Design & Assessment Method for the Integration of Driver Assistance Systems

Dr.-Ing. Jascha Freess, Robert Bosch GmbH, Chassis Systems Control
Robert Kornhaas, Robert Bosch GmbH, Chassis Systems Control
Dr. Michael Fausten, Robert Bosch GmbH, Chassis Systems Control

Abstract

In order to allow the realization of today's functionalities, automotive electric/electronic-architectures (e/e-architecture) have changed from an aggregation of isolated systems to fully interconnected networks. In parallel, the number of ECUs in passenger cars rose to 70 or even 80. The future is going to see a melt of system boundaries, when even more functionalities which link several systems together, are to be integrated. Coping with complexity will be even more important. Still, costs remain the main driver in the development of e/e-architectures. At Robert Bosch GmbH, a method based on graph theory was developed for the design and the assessment of e/e-architectures. It enables systematic analyses and gives indications for the structural design of architectures under development. In this paper, the method will be presented. The partitioning and mapping of software components of a predictive evasion function is used as an example. The method has been applied to several projects. Different architectural approaches for the chassis domain were assessed. One major result is a deduction of a hierarchical structure with centralized and scalable domain control units that will be presented at the end of this paper.

Introduction

With the evolution of mechatronic systems, more and more functions for improving safety, environmental protection, comfort and agility are being introduced in modern automobiles. Today, in state of the art automobiles most of them are controlled by software that is running on electronic control units (ECU). Multiplexing bus systems have replaced many wires and made a manifold of signals available to all functions. By 2010, up to 90% of innovations in the automotive sector will be realized in electrics/electronics (1) and e/e will account for 30% of a car's value by 2015 (2).

Software and bus systems along with the increasing power of microcontrollers allow for a higher integration. Systems boundaries are being melted. Original equipment manufacturers (OEM's) are given the opportunity to allocate and rearrange software components to the ECUs according to their needs. The new software standard AUTOSAR (3) is especially designed to support the reuse of software and the reallocation of software components.

Today's technology also enables a new type of function: high-level *networking functions* combine existing systems in order to create all-new functions. One example is a predicting evasion function that is being researched by Robert Bosch GmbH's program of CAPS – Combined Active & Passive Safety®. As shown in Figure 1, it receives information from existing radar and/or video systems and uses the electric power steering (EPS) in order to guide the driver in emergency situations. It can also modulate the brake pressure for optimum steering maneuvers. Characteristically, such functions are no longer bound to a specific system or ECU, but make advantage of existing ones. In many cases, the functions can even be adapted to different equipment scenarios. In example, the evasion can also work with alternative sensor concepts. Depending on the OEM the application software can be hosted by any ECU and hardware components needn't be manufactured by one single supplier.

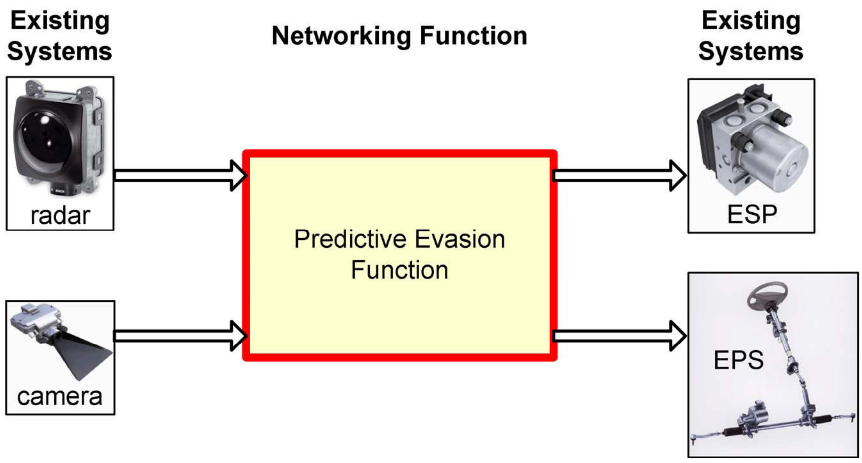

Figure 1: example of an unbound networking function

In addition, these trends are driven by a rising pressure on cost (2) and development efficiency. Along with the integration of new functions, finding an optimum for the structuring of the complete e/e-architecture is key factor to a successful vehicle concept. For systems design and architecture development, a wide exploration space is to be assessed in order to find a concept that is ideal in saving cost while sustaining quality. Coping with the amount of information and the complexity of an automobile's complete electric, electronic and software infrastructure requires new methods for systems design and architecture assessment (4).

Automotive system structures

Many systems of today's world are linked together to either do information exchange or interaction between components of a system to provide different functionality. Examples are the communication business or the supply chain of merchandise. The topologies of such systems are structured in different ways in order to fit best for fulfilling functional and non-functional requirements. For the automotive world the key characteristics are high volume production, limited space and weight for components, adverse ambient conditions, highest functional safety, reliability, robustness and quality targets on a low maintenance base and energy and fuel economic.

For vehicle electronics, the electric/electronic system can be divided into five domains, corresponding to similar functional tasks:

- body
- infotainment
- power train
- passive safety
- chassis

Within a domain, systems and functions are related closely to each other and linked together and represent a vertical organization of the system. If functions are spanning over domain boundaries, structuring is getting more complex.

An alternative way of partitioning is to divide functions by their tasks and represent a horizontal organization of the system:

- **Low-level functions** such as the generic assistance torque controller of an electric power steering (EPS) are closely linked to control specific hardware components.
- **High-level functions** e.g. networking functions like an evasion assist, make use of several low-level functions (so called support functions) on one or different ECU's.

This defines logical abstraction layers and leads to a hierarchical function structure that can be paralleled in hardware. If an ECU controls external components, a **centralized** structure is realized. Control can be shifted to external components until they are coordinated or commanded by a central instance, but decide by themselves. Thus, a **decentralized** structure is implemented. Accordingly, **intelligent components**, e.g. sensors and actuators with build in electronics, are in charge of their own low-level control. On the other side, electronics and control can be included in a **separated ECU** with sensors and actuators as peripherals.

Different models for the networking topology offer a range of structures from strictly hierarchical to "all-flat" and allow for any combination that would fit best.

The final e/e-architecture of a car is a combination of the above shown structuring approaches. Car manufacturers (OEM) are optimizing it for best fitting to functional and non-functional requirements and every car manufacturer has different priorities and constraints.

At Bosch a method was developed to support the top-down design by dividing the systems and finding well-suited mappings of functions on ECUs.

E/E-Architecture design method

The decision for a specific architectural design is being influenced by many factors. Some can be described as dependencies of parts of the e/e-architecture, such as the communication between functional building blocks. The analysis of such dependencies gives indications for the design of a suitable concept.

The method consists of the following steps:

STEP 1: Separation of parts
The e/e-architecture is being split-up in its parts. The parts are being separated into different categories, e.g. software components/functional building blocks, ECUs.

STEP 2: Identification (quantification) of dependencies
The relationships between parts are identified and, if possible, quantified for its magnitude. Relationships can exist between parts of the same or of different categories. More than one relationship can exist between two parts. Different dependencies (types of relationship) are kept strictly separated from each other.

STEP 3: Analysis of dependencies
The analysis is based on methods of graph theory. Parts are considered as nodes, also declared as architectural building blocks. The magnitude of relationship is weighting edges. Each dependency forms a graph of its own. In addition, graphs can be combined in superposition in order to show a completely weighted result.

STEP 4: Clustering of parts
With algorithms of graph theory, clusters of parts, that are more tightly related to each other, are being derived. The clusters give indication for the structuring of the e/e-architecture.

The ideal granularity level of separation of parts has not yet been finalized. A block that is not separated into parts can be considered as a pre-defined cluster. Therefore, it is expected, that – in case of future hierarchical decompositions – similar results can be achieved. The method assumes, that the types of dependencies have already been defined in advance. For the superposition, the meta-dependencies are being used that represent weighted relationships between dependencies. Meta-relationships are pre-defined and fixed. They do not vary with the architecture that is to be designed and therefore needn't be analysed during the design process.

Example application

In this section, the integration approach of the above described predictive evasion function into a future architecture design will be shown. It is a simplified example of an architectural design problem in order to illustrate the design method. Also, from a catalogue of dependencies that is currently being established at Robert Bosch GmbH only three dependencies will be considered:

- Communication (bandwidth)
- Geometric constraints (distances for the length of wires)
- Power distribution (energy transfer)

The use of algorithms of graph theory in this example is reduced to the principle of gravity (nodes repulse each other unless they are attracted by edges). The weights of the edges are shown as their lengths within graphical illustration. The tighter a relationship is, the closer its nodes are depicted. Graphical demonstration is an especially useful feature as clusters can be seen easily.

The work with graph theory during the analysis and clustering is supported by a software tool. We have chosen LOOMEO (5) of TESEON GmbH, Munich. It provides tables for input and administration of relationships and dependencies. Several algorithms of graph theory are provided. For good interpretation of the results the graphical illustrations of graphs as shown in this example is very useful.

In the first step, the evasion function is separated into its parts as shown in Figure 2. The categories are functional control (software components), hardware (sensors and actuators) and localities. Each kind of relationship between those building blocks is treated separately and labeled for its kind. In the example of the evasion function signal bandwidth, distance and energy transfer are labeled separately. For reduced size and complexity, Figure 2 doesn't show underlying low-level functions, even though they can be seen as nodes in the graphical illustrations.

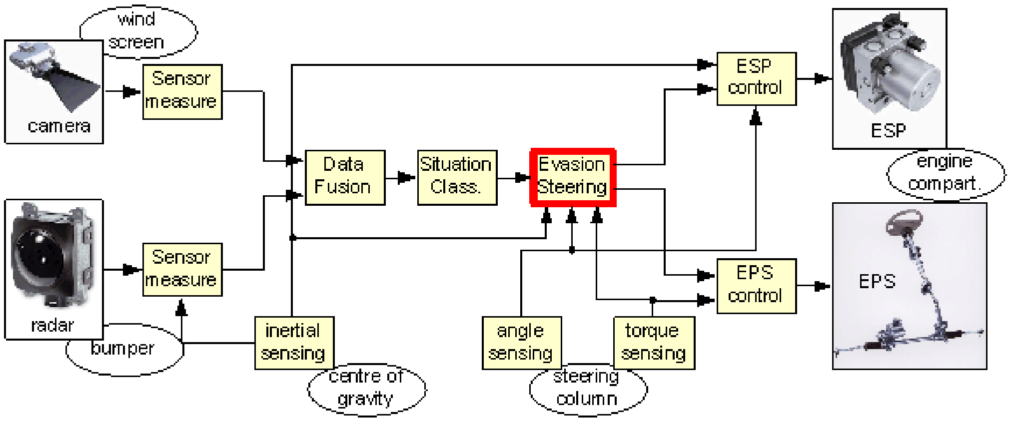

Figure 2: example for separation of parts

Figure 3 shows the dependency of communication as a graph. The illustration is based on the principle of gravity. The length of the edges and the distance of the nodes represent the strength of the signal transfer relationship between the architectural building blocks. The color of the edges also depends on the strength: Thresholds can be set that mark any strength of interest. Parts that do not have communication relationship are not shown in this graph.

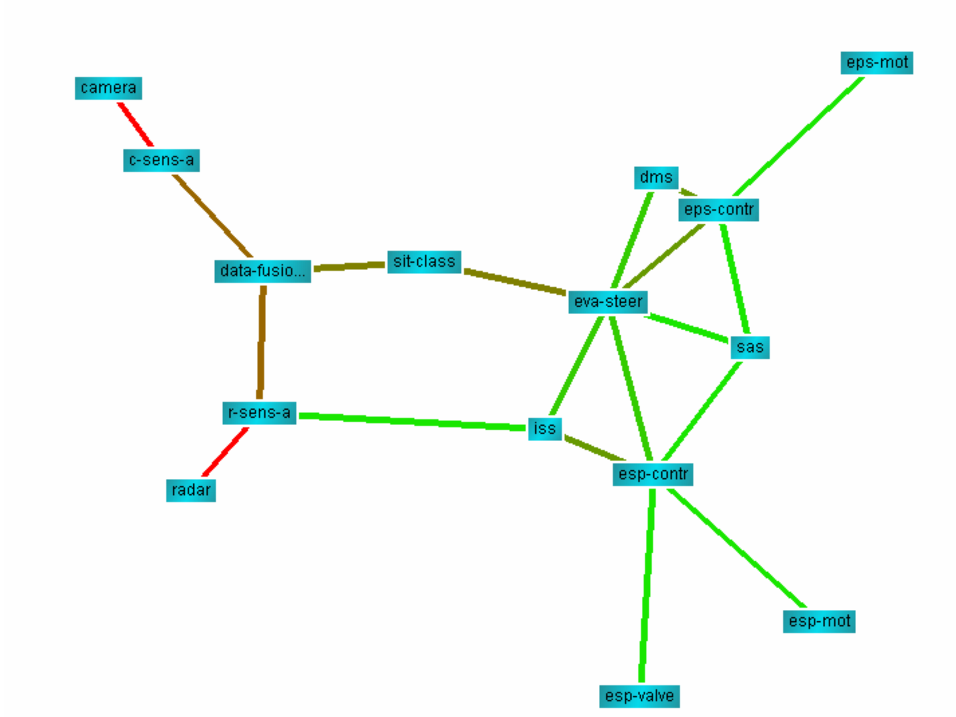

Figure 3: graphical analysis of communication

The geometrical constraints are displayed as another graph (Figure 4). Here, the length and the color of the edges between the parts represent the geometrical distances (strength of the geometrical relationship) between relevant parts. Parts without edges (e.g. functional building blocks) do not qualify for geometrical relationship and therefore have no such dependency.

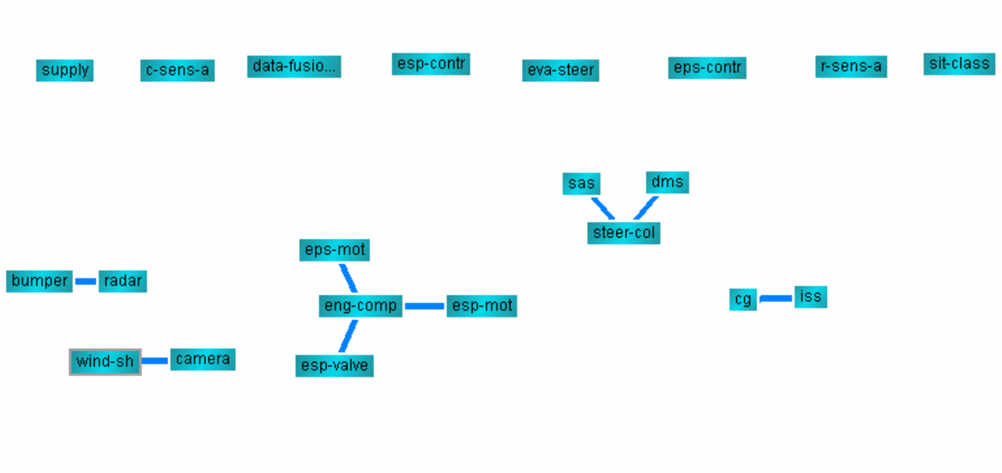

Figure 4: graphical analysis of geometrical constraints

In the same manner, the dependency of power distribution is displayed as a graph. From Figure 5 one can easily see, that the evasion function is mainly controlled by software (compares to functional building blocks) and with signals that are transmitted on bus systems. The energy transfer is reduced to only one edge.

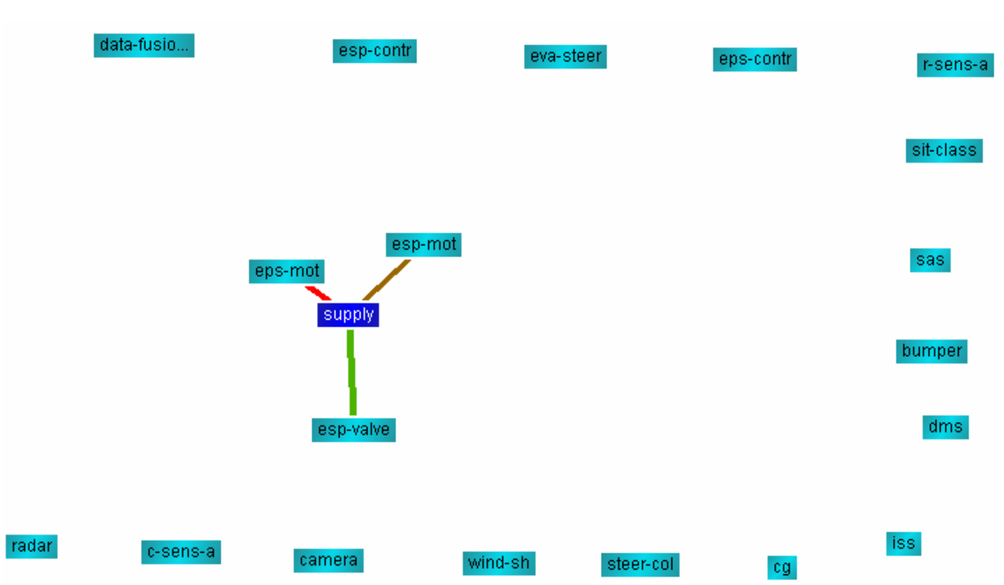

Figure 5: graphical analysis of energy transfer

Figure 6 shows the superposition of the dependencies of communication, geometrical constraints and energy transfer in one graph. Now, the length and the color of the edges represent the strength of the relationship between the parts that are the result of the combination of all three dependencies. The result of the investigation reveals a grouping of the parts into seven clusters.

Figure 6 shows a grouping of video and radar sensor parts (clusters #1, #2) together with their signal processing functions. The clusters are driven by the required high bandwidth (communication) and the best position for the sensors (wind-shield and front bumper). Clusters #3 and #4 are driven by similar reasons. Whereas clusters #5 and #6 bring actuator parts together with their control functions for the low-level control of basic functionalities of the actuator. The positions of these parts are as close as possible to the mechanical actuator in order to minimize loss of power. In this example, the method indicates that the parts within each of these clusters should be set up as one intelligent hardware component with microcontrollers of different sizes for the software components (functional integration).

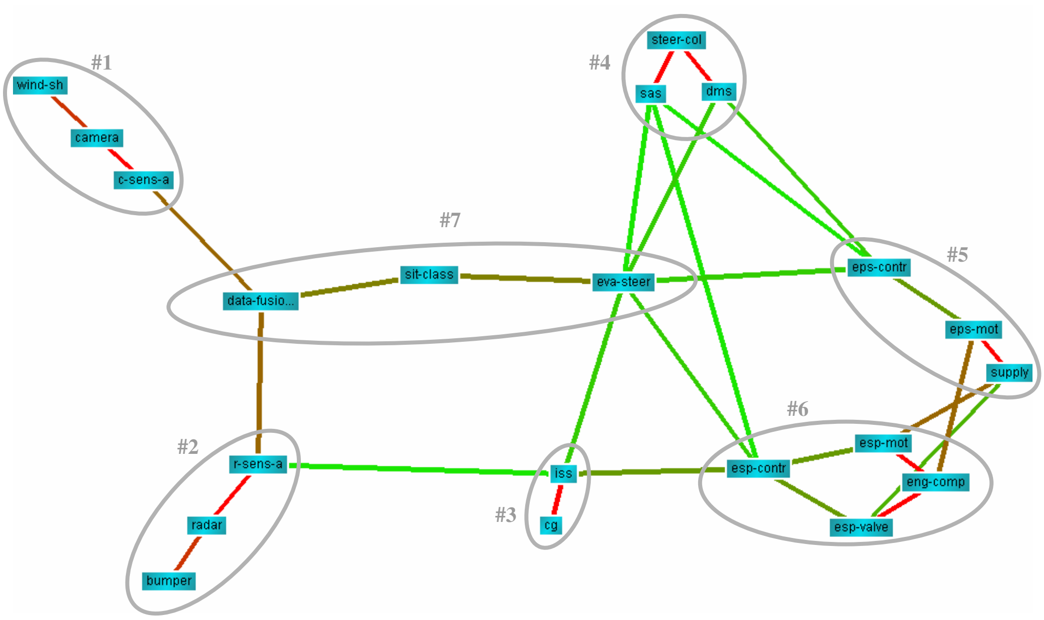

Figure 6: superposition of dependencies and clustering of related parts

Cluster #7 collects three parts that stand aside by themselves, not having as tight dependencies as the parts of the other clusters. Potential targets for their mappings are not as obvious as it is for other functional building blocks. In this example, only three dependencies were used. Additional ones will give more obvious indications and will support the definition of the architectural structure.

Design and assessment results

The presented method has been applied in several projects in order to derive systematic suggestions for e/e-architectures. One of the projects is shown as a simplified example above which also demonstrates the plausibility of the method. The clusters suggest a structure with intelligent sensors for radar/video processing and for the inertial and steering sensing as well as with intelligent actuation for braking and steering.

The above is also an example for what was found in several analyses. There are two different types of clusters: *fixed* (clusters #1-#6) and *unbound* (cluster #7) in respect to the location in the car. These types reflect the split-up into low-level control functions that are most likely seen as fixed and unbound high-level networking functions. Unbound parts can either be combined with others in intelligent sensors/actuators at a fixed location or can constitute a separate ECU. The latter transfers the hierarchical function structure into a hierarchical hardware architecture with central control units.

However, additional hardware is expensive. According to Figure 6, cluster #3 is a smaller one that is clearly apart of the others. State of the art architectures use a sensor cluster of inertial sensors in the centre of gravity that is connected to the ESP via CAN. Thus, the geometrical integration of the parts of cluster #7 within cluster #3 will give synergies. This results in an extension of the sensor cluster towards a central ECU that provides with additional computational power. It hosts high-level networking functions that control the functionality of a domain (here: of the chassis domain). It also allows for the cost efficient integration of the inertial sensor signal processing with run-time checks and monitoring which is currently mapped on the ECU of the ESP.

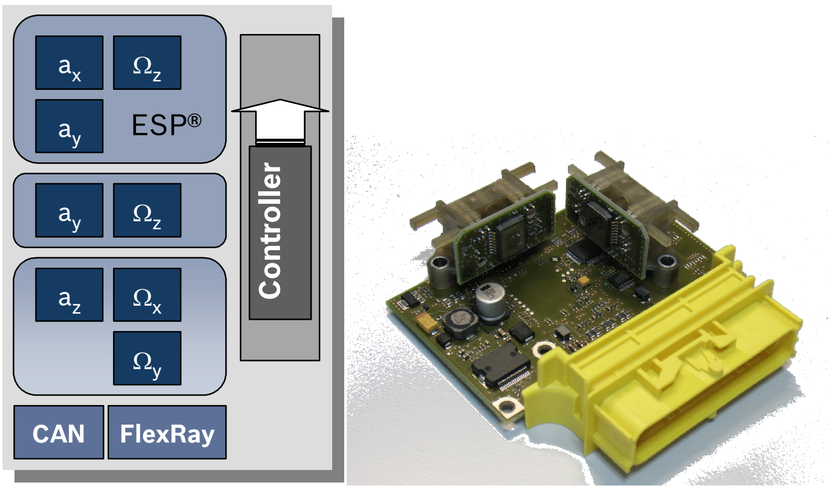

Figure 7: domain control unit as an example for a centralized ECU

Such a centralized domain control unit (DCU) could look like the example that is shown in Figure 7. As consequence of the investigation and transferring the results to other domains, a hierarchical topology of the overall e/e-architecture with one central ECU per domain would result (example shown in Figure 8).

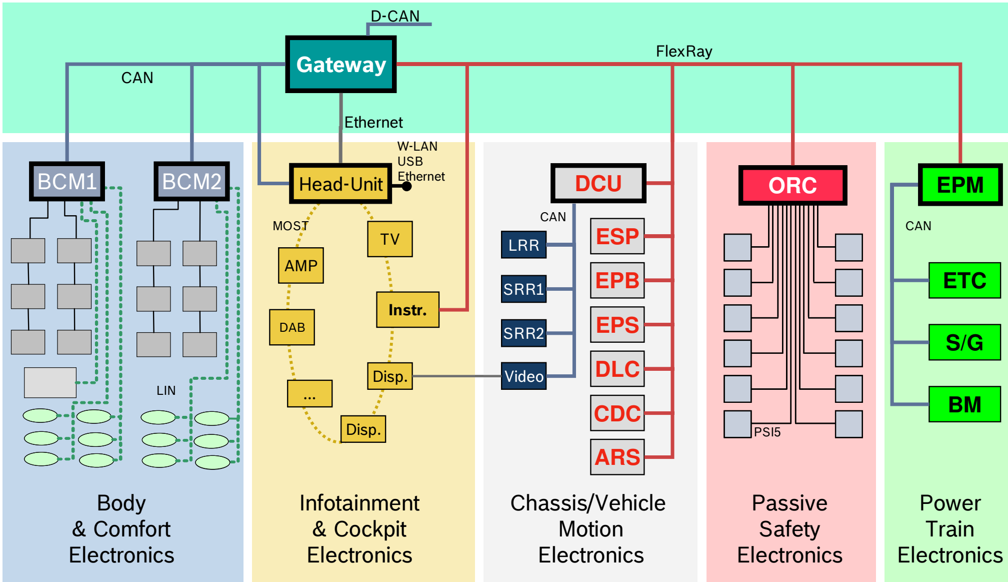

Figure 8: resulting network topology

The example of the evasion function demonstrates the method on a smaller project. More complex projects account for a much higher amount of information and the number of parts (and nodes) is supposed to rise up to several hundred when complete architectures are to be analysed. Thus, besides the increasing effort, the complexity of data input and handling rises as well. Even for partial analyses of e/e-architectures conventional methods (e.g. matrices) show to be ineffective and error-prone. Another tool for modelling of e/e-architectures and input data is necessary. This is addressed in (6).

Conclusion

In this paper, a method for the systematic design and assessment of e/e-architectures by application of graph theory is presented. The method provides a good indication on how to structure an e/e-architecture. It is an early design step in order to limit the exploration space that is being spanned by available options. Therefore, the analysis provides the advantage of fast derivations of better fitting designs. Resulting architectural concepts have a higher stage of optimization. The method in combination with a software tool for analysis and graphical demonstration can help to display complex structures and dependencies. The complexity is not reduced, but coping with it is made easier.

The derived structures of e/e-architecture designs will be reviewed and double-checked to assure that the e/e-architecture fits the needs. Qualitative and quantitative criteria are for example: sizes, platform variation, suitability for system integration, reliability, development risk and effort, cost. The selection and weighting of the criteria depends strongly on customer preferences. In combination with different analysis and customer priorities and constraints, clusters can be calculated and identified.

Considering complete vehicles, several investigations were performed. According to an intensive assessment of different approaches, the best comprise a structure consisting of a backbone architecture for the different domains. The domains themselves are structured hierarchically with centralized domain control units and intelligent sensors and actuators. The centralized, hierarchical architecture that is described above reduces significantly cost and complexity. It is a platform for geometrical and functional integration that allows for a flexible handling of variants and configurations. In detail:

- Standardized, electrical and intelligent electronic components (satellite sensors and actuators) without customer specific algorithms and interfaces
- Domain control units with high amount of microcontroller performance and resources for customer specific algorithms, interfaces and communication links => Platform for geometrical and functional integration
- Connection between domain control unit and satellites via low- or medium band width communication link (CAN, LIN, PAS, PSI, ...)
- Connection between the domain control units via high speed communication link (FlexRay)

Acknowledgement

- CAPS – Combined Active & Passive Safety® is a registered trademark of Robert Bosch GmbH.
- Analysis and graphs were supported by the software tool of LOOMEO (5).

References

(1) J. Koetz, "Elektronik Architekturen über Baureihen - Chancen und Risiken", VDI-Bericht 1907, VDI-Verlag, Baden-Baden, 2005, pp. 487-494

(2) Mercer Management Consulting GmbH, "Mercer Studie Automobilelektronik", Munich, 2006

(3) "Automotive Open System Architecture". Available at http://www.autosar.org

(4) R. Belschner et. al., "Gesamtheitlicher Entwicklungsansatz für Entwurf, Dokumentation und Bewertung von E/E-Architekturen", VDI-Bericht 1907, pp. 511-521, VDI-Verlag, Baden-Baden, 2005

(5) "LOOMEO development group". Available at http://www.loomeo.com/en

(6) S. Powolny et. al., "Model based top down process for automotive E/E-architecture development", SAE World Congress 2008, Detroit, 2008