**A Functional Architecture for Autonomous Driving**

A functional architecture refers to a logical decomposition of the system into component and subcomponents as well as data flows between them. This decomposition is done without reference to the actual technical implementation of the architectural elements in terms of hardware and/or software.

Here we are interested in autonomous driving systems. In the simplest scenario, this can be thought as a cognitive driving intelligence layered on top of a basic vehicle platform [1].

**Functional Components**

A common segregation of the principal functional architectural components of an autonomous driving systems consists of the following modules:

* Perception
* Decision and Control
* Vehicle platform manipulation

The Perception component is responsible for perceiving the external environment/context in which the vehicle operates. The Decision and Control component handles the control of the vehicle motion in response to the external environment that is perceived. The Vehicle platform manipulation deals mostly with sensing and actuation of the vehicle with the intension of achieving a desired motion [1].

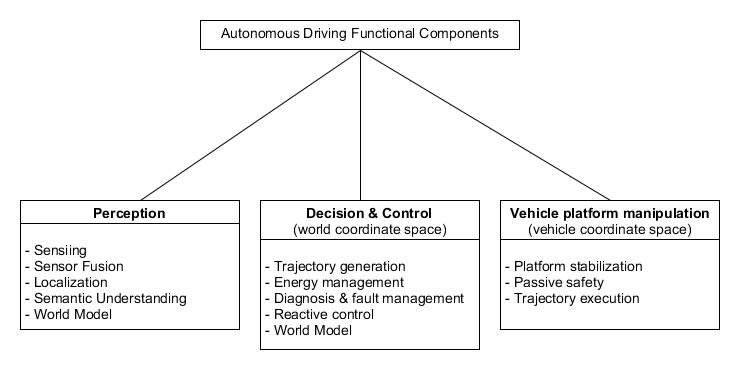


Figure 1 Autonomous driving functional components

Each of the aforementioned functional modules has several components which is shown schematically below

**Perception**

Sensing means gathering data on physical variables using sensors. Perception refers to the semantics i.e. interpretation and understanding, of the data in terms of high level concepts relevant to the task in hand.

The sensing components can be categorized into

* Those sensing the state of the vehicle
* Those sensing the state of the environment in which the vehicle operates

Another categorization of sensor components is from the viewpoint of systems integration. This categorization depends on the amount of processing needed to extract relevant information from the sensor data.

**Sensor Fusion**

The component considers multiple sources of information to construct a hypothesis about the state of the environment. Additionally, the component establishes confidence values for state variables . The component may also perform object association and tracking. Association refers to correlating pieces of information from multiple sensors to conclude that they refer to one and the same object.

For certain system configurations the sensor fusion block may also be used to eliminate some un-associated objects and data that is strongly likely to be superfluous or noise. This reduces the computation and communication load on subsequent components like the decision and control which need to work with the perceived data.

**Localization**

The localization component is responsible for determining the location of the vehicle with respect to a global map with needed accuracy.

Further, it may also aid the sensor fusion component to perform a task known as *map matching* wherein physical location s of detected objects are referenced to map’s coordinate system.

The component typically uses sensors like GPS and inertial measurement units (IMU).

Certain algorithms try to improve on the accuracy of localization by identifying visual landmarks via cameras. The base map layers have traditionally been stored on board however tiled maps can also be used. In the latter case tiles can be dynamically streamed from a service provider based on the vehicle location and may be locally cached.

**Semantic Understanding**

This component may include classifiers for detected objects and it may annotate the objects with references to physical models that predict possible future behavior. Detection of ground planes, road geometries, representation of driveable areas may also happen in this component.

In some cases, the semantic component may also use the ego vehicle data to continuously parametrize a model of the ego vehicle for purposes of motion control, error detection and potential degradation of functionality.

**World Model**

This component holds the state of the external environment as perceived by the ego vehicle

A world model component can be characterized as either passive or active [1]. The former is more like a data store and may lack semantic understanding of the stored data. Hence, by itself it cannot perform physics related computations on the data it holds. Therefore, it cannot actively predict the state of the world given specific inputs. The active world model may incorporate kinematic and dynamic models of the objects it contains and be able to evolve beliefs of the world states when it is given a sequence of inputs [1].

Other components (like Decision & Control) my request a set of predictions of future world states for a specific set of inputs in order to determine the optimal inputs to be applied.

Despite its inability to actively make calculations based on the data it contains, the passive world model is perhaps the most commonly found approach in autonomous driving projects [1].

The Local Dynamic Maps (LDM) is an approach to model a passive world model [1, 2]. An LDM is implemented as a database but conceptually it can be understood as a layered map.

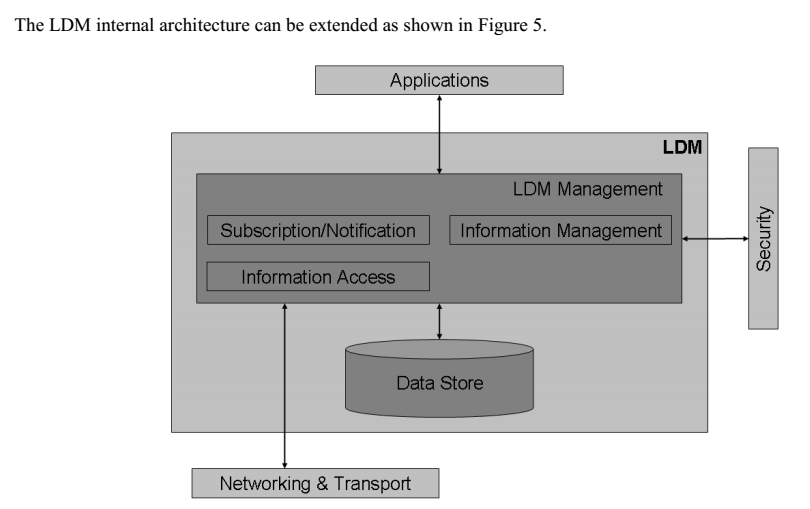


Figure 2 LDM internal architecture. Figure from [2].

The bottom most layers represent the most static beliefs about the world while the topmost layers represent the most dynamic in the sense of time. For example, the lowermost layer may be populated with a static map of the immediate surroundings of the vehicle (roads, permanent features etc.). The next layer may contain temporary objects like diversions due to construction work. The final layer would be populated by fast moving objects detected by the rest of the perception system (other vehicles, pedestrians etc.)[2].

Regardless of its implementation, the world model typically provides an interface to query its contents, add or remove data, concurrency, access control, replication over distributed computational media etc. It may also hold historical information about some or all of its contents.

**Decision & Control**

**Concept of Model Based Enterprise**

**Definition of Model Based Enterprise**

A model based enterprise is the environment in which MBD is the primary product definition or design authority

A model based enterprise (MBE) is an integrated and collaborative environment founded on a Model Based Definition that is shared across the enterprise, enabling rapid seamless and affordable deployment of products from concept to disposal.

A product or systems life cycle goes through the following stages:

* Concept
* Definition
* Materials and Manufacturing
* Use and Service
* End of Life

The model goes through this entire life cycle

Requirements = statements of need

**References**

1. Behere S., Torngren M. *A functional architecture for autonomous driving*
2. *ETSI TR 102 863 V1.1.1 Local Dynamic Map (LDM)-Rational for guidance on standardization*