Functional Architectures for Autonomous Vehicles Review

A. Giavaras

Contents

1 Introduction

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

Basic vehicles already run a larg amount of software. In many cases, this software gas to meet tight constraints with respect to real time processing, failure rate, maintainability and safety [1]. Adding cognitive intelligence into the mix leads to new software components deployed on existing platforms. Thus, a clear mapping between functional goals and software componenents is needed.

Software architecture has been introduced as a means to manage complexity in software systems and help assess functional and non-functional attributes before the build phase. A good architecture is known to help ensure tat a system satisfies key requirements in areas such as functional suitability, performance, reliability or interoperability [2].

Remark 1.1. **Functional Architecture**

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.

Furtheremore, the term is used analogously to the term *functional concept* described in the ISO 26262 automotive standard; a specification of intended functions and necessary interactions in order to achieve desire behaviors. Functional architecture design corresponds to the second step in the V-model. This software development life cycle is imposed by the mandatory compliance with the ISO 26262 automotive standard.

The transfer of total control from humans to machines is classified by SAE as a stepwise process on a scale from 0 to 5 [1]. The zero level involves no automation and five means full automation of all driving aspects under all roadway and environmental conditions. The classification above is meant to clarify the role of a human driver during vehicle operation. The first discriminant introduced is the environmental monitoring agent. In the case of no or partial automation, levels 0-2, the environment is monitored by the human driver whilst for higher degrees of automation, levels 3-5, the cognitive agent becomes responsible for environmental monitoring.

Another discriminant criteria is the responsibility for DDT fall-back mechanisms [1]. Again for low levels of automation, 0-3, the human driver needs to take control in case of a system fall while for levels 4 and 5 a human is no longer needed.

1.1 Terminology

In this section some important terms as defined in SAE J3016 are introduced:

- Dynamic Driving Task (DDT): real-time operational and tactical functions required to operate a vehicle, excluding strategic functions such as trip schedulin gor route planning.
- Driving automation system: hardware and software systems collectively capable of performing part or all of the DDT on a sustained basis. Driving automation systems are usually composed of design-specific functionality called features.
- Operation Design Domains (ODD): the specific conditions under which a given driving automation system or feature is designed to function.
- DDT feature: a design specific functionality at a specific level of driving automation with particular ODD

The SAE definition for DDT specifies for each class of components, see section 2, the functionality that must be automated in order to reach full autonomy, i.e. level 5, [1]:

- Lateral vehicle motion conrol via steering (operational)
- Longitudinal vehicle control via acceleration abd deceleration (operational)
- Monitoring of the driving environment via object and event detection, recognition, classification and response preparation (operational and tactical)
- Object and event response execution (operational and tactical)
- Maneuver planning (tactical)
- Enhanced conspicuity via lighting, signaling ang gesturing (tactical)

2 Functional Architecture

The process of functional architecture design starts by developing a list of functional components and their dependencies. According to the SAE J3016 standard there exist three classes of componenents

- Operational: basic vehicle control
- Tactical: planning and execution for event or object avoidance and expedited route following
- Strategic: destination and general route planning

Each of these components has an incremental role in a hierarchical control structure. This role starts from low level control through the operational class and

finished with a high level overview through the strategic class of components. In between the tactical components handle trajectory palnning and response to traffic events.

The automation of a task resmbles a control loop which receives input from sensors, performs some reasoning and controls the vehicle behavior through actuators [1]. This control loop is similar to the SENSE-PLAN-ACT cycle popular in robotics, see for example [6]. This should not pose a surprise as architecture design fro autonomous vehicles is analogous to the design of a real-time, intelligent, control system. Indeed the robotics and artificial intelligence literature has put forward quite a few reference architectures for such systems.

The architecture for an autonomous vehicle must be capable to represent both reactive and deliberative components in a single artifact so that it can both plan for the future and react in the least of time to unexpected events.

2.1 NIST Real Time Control Systems Architecture

An architecture paradigm that bridges the gap between reactive and deliberative components is the NIST Real Time Control Systems (RCS) reference architecture proposed by Albus.

The architecture does not separate components based on memory but instead it builds a hierarchy based on semantic knowledge. Thus, the components lower in the hierarchy have limited semantic understanding and can generate inputs for higher components. This makes RCS to map better to SAE classification of functional components, see section 3. Furthermore, the architecture allows both for static and dynamic information.

At the heart of the RCS control loop is a representation of the external world, the so calle world model, which provides data for sensor fusion, acts as a buffer between perception and behavior and supports seonsor data processing and behavior generation.

Although Albus proposed the design using a hierarchical control structure, where the perception at lower level nodes generates inputs for highter level nodes therefore increasing the level of abstraction. However, in the context of automotive, it is not always required for outputs from lower nodes to be inputs for higher nodes; it may well be the case that nodes at different hierarchical levels process the same sensor data.

Moreover, low level control loops such as longitudinal or lateral control, which need basic sensor input to execute their tasks need noworld modeling capacity. Thus, typically these components can be reduced to flat, simple control loops with multiple processing steps.

Remark 2.1. **Pipe-and-Filter**

From an architectural point of view, a flat stream of data which passes through different, individual, processing steps can be represented using the pipe-and-filter pattern, see [1]. This paradigm, divides a process into several sequential steps connected by the data flow; the output data of a step is the input to the subsequent step. Each processing step is implemented by a filter component.

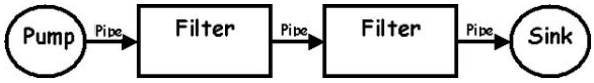

Fig. 1: Schematics of Pipe-and-Filter pattern.

Remark 2.2. **Tee-and-Join Pipeline**

The pipe-and-filetr pattern can only represent sequential processes. For hierarchical structures a variant of the pattern is the tee-and-join pipeline. In this paradigm, the input is passed to either to a low level pipeline, corresponding to a low level control loop or to a higher level one or both.

A component-based architecture can be an alternative to the paradigms remarked above. This style specifies that all components can be interchangeable and independent of each other. While it is popular for functional software architecture description it reveals no information about functional hierarchies. The layered architectural style typically limits component communication because the commands have to flow strictly from the higher layer to the lower one. Thus, it forces to group all decisions at a given layer.

3 Functional Components

This section discusses functional components that are typically required in an autonomous vehicle. The following figure shows the functional components that satisfy SAE J3016 requirements for fully autonomous vehicles. The data flows from left to right simulating a closed control loop. The figure shows three types of entities:

- Functional components
- Classes of components
- Subclasses of components

We have the following map:

- Vehicle control and actuators interface classes correspond to SAE operational functions
- Planning class corresponds to SAE tactical functions
- Behavior generation class maps to both strategic and planning class of functions

A common segregation of the principal functional architectural components of an autonomous driving systems consists of the following modules [3]:

- 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 [3]. Each of the aforementioned functional modules has several components which is shown schematically in figure 2.

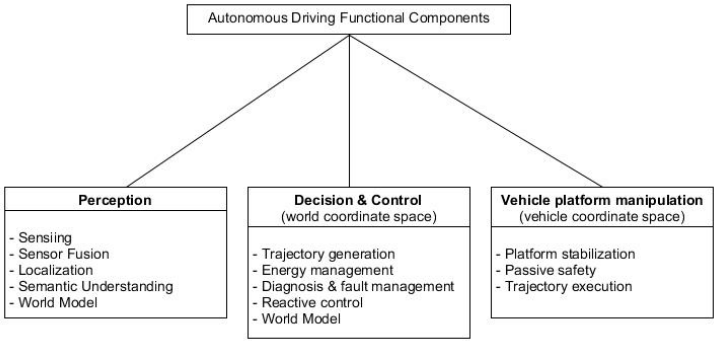

Fig. 2: Autonomous driving functional components.

4 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.

A first categorization of the sensing components can be: [3, 1]

- Those sensing the state of the vehicle

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

Enviromental monitoring can be based on RADAR, LIDAR and camera technologies [1].

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.

4.1 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.

4.2 Localization

The localization component is responsible for determining the location of the vehicle with respect to a global map with needed accuracy (a concept often called global localization). GPS technology is frequently employed for this task. 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 maps 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.

4.3 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 be-

havior. 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.

4.4 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 [3]. 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 [3].

Other components (like Decision and 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 [3].

4.4.1 Local Dynamic Maps

The Local Dynamic Maps (LDM) is an approach to model a passive world model [3, 4]. An LDM is implemented as a database but conceptually it can be understood as a layered map.

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.)[4].

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.

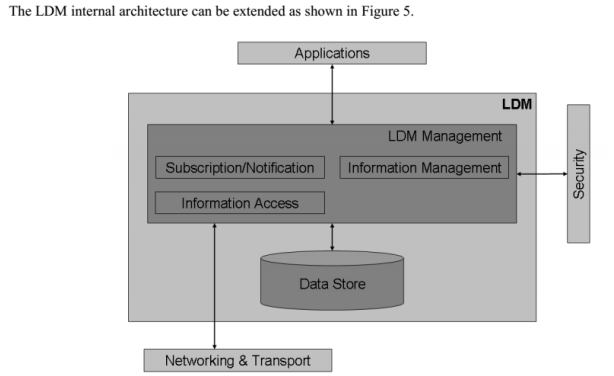

Fig. 3: LDM internal architecture.

5 Decision & Control

The functional components in this category are concerned with the vehicle behavior in the context of the external environment it is operating in [3]. Typically, modules in this category refer to the vehicle as a whole and the way it moves in its environment. Furthermore, energy, fault management concerns and reactive control to unexpected events in the environment are also handled.

5.1 Trajectory Generation

The component continuously generates obstacle free trajectories in the world coordinate system. The trajectory generation is constrained by factors such as energy availability, limitations of the vehicle platform (e.g. non-holonomicity), faults and failures.

5.2 Energy Management

5.3 Diagnosis and fault Management

5.4 Reactive Control

These components are used for immediate responses to unanticipated stimuli from the environment. Existing vehicle features like collision mitigation by braking may be considered as reactive control. These components execute in parallel with the nominal system, and if a threat is identified, their output overrides the nominal behavior requests. Their SENSE-PLAN-ACT loops typically run at least an order of magnitude faster than the nominal system loop [3].

It is sometimes the case that what is considered reactive behavior in the presence of unexpected events, can be dealt with by very fast deliberative behavior. As an example consider the Autonomous Emergency Breaking (AEB) that some passenger cars feature. This is considered a reactive function that monitors a small subset of sensors (compared to full autonomous driving) and initiates braking action in case of imminent collision with a moving or stationary obstacle. The function is constantly active, when it is enabled, and may generate a deceleration demand that overrides other demands on the propulsion subsystem. However, if the perception and trajectory generation components are sufficiently fast, they could detect the threat and generate appropriate trajectories as part of their normal operation and thus negating the need for a specialized AEB system [3].

6 Vehicle Platform Manipulation

This category groups the components that are directly responsible for the motion of the vehicle. They abstract the principal actuation systems and also provide minimum level of stability to the platform while it is in motion.

Although not directly related to propulsion, components related to passive vehicle safety may be included in this category, since they are closely related to scenarios arising from undesirable propulsion and may be triggered by the decision and control components.

6.1 Platform Stabilization

Usually components in this category are related to traction control, electronic stability and anti-lock braking features. The task of these components is to keep the vehicle platform in a controllable state during operation. Unreasonable motion requests may be rejected or adapted to stay within the physical capabilities and safety envelope of the vehicle.

6.2 Trajectory Execution

These components are responsible for executing the trajectory generated by the Decision and Control modules. This is achieved by a combination of longitudinal acceleration (propulsion), lateral acceleration (steering) and deceleration (braking). Since most vehicles already incorporate such components, these can be considered as traditional from the perspective of autonomous driving development.

7 Functional Distribution

We can consider an autonomous vehicle as comprising of two layers; a cognitive layer responsible for intelligently operating the vehicle and the vehicle platform which from the outside it corresponds to the vehicle chassis.

Naturally, this separation leads to the following two questions:

- What kind of information should flow between the cognitive and the vehicle platform layers?
- What changes are necessary if the vehicle platform is to be controlled by a machine and not a human being?

The answers to the previous questions depend on the distribution of functionality attributed to each layer. Currently, most autonomous vehicles are built on top of existing production vehicles therefore the pattern followed is: The vehicle platform contains a network of ECUs that control the functionality such as longitudinal and lateral dynamics or breaking. The manufacturer then allows for some sort of gateway that in return allow for a limited number of set points to be transmitted to the vehicle ECUs. As an example, the cognitive driving intelligence may continuously regulate the set-point of the cruise control function in the vehicle.

From a theoretical perspective, the functionality between the driving intelligence and the vehicle platform can be allocated such that it lies between two extreme cases. The one extreme is when the cognitive agent directly controls the torque outputs of the vehicle platform actuators using a distributed I/O approach. In this case the cognitive agent needs an intimate familiarity with the vehicle platform. Thus, this extreme implies a possibly large degree of coupling between the two layers. The other extreme, Figure 2 (b), treats both the cognitive driving intelligence as well as the vehicle platform as two cooperating, relatively autonomous entities. Neither knows the intimate details about the other. The driving agent makes motion demands of the vehicle platform in world coordinates, which the latter makes a best effort to fulfill. The task of the driving intelligence is to perceive the world and make motion requests in this world, while the task of the vehicle platform is to realize the desired motion requests while keeping its own features and limitations in mind. In such an ideal de-coupling, the same driving intelligence should be able to operate a variety of vehicle platforms, provided the acceleration interface remains the same.

We have already mentioned that the first architecture, introduces strong coupling between the two layers. Furthermore, in order to perform closed loop propulsion control of the vehicle platform, the driving intelligence would need a fairly detailed model of the platform, including its dynamics and the constraints on the vehicle actuators and sensors. Performing fine-grained (low time horizon) control of the actuators by using motion feedback from the perception system places unreasonably high demands on the technical implementation and

performance of the perception system. On the other hand, the latter architecture is attractive because it enables a relatively clean separation of concerns. The driving intelligence need not be concerned with the finer details of how the motion it desires is achieved. The vehicle platform need not be concerned with how and why the motion commands are generated - only whether they are realizable and if so, how to best realize them given the current platform capabilities. Concepts related to stabilization of the platform, like traction control, anti-lock brakes etc. are transparently realized by the vehicle platform, without the driving intelligence having to be aware of them.

Following the best practices of separating concerns and loose couplings, one should strive to achieve as clean a split as possible, between the driving intelligence and vehicle platform. This lowers the cognitive complexity (cognitive effort needed to understand a model) of the architecture, as well as reduces the potential for feature interaction and other undesirable emergent behavior. It also enables better reuse of the driving intelligence and vehicle platform in other projects. That said, any assumptions made regarding the behavior and performance of the vehicle platform need to be made explicit. This is especially true for end-to-end latencies on the fulfillment of acceleration requests, and interpretation of sensor data by the controllers in the vehicle platform. The approach (of Figure 2 (b)) places high demands on the functionality available in the vehicle platform, with regards to its abilities for keeping the platform stable, and retaining basic self-protection measures which may include reactive control. In practice, this is unlikely to be an issue because the high-end vehicles of most automotive OEMs today already incorporate such functionality and it is these high-end vehicles that are the most likely candidates for receiving upgrades to self-driving functionality. The principal modifications needed to these vehicles, as self-driving vehicle platforms, would be in the area of sub-system redundancy, to increase the platform reliability and safety.

8 Functional Architecture (An example)

As shown in Figure 3, the sensing and world model components, although conceptually unified, are split into those dealing with the external environment of the vehicle, and those dealing with the Ego vehicle platform. The split helps to achieve separate technical implementations, if required, when the functional architecture is eventually refined to a technical architecture, following the ISO26262 process. The inter-component arrows in Figure 3 represent data-flows in the direction of the arrow. As shown, the outputs of the sensing components go to the rest of the perception and decision and control components, either directly or indirectly, depending on the level of processing and fusion that is needed.

In our experience, it is useful to establish a data link between localization and sensor fusion. Certain sensors may exhibit repeatable tendencies at fixed loca-

tions along specific routes, like increase in false positives, dropouts etc. Changing the level of confidence in a sensor, based on geographical location is an interesting line of research and the architecture should not be a limiting factor. Another interesting data link in Figure 3 is the connection from the semantic understanding component to the sensor components. This is useful in at least three scenarios:

- Firstly, some specialized autonomous driving situations benefit from so-called focused attention mechanisms. Focused attention means exploring a specific part of the environment more deeply. This may require physical motion of the sensors and/or configuration changes to the sensors (panning a camera to a different field of view, changing the zoom of a lens, etc.). Today, most sensors of most autonomous vehicles are physically fixed to a constant pose with respect to the vehicle coordinate system. But in the domain of mobile and cognitive robotics, it is quite common to have, for example, a pan-tilt-zoom camera to aid the robot in a search task.
- Secondly, calibration changes to the sensors may be needed at runtime (e.g. changing exposures based on time of day, triggering re-calibration if changes in physical alignment are suspected).
- Thirdly, if communication transceivers are considered as a kind of sensor/actuator, the semantic understanding component can use it to respond to incoming communication requests, publish ego vehicle information and make asynchronous requests for information. Such communication requirements are often an integral part of scenarios like cooperative driving, where a vehicle maintains constant communication to the infrastructure and other vehicles in the vicinity.

References

[1] Serban A. C., Poll E., Visser J. *A standard driven software architecture for fully autonomous vehicles* https://pdfs.semanticscholar.org/176d/d7b3289267d8c693fb16f0d6fea480b5fad5.pdf

[2] Garlan D. *Software architecture: a roadmap* ICSE 00, pp. 91-101, ACM. 2000.

[3] Bahere S., Torngren M. *A functional architecture for autonomous driving* ICSE 00, pp. 91-101, ACM. 2000.

[4] ETSI TR 102 *Local Dynamic Map (LDM)-Rational for guidance on standardization.* ETSI TR 102 863 V1.1.1

[5] Rahul Sharma K., Daniel Honc, Frantiek Duek *Predictive Control of Differential Drive Mobile Robot Considering Dynamics and Kinematics* https://pdfs.semanticscholar.org/176d/d7b3289267d8c693fb16f0d6fea480b5fad5.pdf

[6] Murphy R. R. *Introduction to AI robotics* MIT Press, 2000.