|  |  |
| --- | --- |
|  | Moving Picture, Audio and Data Coding by Artificial Intelligence  www.mpai.community |

|  |  |
| --- | --- |
| **Public Document** | |
| **N273** | 2021/06/07 |
| **Source** | Requirements (CAV) |
| **Title** | Initial MPAI-CAV Use Cases and Functional Requirements |
| **Target** | MPAI Members |

Contents

[1 Introduction 1](#_Toc73964706)

[2 The MPAI approach to standardisation 2](#_Toc73964707)

[3 Use Cases 3](#_Toc73964708)

[3.1 Human-to-CAV interaction 3](#_Toc73964709)

[3.1.1 Reference architecture 3](#_Toc73964710)

[3.1.2 Input and output data 4](#_Toc73964711)

[3.1.3 AI Modules 4](#_Toc73964712)

[3.2 Autonomous Motion 5](#_Toc73964713)

[3.2.1 Reference architecture 5](#_Toc73964714)

[3.2.2 Input and output data 6](#_Toc73964715)

[3.2.3 AI Modules 6](#_Toc73964716)

[3.3 CAV-to-environment interaction 12](#_Toc73964717)

[3.3.1 Reference architecture 12](#_Toc73964718)

[3.3.2 Input and output data 13](#_Toc73964719)

[3.3.3 AI Modules 14](#_Toc73964720)

[4 Technologies and Functional Requirements 14](#_Toc73964721)

[4.1 Introduction 14](#_Toc73964722)

[4.2 Human-CAV interaction 14](#_Toc73964723)

[4.3 Autonomous motion 14](#_Toc73964724)

[4.3.1 Summary of CAV Autonomous Motion data 14](#_Toc73964725)

[4.4 CAV-environment interaction 15](#_Toc73964726)

[5 References 15](#_Toc73964727)

[Annex 1 - Terminology 17](#_Toc73964728)

[Annex 2 – ETSI Technical Report 19](#_Toc73964729)

# Introduction

Moving Picture, Audio and Data Coding by Artificial Intelligence (MPAI) is an [international association](http://mpai.community/) with the mission to develop *AI-enabled data coding standards*. Research has shown that data coding with AI-based technologies is generally *more efficient* than with existing technol­ogies. Compression is a notable example of coding as is feature-based description.

The MPAI approach to developing AI data coding standards is based on the definition of *standard interfaces* of *AI Modules (AIM).* The Modules operate on input and output data with standard formats. AIMs can be *combined* and *executed* in an MPAI-specified *AI-Framework* according to the emerging MPAI-AIF standard being developed based on the responses to Call for MPAI-AIF Technologies.

By exposing standard interfaces, AIMs are able to operate in an MPAI AI Framework. However, their performance may differ depending on the technologies used to implement them. Therefore, MPAI believes that *competing* developers striving to provide more performing *proprietary* still *inter­operable* AIMs will naturally create *horizontal markets* of *AI solutions* that build on and further promote AI *innovation*.

This document, titled *Connected Automotive Vehicles* (MPAI-CAV), contains use cases for Human-CAV interaction, the “AI-based Performance Prediction” Use Case and associated Functional Requirem­ents. The MPAI-CUI standard uses AI substantially to extract the most relevant information from the indus­trial data, with the aim of assessing the performance of a company and predicting the risk of bankruptcy long before it may happen.

It should be noted that the AI-based Performance Prediction Use Case will be *non-normative*. The internals of the AIMs will also be *non-normative*. However, the input and output interfaces of the AIMs whose requirements have been derived to support the Use Cases will be *normative*.

This document includes this Introduction and

|  |  |
| --- | --- |
| Chapter 2 | outlines the MPAI approach to standardisation |
| Chapter 3 | describes the Use Cases giving for each the reference architecture, the input and output data and the AI Modules |
| Chapter 4 | provides the requirements for all technologies identified |

The Terms are defined in Annex 1.

# The MPAI approach to standardisation

MPAI standards target components and systems enabled by data coding technologies, especially, but not necessarily, using AI. MPAI subdivides an Implementation of an MPAI-specified Use Case into functional components called AI Modules (AIM). AIMs and AI systems implementing a Use Case are both called Implementations.

MPAI assumes Implementations use Artificial Intelligence (AI) or Machine Learning (ML) or traditional Data Processing (DP) or a combination of these. The implementation technologies can be hardware or software or mixed hardware and software.

An AI system implementing a Use Case is an aggregation of interconnec­ted AIMs executed inside an AI Frame­work (AIF). MPAI is developing and plans on releasing such an AI Framework (MPAI-AIF) standard in July 2021.

The 2 basic elements of the MPAI standardisation are represented in *Figure 1* and *Figure 2*.

|  |  |
| --- | --- |
| Diagram  Description automatically generated | Diagram  Description automatically generated |
| *Figure 1 – The MPAI AI Module (AIM)* | *Figure 2 – The MPAI AI Framework (AIF)* |

*Figure 1* shows a video coming from a camera shooting a human face. The function of this AIM (green block) is to detect the emotion on the face and the meaning of the sentence the human is uttering. The AIM can be implemented with a neural network or with DP technologies. In the latter case, the AIM accesses a knowledge base external to the AIM.

The MPAI approach to developing AI data coding standards is based on the definition of *standard interfaces* of *AI Modules (AIM)* *combined* and *executed* in an MPAI-specified *AI-Framework* (MPAI-AIF). AIMs operate on input data with standard formats and produce output data with standard formats. MPAI is silent on how an AIM produces output data from input data, with the constraint that an MPAI-standardised AIM must execute the normatively specified function.

By exposing standard interfaces, AIMs can interoperate in the MPAI AI Framework. However, their performance may differ depending on the technologies used to implement them.

MPAI believes that *competing* developers striving to provide more performing *proprietary* while still *interoperable* AIMs will naturally create *horizontal markets* of *AI solutions* that build on and further promote AI *innovation*.

Each Use Case normatively defines

Entity 1

The input/output data of the AIMs.

Entity 2

The function of each AIM

The input/output data of the AIM defined in point 1

Entity 3

The function of the Use Case

The input/output data to the AI system implementing the Use Case defined in point 1

The topology and the interconnections of the AIMs

A user of the standard *can* normatively reference one of the following three

Some data formats of Entity 1 only

Entity 1 and 2 only

Entity 1, Entity 2 and Entity 3 for each individual Use Case.

# Use Cases

The MPAI-CAV use cases relate to the 3 main subsystem ins a Connected Autonomous Vehicles. It develops and describes three Refer­ence Models:

*Human-to-CAV interaction*, i.e., the CAV subsystem that responds to hum­ans’ commands and queries, senses human activities in the CAV passengers compartment and activates other subsystems as required by humans or as deemed necessary by the identified conditions

*Autonomous Motion*, i.e., the CAV subsystem that enables its autonomous motion driving the CAV to the intended destination under instructions received through the Human-to-CAV interaction.

*CAV-to-environment interaction*, i.e., the CAV subsystem that communicates with sources of external information, including other CAVs, Roadside Units (RSU) etc..

## Human-to-CAV interaction

### Reference architecture

Humans and CAVs interact in several ways:

Human-CAV dialogue, e.g.,

Commands to Autonomous Motion Subsystem (possibly via radio), e.g.: Go to Pose p, Stop, Park, etc.

Other commands, e.g.: Turn off air conditioning, Turn on radio, Make a phone call, Open window or door, Search for information, etc.

Information requests, e.g.: How long does it take to destination?, What is the route conditions (e.g., traffic jam), What is the weather at destination?

Compartment monitoring, e.g.:

Physical conditions, e.g.: Temperature, There is no sound, There is anomalous noise, There is high-medium-low noise level, Media is playing, , , etc.

Passenger data, e.g.: # of passengers, ID of passengers, Estimated age of passengers, Destination of passengers.

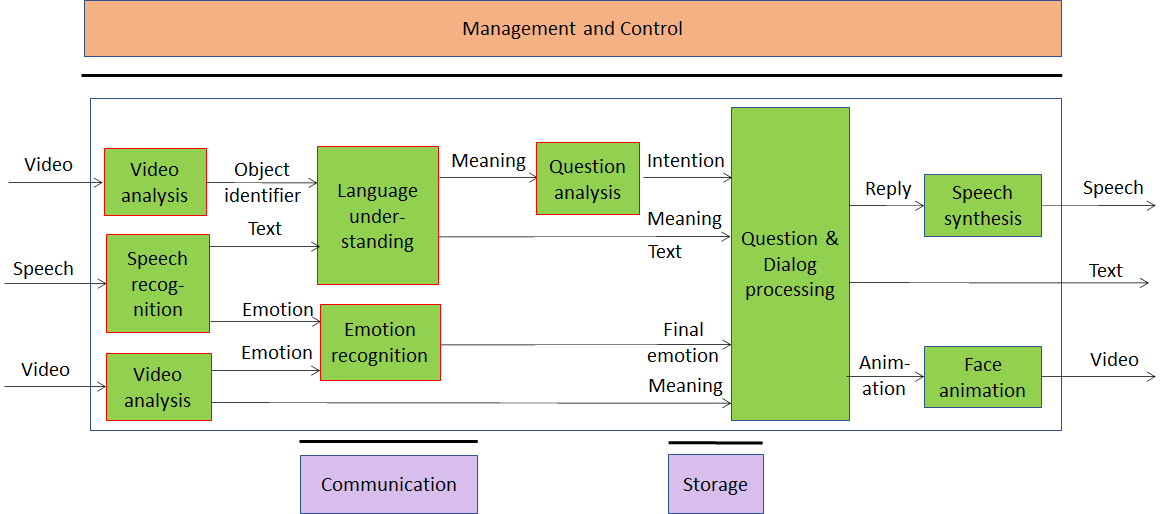
Passenger activity, e.g.: Level of activity of passengers (e.g., quiet at first but getting more talkative, or passengers do not talk), Level of passenger-generated sound, Level of passengers movement, Emotion of passengers (e.g. happy face, long face)

Passenger-to-passenger dialogue, e.g.: I feel unwell, You look nervous, Two passengers shake hands, passengers hold everyday conversation (e.g., soccer, movies, fashion, weather)

Passenger-to-web dialogue

The CAV collects data generated by humans inside the vehicle for possible action. In general, such data are anonymised, if meant for later, e.g., statistical use. Any data specific of a human shall be deleted at the end of the travel.

*Figure 3* is the reference model of Human-CAV interaction. A combination of Conversation with Emotion and Multimodal QA covers Human-CAV interaction needs.



*Figure 3 – Human-CAV interaction Reference Model*

Depending on the technology used (legacy or AI), the AIMs in *Figure 3* may need to access ex­ternal Know­ledge Bases to perform their functions.

### Input and output data

|  |  |  |
| --- | --- | --- |
| Input | Speech |  |
| Input | Video |  |
| Output | Synthetic speech |  |
| Output | Animated video |  |

### AI Modules

The AI Modules of Human-CAV interaction are given in *Table 1*.

*Table 1* *– AI Modules of* *Human-CAV interaction*

|  |  |
| --- | --- |
| **AIM** | **Function** |
| **Speech recognition** | Analyse the voice input and generate text output |
| **Video Analysis 1** | Produces the name of the object in focus |
| **Video Analysis 2** | Extracts emotion from human face |
| **Language understanding** | Analyses natural language expressed as text using a language model to produce the meaning of the text |
| **Emotion recognition** | Fuses Speech and Video emotions |
| **Question analysis** | Analyses the meaning of the sentence and determines the Intention |
| **Question & Dialog processing** | Analyses user’s neaning and/or question and produces a reply based on user’s Inten­tion |
| **Speech synthesis** | Converts input text to speech |
| **Question Answering** | Analyses user’s question and produces a reply based on user’s Inten­tion |
| **Intention KB** | Responds to queries using a question ontology to provide the features of the question |
| **Image KB** | Responds to Image analysis’s queries providing the object name in the image |
| **Online dictionary** | Allows Question Answering AIM to find answers to the question |

## Autonomous Motion

### Reference architecture

When properly instructed, the Autonomous Motion subsystem executes the instructions: go to a pose, change target pose and park. It does that by

using

quasi-static online information (offline maps)

information transmitted from roadside units

information from other CAVs

information from other vehicles and persons

information captured from

electromagnertic sensors (e.g., Lidar, Radar, Camera etc.)

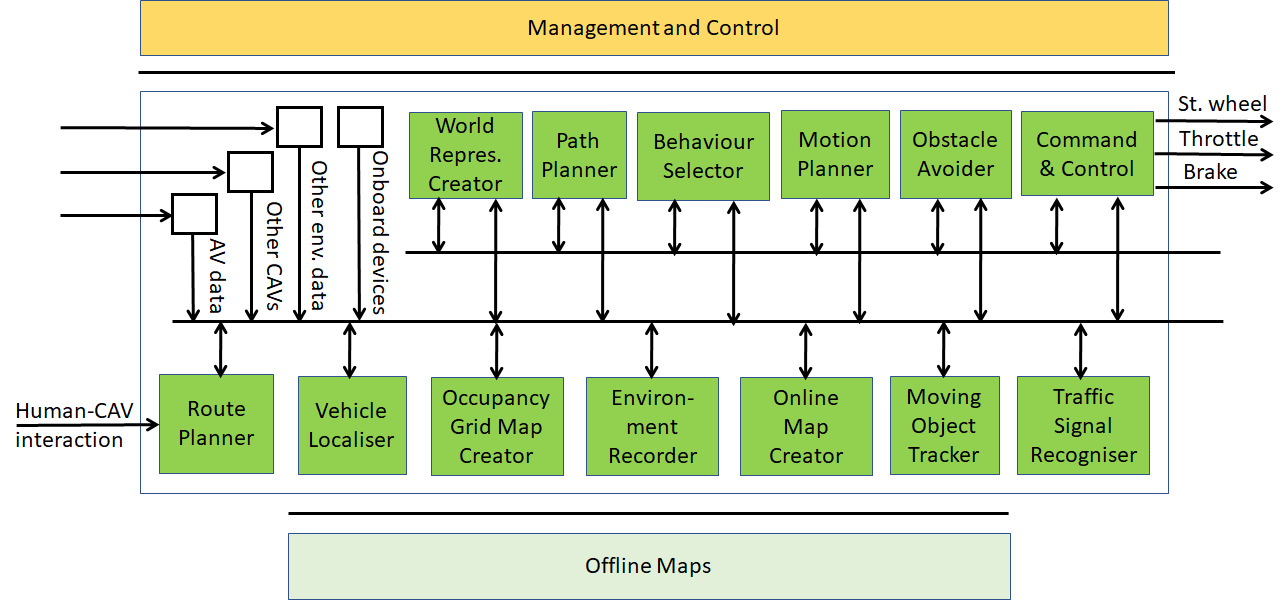
acoustic sensors

complying with traffic rules and regulations

considering safety and physical comfort of passengers.

The Autonomous Motion subsystem should be designed in such a way that different levels of autonomy, e.g., those indicated by SAE International [1], are possible depending on the amount and level of available functionalities.

The MPAI-CAV reference model is given by *Figure 4*.



*Figure 4 – MPAI-CAV Autonomous Motion Reference Model*

With the exception of Route Planner, the AIMs located at the bottom of *Figure 4* process typically high-speed data received from the physical environment or from devices inside the CAV (e.g., gyroscope). These signal sources are represented as white boxes. The AIMs at the top typically operate on the basis of processed lower-speed data received from the AIMs at the bottom. The order of the AIMs from left to right roughly corresponds to a sequential order in which AIMs take action after receiving an instruction.

### Input and output data

|  |  |
| --- | --- |
| Input | 1. Captured by sensors, e.g.,    1. Global Navigation Satellite System (GNSS)    2. Light Detection and Ranging (LIDAR).    3. Radio Detection and Ranging (RADAR).    4. Cameras (2/D and 3D).    5. Ultrasound.    6. Microphones.    7. Wheel encoder. |
|  | 1. Onboard devices    1. Inertial Measurement Unit (IMU)    2. Odometer, etc. |
|  | 1. Structured    1. Other CAVs    2. Static transmitters    3. Offline maps.    4. Entertainment. |
|  | User input |
| Output | Steering wheel actuation |
|  | Throttle actuator |
|  | Brake actuator |

Notes:

* 1. Road wheel-related sensors:
     1. tyre pressure (to be aware of CAV’s reaction to commands).
  2. Inertial Measurement Unit (IMU) contains
     1. Accelerometer
     2. Gyroscope
  3. Offline maps are created using satellite or onboard sensors data collected over multiple passes or or crowd-sourced to a fleet of cars, annotated and curated.

### AI Modules

The AI Modules of Autonomous Motion are given in *Table 2*.

*Table 2 – AI Modules of Autonomous Motion*

|  |  |
| --- | --- |
| **AIM** | **Function** |
| **Route Planner** | computes a Route 𝑊, through a road network, from the CAV’s Current State to the Final Goal. |
| **Vehicle Localiser** | estimates the current CAV State in the Offline Maps |
| **Occupancy Grid Map Creator** | represents the environment as a grid structure of binary values |
| **Environment Recorder** | processes and records a subset of data |
| **Online Map Creator** | creates a map with geometrical and topological properties |
| **Moving Objects Tracker** | detects and tracks position and velocity of moving obstacles in the environment comprising the CAV |
| **Traffic Signal Recogniser** | detects and recognises signs to enable the CAV to cor­rectly decide in conformance with the traffic rules |
| **World Representation Creator** | creates an internal representation of the environment |
| **Path Planner** | generates a set of Paths, considering 1) the current Route, 2) the CAV State, 3) the World Representation, and 4) the traffic rules |
| **Behaviour Selector** | to set a Goal to be reached with a Driving Behavior, avoiding collisions with static and moving objects within the decision horizon time frame |
| **Motion Planner** | define a Trajectory, from the current CAV State to the current Goal following the Behavior Selector’s Path as close as possible, satisfying CAV’s kinematic and dynamic constraints, and passengers’ comfort |
| **Obstacle Avoider** | defines a new Trajectory that avoids obstacles |
| **Command and Control** | makes the car execute the Trajectory as best as the environment allows |

#### Vehicle Localiser

1. Purpose: to estimate the current CAV State in the Offline Maps.
2. Input:

GNSSs (GPS/Galileo/Glonass/BeiDou)

Sensor data

Odometry

Data from other CAVs.

Offline Maps

Requests from other AIMs

1. Output:

State.

1. Notes:

Offline Maps are static maps of the environment that are:

Accessible online

Manually annotated, e.g., indicating pedestrian crosswalks or traffic light positions

Edited, i.e., without dynamic objects captured by the sensors.

Technology analysis.

GNSSs are the cheapest way to get the CAV pose. However, it not always reliable in cities (e.g., interference by tall trees, buildings, tunnels).

Sensor data

LIDAR sensors: offer measurement accuracy and processing ease

In LIDAR plus camera-based setups, LIDAR data are used to build the map, while camera data estimate the CAV’s position relative to the map

Camera-based localisation is cheap, but less precise and/or reliable.

Odometry

Odometry has precision problems because wheels slip and slide. Over time the distance is increasingly decorrelated from wheel rotation. More so when the CAV operates on non-smooth surfaces.

Data from other CAVs

by knowing the Pose and Path of another CAV (received from the CAV) and the distance from the CAV (computed), it is possible to know one’s pose

Data from offline maps

Used to place the CAV on the offline maps.

#### Route Planner

Purpose: computes a Route 𝑊, through a road network, from the CAV’s Current State to the Final Goal.

Input:

Current State.

Final Goal.

Output:

Route in the Offline Maps.

Time estimation.

Technology analysis.

Computing a Route can be reduced to finding the shortest path in a weighted directed graph if the road network is represented by a weighted directed graph where

vertices are the way points

edges connect pairs of way points

edge weights represent the cost of going from a way point to the another.

Techniques

Goal-directed.

Separator-based

Hierarchical

Bounded-hop

Combinations

#### Occupancy Grid Map Creator

Purpose: to represent the environment as a grid structure of binary values.

Input:

Sensor data.

State.

Output:

Occup­ancy Grid Map.

Technology analysis.

Regular spacing metric representation

An OGM is a space representation into discrete fixed size cells, of the order of cm.

An OGM cell contains the probability of occupation of the physical region it represents.

The occupancy probability of an OGM cell is assumed to be

Independent of the occupancy probability of other cells.

Gaussianly distributed around the cell

Represented as a Hilbert map

Represented as a DCT map

Cell occupancy probability is updated based on 3D sensor values.

For simplicity, 3D sensor measurements are projected onto the 2D ground plane.

Varied spacing metric representation

Octree-based Maps stores information with varied 3D resolutions.

Hybrid Maps store occupancy and distance measurements with varied resolutions yielding a representation with grid cells of increasing size from the center of the car.

#### Environment Recorder

Purpose: to process and record a subset of data

Input

Offline Maps

Occupancy Grid Map

State

Data from other CAVs.

Output

Stored data

Note

Candidate data for processing and storage are: point cloud of the environment, environ­ment data from other CAVs etc.

#### Online Map Creator

Purpose: to create a map with geometrical and topological properties.

Input:

Offline Maps

Occupancy Grid Map

State

Data from other CAVs.

Output:

Online Map.

Note:

Online Map should only contain a static representation of the environ­ment (desirable).

Pose, class and status of traffic signals

Horizontal: lane markings etc.

Vertical: speed limits, traffic lights, etc.

Traffic rules are typically embedded in road maps.

Technology analysis.

Metric representations

Grid maps map the environment into a matrix of fixed size cells

Topological representations

Topological maps hold more complex information, including multiple lanes, lane crossings, and lane mergers.

#### Moving Objects Tracker (MOT)

Purpose: to detect and track position and velocity of moving obstacles in the environment comprising the CAV.

Input:

Offline Maps

State

Data from other CAVs.

Output:

Pose and velocity of moving obstacles.

Note: moving obstacles can be

other vehicles

pedestrians

Technology analysis of MOT

Traditional is divided in: data segmentation, data association, and filtering

The positions of moving objects are estimated using ranging sensors, e.g., LIDAR and RADAR, or monocular/stereo cameras.

Objects with velocity above a given threshold are considered moving vehicles.

The size of the cube bounding box for each cluster is used to call vehicle a cluster.

Model-based uses physical models of sensors and geometric models of objects to directly infer from sensor data

Stereo vision based detects and tracks moving objects using color and depth infor­mation obtained from stereo pairs of images.

Grid map based builds an object-based description of the scene by segmentating, associating, and filtering the OGM of the dynamic environment

Sensor fusion fuses LIDAR-RADAR-camera data to explore their individual character­istics and improve environment perception. it is divided in two layers

The sensor layer extracts features from sensor data for potential description of a moving object hypothesis based on a point or box model, associates features with current hypotheses from the fusion layer

The fusion layer selects the best tracking model for each hypothesis and estimates (or updates the estimation of) the hypothesis state based on proposals from the sensor layer.

Deep learning uses deep neural networks to detect positions and geometries of moving objects, and to track their future states based on current camera data

#### Traffic Signal Recogniser

Purpose: to detect and recognise signs included in the traffic rules to enable the CAV to cor­rectly decide in conformance with the traffic rules.

Input:

Sensor data

State

Offline Maps

Online Maps

Data from other CAVs.

Output:

Traffic signals’ pose, class and status

Traffic rules.

Technology analysis

Traffic light detection and recognition

Detects the position of one or more traffic lights around the CAV and recognises their states (RGY).

Methods used are model-based and learning-based

Traffic sign detection and recognition

Detects the locations of traffic signs and recognises their categories (e.g., speed limit, stop, and yield sign.)

Model-based using simple features (e.g., colors, shapes, and edges)

Learning-based leveraging simple features, but evolving into more complex features (e.g., patterns, appearance, and templates)

Pavement marking detection and recognition

detects the positions of pavement marking and recognizing their types (e.g., lane markings, road markings, messages, and crosswalks)

CNNs are used to detect single or multiple traffic signs [10]

#### World Representation Creator

Purpose: to create an internal representation of the environment.

Input:

State

Pose, class and status of traffic signalisations

Pose of static obstacles

Pose and velocity of moving obstacles

Output:

Representation of the environment.

#### Path Planner

Purpose: to generate a set of Paths, = {𝑃1, 𝑃2, …,𝑃|𝑃 |}, considering 1) the current Route, 2) the CAV State, 3) the World Representation, and 4) the traffic rules. Paths extend some tens/hun­dreds of metres.

Input:

Route

State

Traffic Rules.

Output:

A set of Paths.

Technology analysis

Graph search based

Interpolating curves

Technology analysis

#### Behaviour Selector

Purpose: to set a Goal to be reached with a Driving Behavior, avoiding collisions with static and moving objects within the decision horizon time frame (ca. 5 s).

Input:

Pose.

Output:

Path.

Pose and velocity of moving obstacles.

Technology analysis

Finite State Machines

The Behavior is represented by states and transitions are based on discrete rules stemming from perception information. The states are drawn from several urban traffic scenarios.

Ontology

An ontology-based Knowledge Base is used to model traffic regulations and sensor data in order to help CAVs understand the world.

Markov decision processes

#### Motion Planner

Purpose: to define a Trajectory 𝑇, from the current CAV State to the current Goal following the Behavior Selector’s Path 𝑃𝑗 as close as possible, satisfying CAV’s kinematic and dynamic constraints, and passengers’ comfort.

Input:

Path.

Output:

Trajectory.

Technology analysis

A Trajectory may be defined as a sequence of

Commands, i.e. 𝑇𝑐 = {𝑐1, 𝑐2, …, 𝑐|𝑇 |}, where each command 𝑐𝑘 = (𝑣𝑘, 𝜑𝑘, 𝛥𝑡𝑘) and 𝑣𝑘 is the desired velocity at time 𝑘, 𝜑𝑘 is the desired steering angle at time 𝑘, and 𝛥𝑡𝑘 is the duration of 𝑐𝑘;

States 𝑇𝑠 = {𝑠1 , 𝑠2 , …, 𝑠|𝑇 |), where each state 𝑠𝑘 = (𝑝𝑘, 𝑡𝑘) and 𝑝𝑘 is a pose, and 𝑡𝑘 is the time in seconds after which 𝑝𝑘 is expected to be achieved.

Motion planning methods can be categorised into 4 classes: graph search, sampling, interpolating curve, and numerical optimization.

#### Obstacle Avoider

Purpose: define a new Trajectory that avoids obstacles.

Input:

Trajectory from Motion Planner.

Pose and velocity of moving obstacles.

Output:

Trajectory.

#### Command and Control

Purpose: to make the car execute the Trajectory as best as the environment allows.

Input:

Trajectory from Obstacle Avoider.

Output:

Effort commands to steering wheel actuator

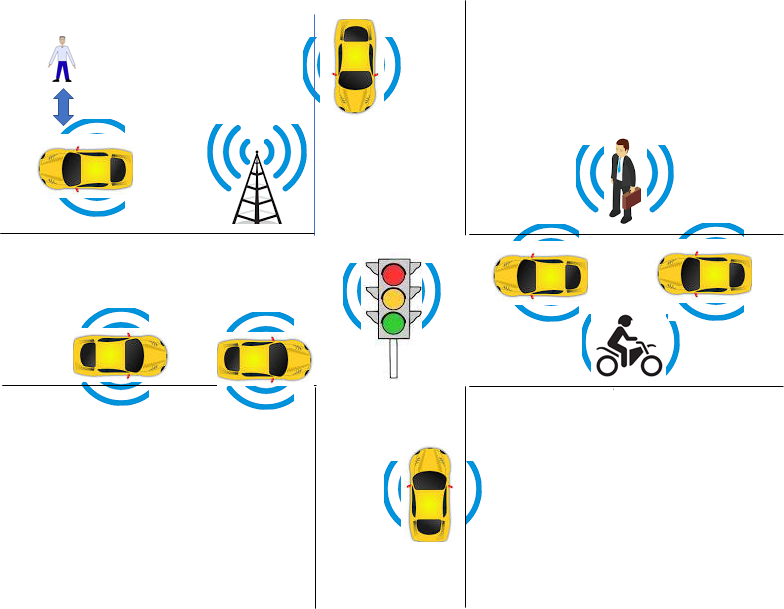
Effort commands to throttle actuator

Effort commands to brakes actuator.

## CAV-to-environment interaction

### Reference architecture

*Figure 5* depicts the environment applicable to MPAI-CAV.



*Figure 5 – The MPAI-CAV Environment*

CAVs can communicate via radio with other CAVs and other information sources.

CAVs can improve their perception capabilities by exchanging information about what they sense with other entities:

Neighbouring CAVs transmitting data in broadcast or unicast mode.

Other vehicles, such as electric scooters, bicycles.

Pedestrians.

Fixed equipment (e.g., traffic light, bus stop)

The following categories of vehicular communication are part of the literature or industry effort:

|  |  |  |
| --- | --- | --- |
| V2V | Vehicle-to-Vehicle | communication between vehicles to exchange information about the speed and position of surrounding vehicles |
| V2I | Vehicle-to-Infrastructure | communication between vehicles and road infrastructure. |
| V2X | Vehicle-to-Everything | communication between a vehicle and any entity that may affect, or may be affected by, the vehicle |
| V2R | Vehicle-to-Roadside | communication between a vehicle and Road Side Units (RSUs). |
| V2P | Vehicle-to-Pedestrian | communications between a vehicle and (multiple) pedestrian device(s) and to other vulnerable road users, e.g., cyclists, in close proximity |
| V2S | Vehicle-to-Sensors | communication between a vehicle and its sesnors on board |
| V2D | Vehicle-to-Device | communication between a vehicle and any electronic device that may be connected to the vehicle itself |
| V2G | Vehicle-to-Grid | communication with the power grid to sell demand response services by either returning electricity to the grid or by throttling their charging rate |
| V2N | Vehicle-to-Network | broadcast and unicast communications between vehicles and the V2X management system and also the V2X AS (Application Server) |
| V2C | Vehicle-to-Cloud | communication with data centers and other devices connected to the internet |

Technologies exist that support at least some aspects of the communivation types of the table:

Radio access, e.g., visible light communication, mmWave, Cellular-V2X, and 5G

Radio resource management (RRM) for vehicular communication using cellular technology

3GPP Release 14: air interfaces and core network technologies to support V2X communic­ation.

Vehicular ad hoc network (VANET)

Dedicated Short-Range Communication (DSRC): 5.9 GHz band with a range of ~300 metres.

Software defined vehicular networks (SDVN)

Internet of vehicles (IoV)

Protocol stack of the intelligent transportation system (ITS)

Cooperative Awareness Messages (CAMs) messages related to the status of CAV’s sent via wireless broadcast in VANETs.

Cooperative or collective perception improve CAV’s perception beyond the sensors’ detection range

Traffic situation can be extracted from Local dynamic map (LDM) that aggregates CAMs.

### Input and output data

#### CAVs within range

MPAI is developing a different payload as indicated in *Table 3*. This relies on a common world volumetric model. CAVs communicate in broadcast mode with other CAVs.

*Table 3 – MPAI-CAV Interaction with Environment data*

|  |  |  |
| --- | --- | --- |
|  | **Data type** | **Description** |
| V | CAV identity | Digital equivalent of plate number, including CAV model |
| V | Attitude-Path-Trajectory | See definitions |
| O | Spatial attributes | Position, velocity, acceleration, bounding box and semantics of objects in the environment |
| V | World representation | CAV’s world representation. (original or after fusion?) |
| V | Distance | Estimated distance between the CAV and all other CAVs. |
| E | Events | E.g., Works, Traffic jams, Number of cars at a traffic light etc. |

What is the size (MByte/GByte) of a lidar scan? say 17 Mpoint ~550 MB

difference maps

#### Other vehicles (not CAVs)

Other vehicles can be scooters, motorcycles, bicycles, other non-CAV vehicles.

They transmit their position as derived from GPS?

#### Pedestrians

Their smartphones can transmit their coordinates as available from GPS

#### Fixed equipment

Fixed equipment are traffic lights, bus stops, road side units.

**Traffic lights** can transmit

geographic coordinates

state (Green-Yellow-Red), time to change state

lane marking

speed limits

pedestrian crosswalks

**Road side transmitters** can transmit

Geographic coordinates

3D representation of environment

### AI Modules

# Technologies and Functional Requirements

## Introduction

The Functional Requirements refer to the individual technologies identified as necessary to implement Use Cases belonging to given MPAI-CAV application area using AIMs operating in an MPAI-AIF AI Framework. The Functional Requirements developed adhere to the following guidelines:

AIMs are defined to allow implementations by multiple technologies (AI, ML, DP)

DP-based AIMs need interfaces such as to a Know­ledge Base. AI-based AIM will typically require a learning process, however, support for this process is not included in the document. MPAI may develop further requirements covering that process in a future document.

AIMs can be aggregated in larger AIMs. As a consequence, some data flows of aggregated AIMs may not neces­sarily be exposed any longer.

## Human-Cav Interaction

## Autonomous Motion

### Summary of CAV Autonomous Motion data

The table gives, for each AIM (1st column), the input dats (2nd column) from the AIM (3rd column) and the output data (4th column).

*Table 4 – MPAI-CAV Autonomous Motion data*

|  |  |  |  |
| --- | --- | --- | --- |
| **CAV AIM** | **Input** | **From** | **Output** |
| **Route Planner** | State | Vehicle Localiser | Route  Estimated time |
| **Vehicle Localiser** | Sensor data | Input Data | State |
| Odometry | Onboard devices |
| Offline Maps | Input Data |
| Sensor Data | Other CAVs |
| Final Goal | User |
| **OGM Creator** | Various Data | Input Data | Occup­ancy Grid Map |
| **Environment Recorder** | State | Vehicle Localiser | -- |
| OGM | OGM Creator |
| Data (TBD) | Other CAVs |
| **Online Map Creator** | State | Vehicle Localiser | Online Map |
| Offline Maps | Input Data |
| Occup­ancy Grid Map | OGM Creator |
| Various Data | Other CAVs |
| **Traffic Signal. Detector** | State | Vehicle Localiser | Traffic signals  Traffic rules |
| Sensor data | Input Data |
| Offline Maps | Input Data |
| Online map | Mapper |
| Various Data | Other CAVs |
| **Moving Objects Tracker** | State | Vehicle Localiser | Moving objects’ poses and velocities |
| Online Map | Mapper |
| Various Data | Other CAVs |
| **World Representation Creator** | State | Vehicle Localis. | World Representation |
| Array of traffic signals | Traffic Signalis. Detector |
| Static object poses | OGM Creator |
| Moving object’s poses and velocities | Moving Objects Tracker |
| **Path Planner** | Route | Route Planner | Set of Paths |
| State | Input Data |
| Traffic Rules | Traffic Signalis. Detector |
| **Behaviour Selector** | Pose | Vehicle Localis. | Path |
| Pose & velocity of moving objects |
| **Motion planner** | Path | Behaviour Selector | Trajectory |
| **Obstacle Avoider** | Trajectory | Motion Planner | Trajectory |
| **Command and Control** | Trajectory | Obstacle Avoider | Actuation of steering weel  Actuation of throttle  Actuation of brakes |

### Environment sensor data

#### Global Navigation Satellite System (GNSS)

#### Light Detection and Ranging (LIDAR)

#### Radio Detection and Ranging (RADAR)

#### Cameras (2/D and 3D)

#### Ultrasound

#### Microphones

#### Wheel encoder

### Onboard device data

#### Odometer

#### Accelerometer

#### Road wheel sensor

### User input data

### Offline map

### State

### Goal

### Route

### Occupancy Grid Map

### Online map

### Traffic signals

### Traffic rules

### Pose

### Velocity

### World representation

### Path

### Trajectory

### Output data

#### Steering wheel actuation

#### Throttle actuation

#### Brake actuation

## CAV-environment interaction

### CAV identity

### Attitude-Path-Trajectory

### Spatial attributes

### World representation

### Distance

### Events

# References

SAE International Releases Updated Visual Chart for Its “Levels of Driving Automation” Standard for Self-Driving Vehicles, https://www.sae.org/news/press-room/2018/12/sae-international-releases-updated-visual-chart-for-its-%E2%80%9Clevels-of-driving-automation%E2%80%9D-standard-for-self-driving-vehicles

ISO 8855: "Road vehicles -- Vehicle dynamics and road-holding ability -- Vocabulary"

Rodolfo W. L. Coutinho and Azzedine Boukerche, Guidelines for the Design of Vehicular Cloud Infrastructures for Connected Autonomous Vehicles, IEEE Wireless Communications - August 2019

Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B. Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M. Paixão, Filipe Mutz, Lucas de Paula Veronese, Thiago Oliveira-Santos, Alberto F. De Souza; Self-driving cars: A survey; Expert Systems With Applications 165 (2021) 113816

D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, “Multi-column deep neural network for traffic sign classification,” Neural Netw., vol.32, pp.333–338, Aug. 2012

ETSI TR 103 562 V2.1.1 (2019-12), Analysis of the Collective Perception Service (CPS); Release 2.

CAR 2 CAR Communication Consortium, https://www.car-2-car.org/

Usman Ali Khan and Sang Sun Lee; Distance-Based Resource Allocation for Vehicle-to-Pedestrian Safety Communication; https://www.mdpi.com/2079-9292/9/10/1640/pdf

Gokulnath Thandavarayan, Miguel Sepulcre, and Javier Gozalvez; Generation of Cooperative Perception Messages for Connected and Automated Vehicles; IEEE Transactions on Vehicular Technology, Vol. 69, No. 12, December 2020

Pranav Kumar Singhab, Sunit Kumar Nandiac, Sukumar Nandi; A tutorial survey on vehicular communication state of the art, and future research directions; Vehicular Communications Volume 18, August 2019, 100164

https://phantom.ai/assets/uploads/PAI%20Renesas%20Partnership%20Announcement%20(1).pdf

# Annex 1 - Terminology

|  |  |  |
| --- | --- | --- |
| **Term** | **Acron.** | **Definition** |
| Advanced Driver Assistance System | ADAS | Electronic systems that assist drivers in driving and parking functions |
| Aggregate Programming | AP | (Paradigm) prescribes that each AIM M periodically and asynch­ronously evaluates a program P (the same for all devices) by per­forming the following steps  collects the non-expired message received from the neighbour AIMs and (possibly) other local data from M;  evaluates the program P;  sends messages to the neighbour AIMs and (possibly) perform other action local to M. |
| AI Framework | AIF |  |
| AI Module | AIM | A computational entity with a defined (and fair and ethical) purpose (local or networked, single or multi processor) that exposes a set of MPAI interfaces that can be implemented as HW signals, SW APIs, protocols).Whatever is inside an AIM is not relevant. It can be connectionless or connection oriented. |
| Collective Awareness | CA | Periodic exchange of status information between ITS-Ss (ETSI) |
| Collective Perception | CP | Sharing the perceived environment of an ITS-S based on perception sensors (ETSI) |
| Collective Perception Message | CPM | Enables a CAV to share information about detected objects with other CAVs (ETSI) |
| Collective Perception Service | CPS | Enables CAVs to share information about other road users and obstacles that were detected by its perception sensors (ETSI). |
| Cooperative Awareness Message | CAM | Messages exchanged in the ITS network between ITS-Ss to create and maintain awareness of each other and to support cooperative performance of vehicles using the road network (ETSI) |
| Command and Control | CAC | The AIM converting AOD’s decisions into actual commands and controls. |
| Communication |  | The infrastructure that connects the Components of an AIF and distributed AIMs |
| Component |  | An element of the AIF Reference Model |
| Connected and Autonomous Vehicle | CAV | A vehicle capable to reach an assigned target by planning a route and acting on the CAV after sensing and interpreting the environment and possibly exchanging information with other CAVs. |
| Computational Field | CF | A distributed data structure that associates a value to each AIM. Each value is stored in the corresponding AIM, which can therefore read it |
| Decision horizon |  | The estimated time between the current state and the 𝐺𝑜𝑎𝑙𝑔 |
| Driving behaviour |  | A collection of behaviours, such as lane keeping, intersection handling, traffic light handling, etc. |
| Execution |  | Component where AIM workflows are executed. It receives external inputs and produces the requested outputs both of which are application specific |
| Goal |  | 𝐺𝑜𝑎𝑙𝑔=(𝑝𝑔,𝑣𝑔) is the pair 𝑝𝑔 and associated velocity. |
| Inertial Measurement Unit | IMU | Inertial positioning devices such as accelerometer, gyroscope, odometer |
| Machine Learning | ML |  |
| Management and Control | MAC | Components that manages and controls the AIMs in the AIF, so that they execute in the correct order and at the time when they are needed |
| Occupancy Grid Map | OGM | A representation of the environment as evenly spaced grids of 1/0 (presence/absence) representing an obstacle at that location computed using sensor data and CAV’s State. |
| Offline Map |  | An offline-created map of a location with annotation |
| Online map |  | An online-created map Merging Offline Maps and the Occupancy Grid Map computed online using sensors’ data  and the current car’s State. |
| Path |  | 𝑃𝑗 = {𝑝1 , 𝑝2 , …, 𝑝|𝑃|} is a sequence of CAV Poses 𝑝𝑖 = (𝑥𝑖,𝑦𝑖,𝜃𝑖) in the Offline Maps. |
| Pose |  | 2D coordinates of the CAV in the Offline Maps with its orien­tation *p* = (𝑥,𝑦,𝜃) |
| Remission Grid Map |  | A grid map of reflectance intensity distribution of the environ­ment measured by a LIDAR scanner |
| Route |  | A sequence of Way Points |
| State |  | The set of: pose, linear and angular velocity, acceler­ation etc. characterising the CAV at a given time |
| Storage |  | A Component used, e.g., to store inputs and outputs of the indiv­idual AIMs, data from the AIM’s state and intermediary results, shared data among AIMs etc. |
| Trajectory |  | A sequence of commands 𝑐𝑘 = (𝑣𝑘, 𝜑𝑘, 𝛥𝑡𝑘), where 𝑣𝑘 is the des­ired velocity at time *t*𝑘, 𝜑𝑘 is the desired steering angle at *t*𝑘, and 𝛥𝑡𝑘 is the duration of 𝑐𝑘. Other definitions of Trajectory exist. |
| Way Point | WP | A point 𝑤𝑖 given as a coordinate pair (𝑥𝑖, 𝑦𝑖), in an Offline Map |

# Annex 2 – ETSI Technical Report

ETSI specifies the Collective Perception Service (CPS) in its Technical Report [6]. The CPS includes the format and generation rules of the Collective Perception Message (CPM).

The CPM message format is (H=header, C=container, M=mandatory, O=optional).

*Table 5 – ETSI Collective Perception Message format*

|  |  |  |  |
| --- | --- | --- | --- |
| PDU header | H | M | protocol version, message ID and Station ID. |
| Management | C | M | transmitter type (e.g., vehicle or RSU) and position. |
| Station Data | C | O | transmitter heading, velocity, or acceleration etc. |
| Sensor Information | C | O | transmitter (e.g., speed, heading, or acceleration)  capabilities of the vehicle’s sensors. |
| Perceived Object | C | O | detected objects (e.g., distance, speed and dimensions)  time at which the measurements were done.  A CPM can report up to 128 detected objects |
| Free Space Addendum | C | O | free space areas/volume within the sensor detection areas |

Every 0.1s a CPM is generated if one of the 3 conditions is satified

no CPM has been generated in the last 1s

a new object has been detected

since last CPM sending info about a previously detected object (it must have an ID)

the following attributes have changed:

Absolute position ΔP > 4 m

Absolute speed ΔV > 0.5 m/s

more than 1s has passed (ΔT > 1 s).

ETSI makes use of a common coordinate system. A vehicle can communicate its absolute coordinates roll, pitch and yaw (Attitude).

Different CPM generation rules have been investigated [9].