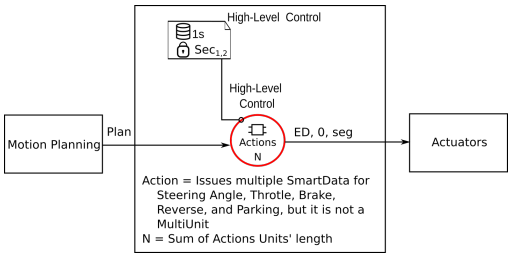

Figure 6: High-Level Control

Figure 6 presents the High-Level Control module. It is responsible for consuming the future states list outputted by the Motion Planning module. At each iteration of the consumption, the High-Level Control

issues the set points for each actuator. Therefore, it does not produces a Multi-Unit SmartData. Instead, it produces multiple SmartData for different targets. The High-Level Control actuators comprise Steering (rad SI UNIT), Throttle (m/s^2 SI UNIT), Brake (m/s^2 SI UNIT), Reverse (Digital UNIT for a Switch), and Parking (Digital UNIT for a Switch). These actuation set-points are forwarded to the simulation by the respective service. The simulator, or the respective target system, is expected to handle the low-level Control following the control rules set in place in the system, such as clutch and gear control (if necessary) or fuel injection. Figure 7 presents the main Actuator SmartData for the AV.

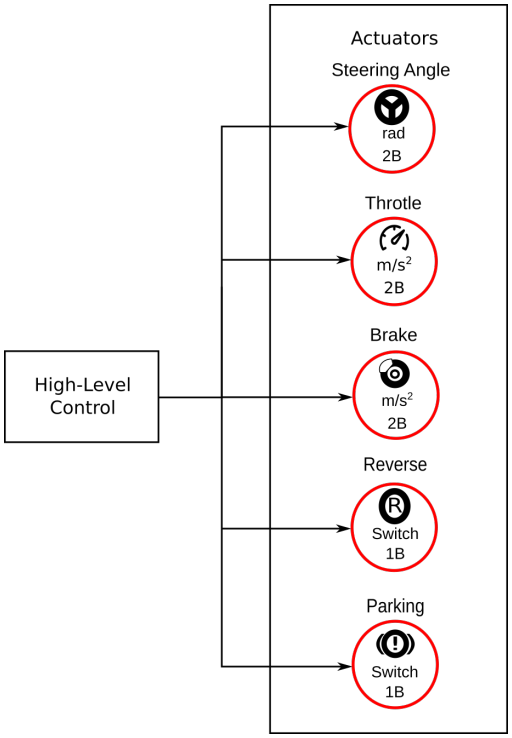

Figure 7: Actuation

Description	Unit	Desc	Semantics
Acceleration	0xC4962924 (F32)	m/s^2	Throttle set-point meters per second squared (dev=5) and Brake (dev=6)
Steering Angle	0xC4B24924 (F32)	rad	The steering angle of the ego vehicle in radians (dev=9)
Parking	0x00000001 (I32)	switch (Boolean)	The state (on/off) of the hand brake (dev=0)
Reverse	0x00000001 (I32)	switch (Boolean)	If the reverse gear is engaged (dev=1)

10. Dataset-Specific SmartData Model Information

- SmartData Model VeReMi Message Extension
- Sensing SmartData Model for AUDI's A2D2 Dataset
- SmartData Model OpenCOOD Dataset.