Laboration 2: Software Architecture Design and Documentation

Group 6:

Naida Demirovic ndc17001

Nermin Imamovic nic17001

Lab 2.1 Stakeholders

Stakeholders:

1. Software developers (employees)
2. Company (investors)
3. External companies that are developing other devices (partners)
4. Driver (user)
5. Government

Stakeholders and views:

Software developers use *implementation view* because of having the capture of the architectural decisions made for implementation. Software developers use *concurrency view* to have a concurrency structure of the system which represents parts of system that can be used concurrently in how it is coordinated and controlled.

Company uses *work assignment view* to have clear representation of all actions related to creating phases of system. Company uses *decomposition view* to have a clear detail view to all the parts of the system.

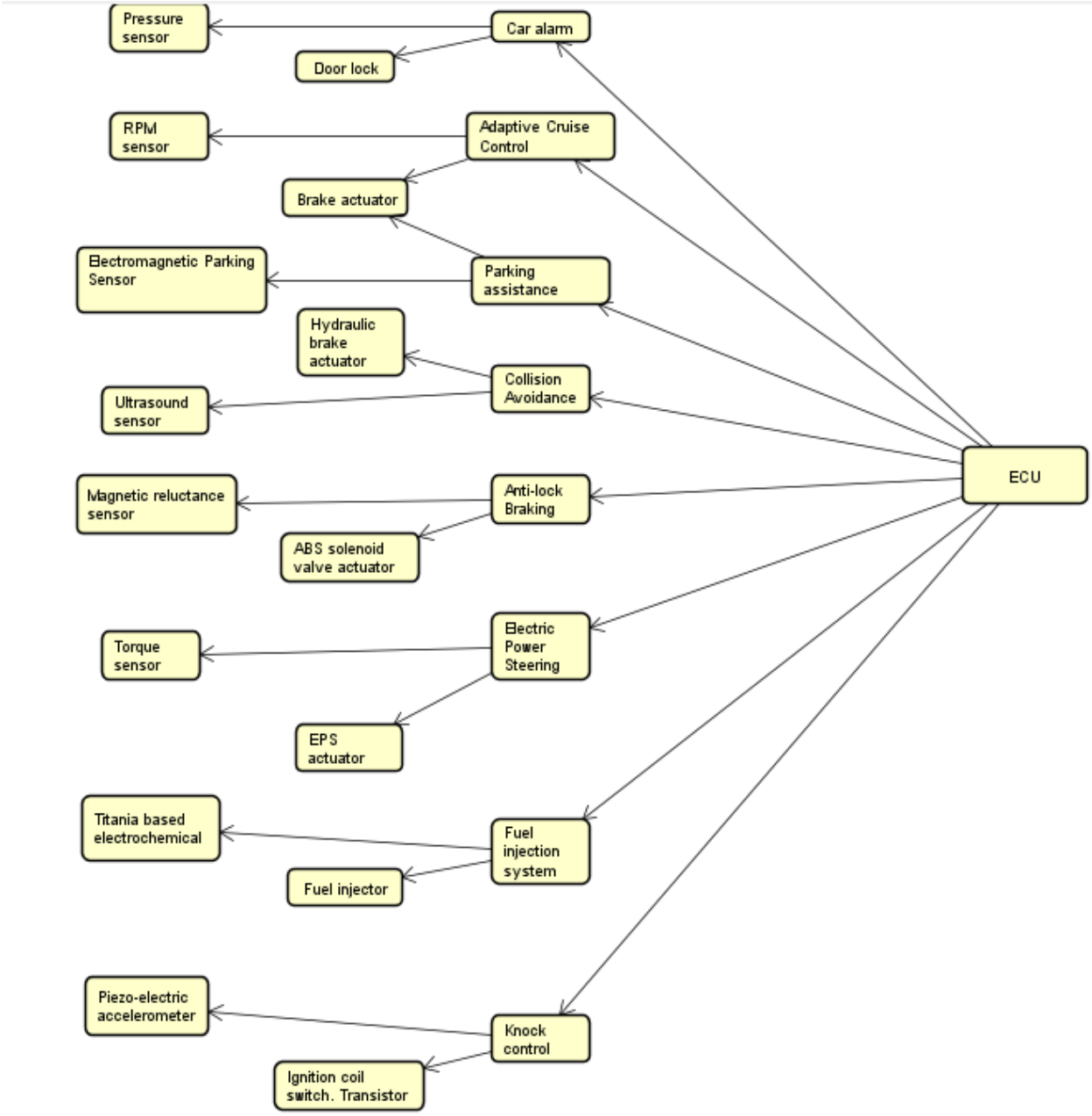
External companies use *decomposition view* to have a clear detail view to all the parts of the system that they are developing and using. External companies use *concurrency view* to have a concurrency structure of the system which represents parts of system that can be used concurrently in how it is coordinated and controlled. This is related to the parts of system that they are using and developing.

Driver (user) uses *uses view* because through that view it can be shown in what way he/she is using each unit of ECU system.

Government uses *concurrency view* to have a concurrency structure of the system which represents parts of system that can be used concurrently in how it is coordinated and controlled. Government needs to have a clear view of what does ECU system consist and what its main purpose is.

Estimated effort: 5 hours

Lab 2.2 1) Decomposition/uses view



Lab 2.2 2) List of elements

In this section we have listed and explained elements of Electronic Control Units in vehicle. Also, we have provided a little descriptions about them and relations between them.

Submodules of ECU:

Car alarm. Detection of unwanted actions on cars in form of a a siren sounds.

Adaptive Cruise Control. Possibility of the speed adaptation for keeping a fixed distance in the front of vehicle.

Collision Avoidance. Steering and breaking in case of possible collision.

Parking assistance. Warning for drivers when parking the car in case of getting to close to obstacles.

Anti-lock Braking. Preventing the brakes from locking.

Electric Power Steering. Assisting to the driver in steering the car with the help of an electric motor.

Fuel injection system: Used for determining the necessary amount of fuel, and its delivery into the engine.

Knock control: Used for determining internal combustion of the air/fuel mixture.

Actuators and sensors used by the elements:

Pressure Sensor. Using for measuring of sensors.

RPM Sensor. Using for measuring operating of some machine.

Brake Actuator. Converting the compressed air force within a trailer's air reservoir into a mechanical force.

Magnetic Reluctance. Allowing the permanent magnet's field for inducing a current in the pick-up coil.

Electromagnetic Parking Sensor. Sensor intended for alerting the driver to obstacles.

Ultrasonic (Ultrasound) Sensor. A device that can measure the distance using sound waves.

Torque Sensor. Device for measuring and recording the torque on a rotating system.

IAC Actuator. Device commonly used in fuel-injected vehicles to control the engine's idling RPM.

EPS Actuator - Actuator used by Electric Power Steering.

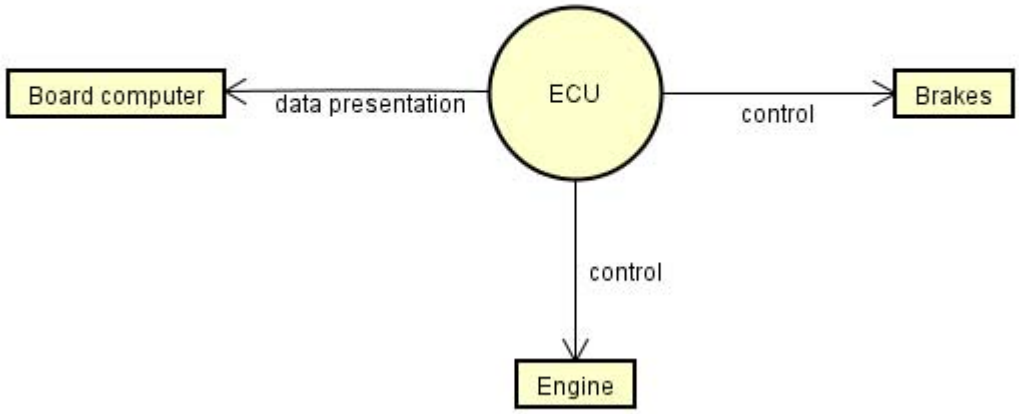
Hydraulic Brake Actuator - Actuator used By Collision Avoidance.

Door lock - Actuator used by Car Alarm.

Relations between elements:

As we can see on the decomposition view, there are subsystems of what are the elements of the big system ECU. Each of these subsystems has its own modules that are representing sensors and actuators. There are actuator modules used by more than one subsystem such as some actuator.

Lab 2.2 3) Context diagram



Lab 2.2 4) Variability guide

Architecture, from the perspective of this particular view, is created so that every unit of ECU system is independent. If some unit is missing it won't affect other units. It means that it supports variations in functionality. But if some part of ECU system's hardware stops working then the functionality that is connected to it won't work in appropriate way.

Lab 2.2. 5) Rationale of the design

We have chosen to design this view in a way that it is transparent and understandable. We have clearly separated the parts (levels) of this system. We have three parts (levels): ECU system, ECU system's functions (units), actuators and sensors. Actuators and sensors are in the same level but it looks separated in the view so that it is clear what is actuator and what is sensor because they don't do the same function. We have connected those parts with the lines so that it can be seen which part uses which part.

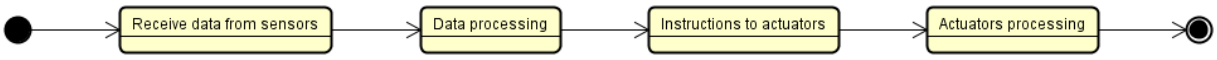
References:

Len Bass, Paul Clements, Rick Kazman, *Software Architecture in Practice*, Third Edition, ISBN 978-0-321-81573-6, Addison-Wesley, 2012.

Nick Rozanski, Eoin Woods, *Software Systems Architecture: Working With Stakeholders Using Viewpoints And Perspectives*, Second Edition, Addison-Wesley, 2011.

Estimated effort: 11 hours

Lab 2.3 1) Concurrency view



Lab 2.3 2) Concurrency view description

There are four elements shown in concurrency view:

Receive data from sensors. This element represent gathering data from sensors to ECU.

Data processing. This action is happening in ECU.

Instructions to actuators. Together with data processing, ECU sends instruction to actuator.

Actuators process. Actuators are executing their function.

Relations between elements:

At the same time it is possible to execute gathering data from sensors, it is not necessary to list action for every sensor in this situation. Same thing is with actuators process, they can do their task at the same time.

Lab 2.3 3) Variability guide

All states of the concurrency view are connected through relations and if some variations are done to each state it will affect on every next state. Changes on one sensor won't affect on the other sensors. The same is for the actuators.

Lab 2.3 4) Rationale of the design

We have chosen to design this view in a way that it is transparent and understandable. We have clearly separated states of this system. We have four states: receive data from sensors, data processing, instructions to actuator and actuators processing. We have connected those states with relations so that it can be seen which state depends on the previous one(s).

References:

Len Bass, Paul Clements, Rick Kazman, *Software Architecture in Practice*, Third Edition, ISBN 978-0-321-81573-6, Addison-Wesley, 2012.

Nick Rozanski, Eoin Woods, *Software Systems Architecture: Working With Stakeholders Using Viewpoints And Perspectives*, Second Edition, Addison-Wesley, 2011.

Estimated effort: 6 hours