



ARCHITECTING COMPLEX ENGINEERED SYSTEMS

U. Sellgren^{1,✉} and D. Williamson^{1,2}

¹ KTH Royal Institute of Technology, Sweden, ² Scania CV AB, Sweden

✉ ulfse@kth.se

Abstract

Novel products are commonly realized by integrating heterogeneous technologies. Product architecting focus on defining the scheme by which the product functions are allocated to physical components. A DSM-based clustering method that integrates technical complexity and strategic concerns has previously been proposed. It has been shown that interaction weights in the DSM may affect the clustering result. A complexity-based interaction strength model to be used in DSM clustering is proposed here. The case study gives promising results from both interaction performance and safety points of view.

Keywords: design structure matrix (DSM), complex systems, systems engineering (SE)

1. Introduction

Functional performance of high-performing physical products is commonly realized by synergistic configurations of components based on heterogeneous technologies, such as mechanical, electrical, electronics and control software, i.e. mechatronics or Cyber-Physical Systems (CPS). Such products are commonly referred to as complex systems. Complexity is also described as not just unavoidable in systems with heterogeneous technology, but actually required (Rodriguez-Toro et al., 2004). In engineering design, complexity is often viewed as a measure of the uncertainty of fulfilling the specified functional requirements (Suh, 1999), i.e. it is viewed as a relative measure of the actual knowledge compared to what we want/need to know. Uncertainty is usually reduced as we gain knowledge in the development process, but a dilemma is that decisions must be made on incomplete or fuzzy knowledge. Törngren and Sellgren (2019) discussed complexity related to development of CPS and presented and discussed six complexity facets causing humans to perceive complexity and five consequences of complexity. The six facets are heterogeneity/diversity, size and computability, uncertainty and change, dynamics and/or structure, incidental/essential, unintended/accidental.

The complexity challenge is commonly addressed with systematic, and structured processes with a systematic capture of performance and behaviour of the evolving product from physical and virtual tests. *Systems engineering* (SE) is a structured product development process (INCOSE, 2007). The SE process is conceptually represented as the V-model (see Figure 1 - left). The main purpose of the V-model is to enable technical complexity to be managed by moving from abstract to concrete in a stepwise manner and to decompose the large problem into several smaller problems in the process and, thus, enabling the development team(s) to “eat the elephant bite by bite”.

Product architecture is “the scheme by which the function of a product is allocated to physical components”, and more specifically “(1) the arrangement of functional elements; (2) the mapping from

functional elements to physical components; (3) the specification of the interfaces among interacting physical components” (Ulrich, 1995). Ulrich and Eppinger (2000) focused on the interactions and stated that product architecture is “the scheme by which the chunks (i.e. the modules) of a product interact”. Product architecture can be categorized as being modular, integral, or hybrid based on the type of mapping between functional elements and physical components (Höltkä-Otto, 2005). *Product architecting* is an early, and highly iterative, sub-process within SE (Figure 1) that involves conceptual system design, product building block identification and product layout design (Dieter et al., 2013).

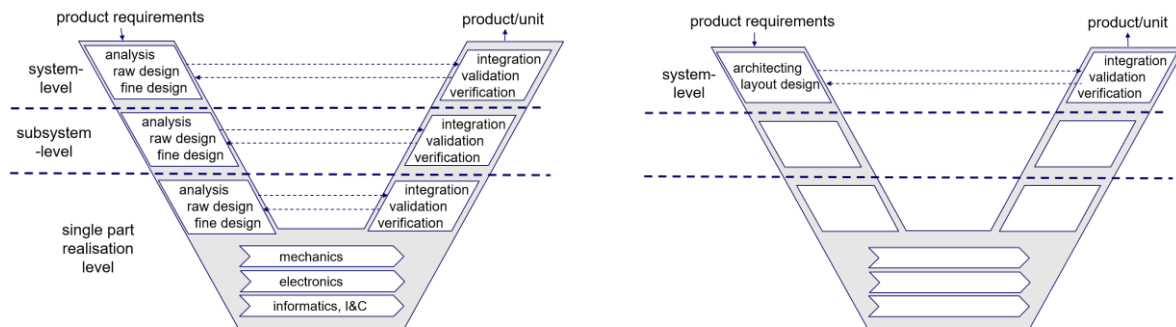


Figure 1. The V-model (VDI, 2004), left, and the architecting sub-process, right

Model-Based Systems Engineering (MBSE) may be defined as “the formalized application of modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases” (INCOSE, 2007). It is generally believed that MBSE may increase the pace at which knowledge is gained when developing novel products, by enabling a faster complexity reduction than with traditional document-based SE processes. Furthermore, MBSE also enables the product development team to keep the design degrees of freedom for a longer time by elaborating and experimenting more with computer models and less with physical prototypes as in traditional SE processes (see Figure 2).

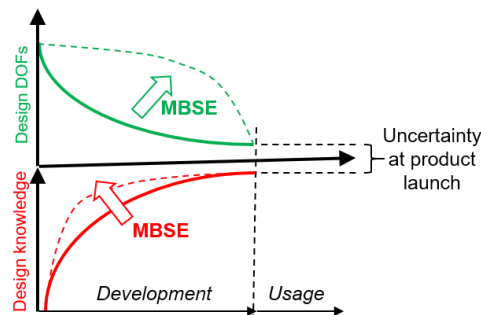


Figure 2. Potential benefits of MBSE, from (Törngren and Sellgren, 2019)

Graphs and matrices are the two most commonly used representations of component-to-component interactions. The two representations are complementary. The most important property of the Design Structure Matrix (DSM) (Eppinger et al., 1994), also commonly referred to as the Dependency Structure Matrix, is that it is scalable from small conceptual system to very large detailed system (Börjesson and Sellgren, 2013). Cluster analysis of a DSM is one of several methods to identify groups of components, i.e. clusters, with minimum dependencies between the clusters. Cluster analysis is an efficient method to find clusters of components, i.e. subsystems or modules that concurrently can be further engineered and detailed, i.e. it is a way to reduce negative effects from technical complexity. Eppinger and Browning (2012) proposed four generic types of interactions/relations between pairs of function carriers, i.e., components, in a Product Architecture DSM (paDSM). These relation types are *spatial* relations and functional flow of *matter*, *information* and *energy*. Relation weights, also known as interaction strengths, can be used to represent their relative importance.

Business strategies, like planned upgrades, in-sourcing/out-sourcing, purchase/make, etc., e.g. (Blackenfelt, 2001), add significant complexity to the architecting stage. In an attempt to simultaneously

treat technical (structural) complexity and company-specific business strategies, [Williamsson and Sellgren \(2016\)](#) introduced the Integrated Modularization Methodology (IMM), which utilizes the DSM clustering IGTA++ algorithm developed by [Börjesson and Sellgren \(2013\)](#). The core of IMM is to integrate company-specific module drivers (represented by the MIM) with a paDSM into a strategically adapted DSM (saDSM), which is then clustered. [Williamsson et al. \(2018\)](#) showed that IMM can propose reasonable module candidates from both technical complexity and company specific strategy points of view, i.e. to assist *module identification*. The next product architecting step focuses on product *layout design* ([Dieter et al., 2013](#)), which is a highly iterative sub-process. Making a rough spatial layout of the principal product components enables analyses of potential spatial, thermal, or electrical interferences between components within the module candidates. [Williamsson and Sellgren \(2019\)](#) augmented the IMM matrix, by including interferences, i.e. unintended/harmful interactions model, as proposed in the Affordance Design Theory. In order to distinguish desired from undesired relations, positive and negative values could be used to represent interactions between DSM components, where a negative value would indicate an undesired/harmful interaction. [Williamsson and Sellgren \(2019\)](#) showed that paDSM clustering with negative relation weights, representing undesired/harmful relations, may propose clusters with internal physical interference, but that the affordance-extended IMM approach is capable of proposing clusters with no conflicting interferences, nor conflicting strategies. The IMM method is, thus, capable of representing and analysing effects from technical complexity, strategic aspects and physical interferences, and to do that in any combination.

To compare clustering results from different alternative DSM and IMM representations, [Williamsson et al. \(2018\)](#) developed a Cluster Match Matrix (CMM) and used the CMM to show that the relative weights of the different types of functional relations could have a significant influence on paDSM clustering. How to define proper and useful weights of the different types of interactions, dependencies and interferences is still an unresolved issue.

This paper aims at providing a contribution to MBSE by enabling paDSM-based clustering of complex heterogeneous systems. More specifically, the specific objective is to extend the function-means oriented IMM clustering method by proposing logically reasonable weights for the intended technical interactions/dependencies between the system constituents. A delimitation is that strategic reasons and layout constraints derived from hazardous or unwanted interferences are not treated here. The following three research questions are addressed:

- How could we weight the importance of mechanical, electrical software components in heterogeneous systems?
- How could we weight the importance of spatial relations, energy, mass and signal interactions between pairs of system components?
- How could interactions with performance critical and safety critical components and interactions be properly treated?

The research is based on a case study of a battery-electric driveline of a heavy duty truck. In section 2, a complexity-based weighting method is proposed. Section 3 elaborates on a case, where the proposed weighting method is applied. The results are presented and discussed in sections 4 and 5. Section 6 and 7 conclude the work and proposes further research.

2. Complexity approach to represent interaction weights

2.1. System complexity

Complexity is not just unavoidable in systems synthesized from heterogeneous technologies, such as CPS, but could also be actually required. Consequently there is a need for complexity management techniques ([Rodriguez-Toro et al., 2004](#)). A multitude of complexity measures have been proposed in literature. Such definitions tend to fall into two categories ([Rodriguez-Toro et al., 2004](#)):

- *Structural-based*, where complexity is considered a property of the object, i.e. a system property, which depends on number of components, number of interactions between components, number of types of components and interactions, etc., e.g. ([Patzak, 1982](#)).

- *Information-based*, where complexity is perceived, i.e. considered a property of the subject, i.e. the human “observer”, e.g. (Casti, 1992).

These two categories of complexity measures have usually addressed challenges related to different SE aspects, such as product design (Ghosh et al., 2018), manufacturing (Suh, 1999; Mattsson et al., 2011), assembly (Mattsson et al., 2014), logistics (Blecker et al., 2005; de Leeuw et al., 2013). The approach used here is to represent and treat complexity as a quantifiable and intrinsic system property that can be used in architectural analyses with the purpose to present architectural cause and effect information/knowledge to the actor subject(s).

Pugh (1990) proposed a quantitative structure-based complexity measure:

$$C = \frac{K}{f} \sqrt[3]{N_p N_i N_t} \quad (1)$$

where N_p is the number of parts, N_t the number of part types, N_i the number interconnections, f the number of functions, and K is (an arbitrary) constant.

2.2. Interaction strengths

The complexity measure in Equation (1) is not just a “measure by numbers”, but also a recognition that component variety and the number of relations between the components add complexity and that known purposes, i.e. functions, provide meaning and, thus, reduce uncertainty and perceived complexity. When researching the connection between product complexity and vertical integration in the automotive industry, Novak and Eppinger (2001) found that “when a product involves novel architecture or technologies, there is not a stable and well-understood set of interactions between components. The process of identifying and understanding these relationships adds to the difficulty of coordinating development.” Consequently, they added product *novelty* as a third aspect, complementary to the number of components and the number of component interactions, in their proposed product complexity model.

In the present study, we focus on the complexity caused by interactions at the interfaces between pairs of components, and the strength of the interaction/interference, or dependency, at each interface. As shown in Figure 3, we represent the interaction at interface i as relation r_i , which is further defined as a spatial relation r_{is} , flow of information r_{ii} , energy r_{ie} , or matter r_{im} , or a combination thereof. The strength of an interface relation, or the level of dependency, is represented by a weight factor W_i which is the sum of the potentially relevant four functional interface relations, i.e.:

$$W_i = W_{is} + W_{ii} + W_{ie} + W_{im} \quad (2)$$

The weight of each of the four types of functional relations W_{ij} , where $j=s, i, e, \text{ or } m$ is proposed to be:

$$W_{ij} = r_{ij} \cdot \sqrt[3]{W_{ijn} W_{ijp} W_{ijs}} \quad (3)$$

where r_{ij} = the type of functional relation, represented by the value 1, 0, or -1, if it is intentional, not active, or negative physical interference, respectively, W_{ijn} = implementation novelty, with values 1 (standard), 3 (partly new), or 9 (completely new), W_{ijp} = performance importance, with values 1 (standard), 3 (important), or 9 (critical), and W_{ijs} = safety criticality, with values 1 (standard), 3 (important), or 9 (critical). The weight scheme is thus, consistent with the weights of each strategic reason for a function carrier, i.e., a principal or detail component, to be a product module as used in the Module Indication Matrix (MIM), which is a core representation in the Module Function Deployment (MFD) modularization methodology, e.g. (Erixon, 1998). Mattsson et al. (2011) proposed a 0,1,3,9 weighting scheme for the complexity index (CXI) for quantifying perceived production complexity, and argued, but did not verify, that it might be beneficial to also add a weight of 5, to bridge the gap between 3 and 9.

An example of a signal transfer function that is performance-critical (weight = 9) is the transfer of video signals (very large bandwidth) from a camera sensor in an autonomous vehicle to an object recognition ECU unit. In a vehicle operated by a human driver, assisted with efficient adaptive cruise control, the performance of the signal transfer is not that critical, suggesting a performance weigh value of 3. With regular cruise control, a reasonable weight value could be 1. The signal transfer may be more or less safety-critical, depending on the reliability of the transfer mechanism and if there is redundancy or not. If we leave out image sensor(s), reliability, photosensitivity, resolution and sampling frequency, and ECU

performance in terms of reliability and speed, we might find that it is crucial, from an active safety point of view, to have an as short “line of command” as possible from the decision control unit to the actuators in the manoeuvring system. In such a case, the signal transfer would have a safety-critical weight of 9. The performance of the signal transfer from decision unit to actuators, would benefit from minimum delay of signals with a small bandwidth, implying a performance weight of 3.

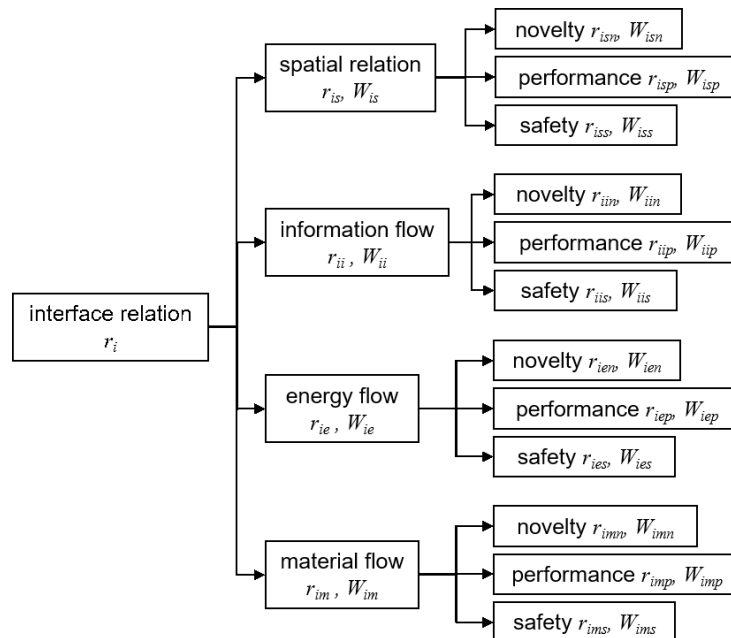


Figure 3. Functional relations r with interaction strengths, or weight factors W

3. Case

Figure 4 shows a conceptual candidate for an autonomous heavy-duty tractor for long-haulage logistics hub to hub transport. The basic concept is a 6x4 Battery Electric Vehicle (BEV), that is a six wheeler with four wheels driven and the two remaining wheels steered. The maximum gross train weight (GTW) is 44 tons and the required range is 800 km driving between logistics hubs on highway. A pre-study proposed, based on the required total propulsion force, four frame mounted permanent magnet high voltage AC axial flux motors, hub reductions mounted on each driven wheel, leaf spring suspensions on all six wheels, a multitude of sensors for object recognition, road assessment and condition monitoring, and a protective cover. The required set of batteries were mounted in the frame with a battery charging unit (BCU) assisted by necessary inverters and charging control (battery performance aspect), sensors and ECU:s with battery state monitoring and control and thermal management (safety aspect), as shown in Table 1.

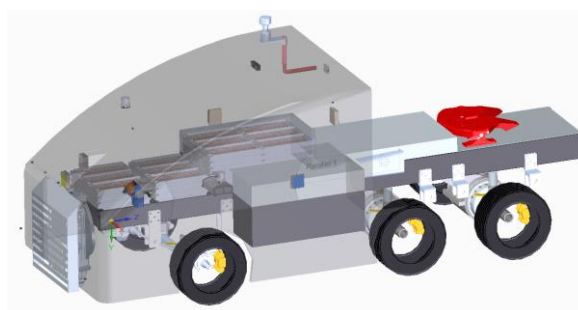


Figure 4. Heavy-duty autonomous truck with “fifth wheel” active trailer connection interface

The mean novelty value of the 88 principal components is 4.27 and the rms value is 5.41, which indicate that a significant engineering effort is required to realize this product, i.e. it is a high risk project.

Table 1. The 88 principal components and their speculated level of novelty

Function carrier	# of	Novelty	Function carrier	# of	Novelty
Truck frame	1	1	Battery heater	1	1
Truck cabin body	1	3	Battery cooling lines	1	1
Wheel suspension	6	1	Battery heating lines	1	1
Wheel hub	6	1	Object recognition unit (OR)	1	9
Friction brake	6	1	Camera sensor units	4	9
Hub reduction	4	3	Lidar	4	9
Anti-lock brake control, ABS	1	3	Radar	4	9
Steering gearbox	1	1	Sonar	4	9
Steering motor	1	1	Active safety analysis unit	1	9
Steering motor controller	1	3	Active safety management	1	9
Traction motor	4	3	Condition monitoring	1	9
Traction motor control unit	4	3	Condition management	1	9
Batteries	1	9	Road assessment unit	1	9
Battery charging unit (AC-DC)	1	9	GPS	1	3
Battery state analysis unit	1	3	Position analysis	1	9
Battery voltage monitoring	1	3	Motion planning	1	9
Battery current monitoring	1	3	Power and steering management	1	3
Battery temperature monitoring	1	3	Remote state communication	1	3
Battery safety protection unit	1	3	Remote override and diagnostics	1	9
Thermal management unit	1	3	Electronic stability program, ESP	1	3
Electric inverter and control unit	1	9	Angular wheel sensor	6	3
Cooling/heating pump	1	1	Traction control unit	1	3
Heat radiator	1	1	Gyro	1	3
			Total # of principal components	88	

Table 2. Interaction/dependency types, novelty, performance and safety criticalities

Interacting pair of components	Type	Nov.	Perf.	Safe.	Comment
Frame - Cabin body	s	1	3	1	Large force
Frame - Suspension (6)	s	1	3	1	Large force
Suspension (6) - Wheel hub (6)	s	1	3	1	Large force
Wheel hub (6) - Friction brake (6)	s, e	1, 1	3, 9	3, 9	Safety critical
Hub reduction (4) - Wheel hub (4)	s, e	1, 1	3, 3	3, 3	Large force, high safety
Traction motor (4) - Hub red. (4)	s, e	1,1	9,9	3,3	Performance critical
Traction motor (4) - Frame	s	1	1	1	Standard
Motor controller (4) - Motor (4)	e	1	9	1	Performance critical
Batteries - Motor controller (4)	e	1	9	1	Performance critical
Electrical cntrl. - Motor cntrl. (4)	i	1	9	9	Performance & safety critical
Battery monit. (3) - Battery state	i	1	3	3	High power perform. and safety
Battery state - Battery safety	i	1	3	3	High signal performance & safety
Battery temp. - Temp mngmt.	i	1	3	9	Safety critical
Temp mngmt. - Battery safety	i	1	1	9	Safety critical
Battery safety - Cooling pump	i	1	1	9	Low performance, safety critical
Battery safety - Electric control	i	1	9	9	Performance and safety critical
Camera package (4), Lidar (4), Radar (4), Sonar (4)(on cabin body) - OR	(s), i	(1), 1	(3), 9	(3), 9	Performance and safety critical
.....					

The four functional interactions (spatial *s*, energy *e*, matter *m*, information *i*) are all given the same weight of 1.0. Although, some researchers, e.g. (Stone et al., 2000) that have thoroughly reverse

engineered the architectures of commercial products claim that a non-branching flow of matter is a dominant flow, and consequently is a relatively stronger technical module driver, we are not prioritizing the four types of functional interactions. In total the paDSM represented 54 spatial interfaces with non-causal interactions, 29 interfaces with energy (power) transfer, 6 with material transfer, and 97 interfaces with information (signal) transfer. Some of the many dependencies, modelled as interaction strengths in the paDSM, are briefly presented in Table 2.

4. Results

Ten thousand iterations with the IGTA++ DSM-clustering algorithm grouped the 88 principal components into 20 clusters (conceptually shown in Figure 5), with each cluster having a more or less integral architecture. That is, the dependencies are large between the components within a cluster, but cluster to cluster dependencies are small. The clustering convergence rate is inversely proportional to the cluster number, i.e. the top-left clusters in Figure 5 are the most stable ones. The wall-clock time on a regular laptop was 0.88 s per iteration, adding up to two hours and 28 minutes.

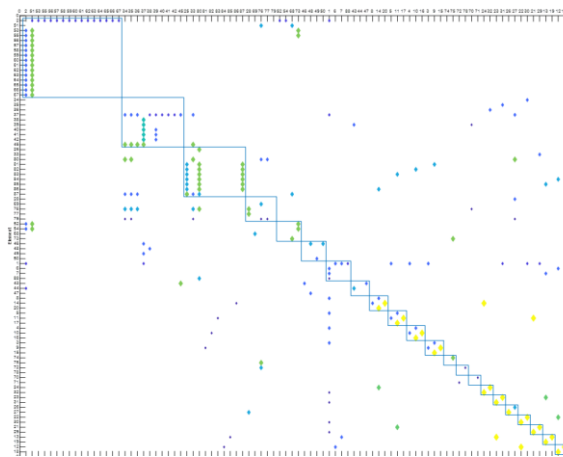


Figure 5. The 88 DSM components grouped into 20 clusters from #1 top left to #20 bottom right

The 20 clusters in Figure 5 are:

- Cluster 1: *Truck cabin body* with two of the four exterior *Camera units* (one front and one rear), all the four *Lidar, Radar* and *Sonar* sensors, and the *Object recognition unit*.
- Cluster 2: *Motor control units* for the four driven wheels, the *Batteries, Battery charging unit, the Battery voltage, Battery current* and *Battery temperature* monitoring sensors, the *Battery state analysis unit*, and the *Electric control unit*.
- Cluster 3: *Angular sensors* for the six wheels, *Traction control, ABS*, and the *ESP*.
- Cluster 4: *Steering motor controller, Active safety management, Motion planning, Power and steering management*, and the *Remote override unit*.
- Cluster 5: *Active safety analysis, Road assessment analysis unit*, and two of the four exterior *Camera units* (one front and one rear).
- Cluster 6: Main system for battery cooling and pre-heating, i.e. the *Pump, Cooling lines, Heater, and Heating lines*.
- Cluster 7: *Truck frame, Gyro, and two of the six Wheel Suspensions* (two of the driven wheels).
- Cluster 8: *Thermal management unit, Battery safety protection unit*, and the *Heat radiator*.
- Clusters 9, 10, 11 and 12: *Suspension, Wheel hub, and Friction brake* (for the two steered wheels and two of the four driven wheels).
- Cluster 13: *GPS and Position analysis unit*.
- Clusters 14 and 15: *Condition management* and *Remote state communication* in cluster 14 and *Condition monitoring* and *Condition analysis* in cluster 15.
- Clusters 16, 17, 19 and 20: *Wheel Hub reduction and Traction motor* for the four driven wheels.
- Cluster 18: *Steering motor and Steering gearbox*.

4.1. Discussion of the clustering result

Below follows a conceptual discussion of the 20 clusters proposed for the safety- and performance-aware paPDM:

- Sensor analysis units, such as Object recognition Road assessment units, are separated from the actual decision and actuation modules. Object recognition, or perception, relies on high speed and, in many cases, large bandwidth signal transfer from sensors to a perception analysis unit with embedded sensor fusion and pattern recognition software. The output from a perception module is generally a relatively small amount of data that should be accessible by a decision/management unit, which consequently, at least from performance reason, do not have to be placed tightly integrated with the perception unit.
- The object recognition (OR) unit relies on high speed transfer of large bandwidth data, especially from the cameras. From performance and safety point of view, it makes sense that all these sensors are grouped with the OR unit in a protective cabin body.
- Clustering of the six wheel suspensions seems to be unclear, they are either grouped with the frame or with a “wheel unit”. It would be more logical if the four suspension for the driven wheels would include suspension, or not, and that the individual suspensions for the two steered wheels would be included in the wheel module or not.

4.2. Interaction weight sensibility study

To study the effect from the performance and safety weights of the represented interactions on the proposed clusters, paDSM clustering with varied weights for the performance and safety complexity parameters were performed.

With a *binary interaction model*, i.e. an interface relation is given a weight of 1 for any type of intended interaction or any combination of de four potential functional relations. Here we got 21 clusters. Cluster 1 included the *Cabin body*, and all *Cameras*, *Lidars*, *Radars* and *Sonars*, as well as the *Object recognition* unit and the *Road assessment unit*. Which is similar by the more performance and safety prioritized proposal above. An observation is that generally there were no decomposition of sensing, perception, action decision, and actuation. As an example, for cluster 5 the algorithm proposed a configuration of *Condition monitoring*, *Condition analysis*, *Condition management* and *Remote communication*. Similar clustering were proposed for the wheel *Angular sensors* and the units for *ABS*, *Traction control*, and *ESP*, which also were clustered in, what seems to be, a rather random way. Furthermore, *Motor controllers* were clustered with *Electric control*, *Motion planning*, *Power and steering management*, *ESP*, and *Remote control*, but not with the *Angular wheel sensors*, nor with *ABS* or *Traction control*. Clusters 6-9 for the four driven wheels made some sense, including *Suspension*, *Wheel hub*, *Friction brake* and *Angular sensor*.

A clustering simulation with the four types of *functional interactions*, each adding a weight of 1, i.e. the interaction between a pair of interfaced system components has a total value of 1/2/3/4, depending on the number of active functional types. Proposed a clusters in between the DSM with more detailed interaction weights and the one with binary weights. That is, there were no logical decomposition of the sensors, perception, planning, and actuation system constituents.

5. Discussion

The more long-term goal of the presented research is to make a contribution to MBSE, by developing efficient and robust methods and computer-based tools to assist and potentially also automate major parts of the SE product architecting stage. This is a challenging task that requires significant effort and thorough verification as well as analysis of how it most efficiently may assist the SE process.

The main hypothesis behind the present research is that computer-based product cluster analysis benefit from a quantitative complexity measure, as well as means to represent and communicate product architecture related complexity. A complexity-based measure to quantify the strength of intended interactions is proposed and exemplified. An interesting observation is that the proposed clustering method propose a hierarchical control structure, with signal analysis, i.e. perception, clustered with the sensors, and with more loosely connected decision and planning layer that does not

have to be tightly integrated with the sensor and perception module, and a decision module that preferably is clustered with the actuators, a structure that is also proposed in e.g. (Oliviera, 2019). Furthermore, an integrated battery and battery management unit (BMU), i.e. sensors, state analysis, charging, and electric power output control, is proposed within cluster 2. This is in line with novel technological solutions to maximize battery performance and to minimize risk for fires caused by battery and/or inverter overheating. Clustering proposed a battery cooling and heating cluster separate from the battery, which is questionable.

The clustering result depends on the quality of the paDSM, i.e., that the different function carriers (components) have been decomposed deep enough, that the decomposition is balanced, i.e. similar depth all over the system, and that the interactions and their weights are reasonably represented. The sensitivities of the proposed performance and safety criticality parameter values may be analysed by changing the weights of the interactions that are judged as being most uncertain.

The presented case highlights that clustering based on technical complexity only will cause some ambiguous clusters, e.g. the six suspensions in the presented case are not clustered in a fully consistent manner. This deviation from expected result can potentially be addressed by adding the strategic module drivers as presented in (Williamsson et al., 2018), e.g. will the effect from assemblability requirements cluster the suspensions with the frame or with the wheel modules? The effects from interferences, unintended interactions, may influence the clustering result. For example, will the electro-magnetic fields generated by the motors, affect the target clusters for the control units. A method to include the effects from interferences to paDSM clustering is proposed and elaborated on in (Williamsson and Sellgren, 2019).

6. Conclusions

Refining a component-DSM into a product architecture DSM with the intended interactions represented as functional relations, i.e. spatial relation, and/or energy transfer, matter transfer, or information transfer, enables more logical clustering of a system.

It is still unclear if clustering benefit from the four types of functional relations having different virgin weights, e.g. if mass transfer could be considered more complex and thus motivating a larger relative weight value, or if this could be more effectively handled with a higher mass transfer performance criticality value.

A method to include and treat interaction performance criticality and safety criticality in the clustering simulation, has been proposed, and it has shown to provide some benefits to the proposed system cluster candidates.

The presented example shows that the influence and importance of the mechanical, electrical, and software components, will propose a reasonable multi-level architecture, when taking signal performance and safety concerns into account.

7. Future research

The clustering effects from strategic module drivers, as presented in (Williamsson et al., 2018) will be studied. Layout architecting will also be targeted, i.e., a study of the effects from interferences on the modular architecture, and the constraints imposed on the layout. How interferences affect the proposed cluster structure will be studied with the approach proposed in (Williamsson and Sellgren, 2019).

References

- Blackenfelt, M. (2001), "Managing complexity by product modularization"!, Doc. Thesis, Dept. of Machine Design, KTH Royal Institute of Technology, Stockholm, Sweden.
- Blecker T., Kersten W. and Meyer C. (2005), "Development of an approach for analyzing supply chain complexity", MPRA Paper No. 5284, posted 12 Oct 2007 UTC, <http://mpra.ub.uni-muenchen.de/5284/> Accessed 2019-10-29.
- Börjesson F. and Sellgren U. (2013), "Fast Hybrid Genetic Algorithm for Clustering Design Structure Matrix", *ASME Design Engineering Technical Conferences*, Portland, OR, Paper number DETC2013-12041, <https://doi.org/10.1115/DETC2013-12041>

- Casti J.L. (1992), "The simply complex: trendy buzzword or emerging new science", Santa Fe Institute, Santa Fe, New Mexico, USA.
- De Leeuw S., Grotenhuis R. and van Goor A.R. (2013), "Assessing complexity of supply chains: evidence from wholesalers", *International Journal of Operations & Production Management*, Vol. 33 No. 8, pp. 960-980, <https://dx.doi.org/10.1108/IJOPM-07-2012-0258>
- Dieter, E. et al. (2013), "Engineering Design", 5th edition, McGraw-Hill, New York, NY.
- Eppinger S.D. et al. (1994), "A- Moel-Based Method for Organizing Tasks in Product Development", *Research in Engineering Design*, Vol. 6 No.1, pp. 1-13.
- Eppinger, S.D. and Browning, T.R. (2012), *Design Structure Matrix Methods and Applications*. The MIT Press, Cambridge, Massachusetts.
- Erixon, G. (1998), "Modular function deployment: a method for product modularization", Doc. Thesis, Dept. of Manuf. Sys., KTH Royal Institute of Technology, Stockholm, Sweden.
- Ghosh A. et al. (2018), "Measuring the complexity of product configuration system", *Proceedings of the 20th Configuration Workshop*, Graz, Austria, pp. 61-68 (CEUR Workshop Proceedings), ISSN 1613-0073.
- Höltkä-Otto, K. (2005), "Modular Product Platform Design", Doc. Thesis, Department of Mechanical Engineering, Machine Design, Helsinki University of Technology, Helsinki, Finland.
- INCOSE (2007), "SE Vision 2020", INCOSE-TP-2004-004-02, Sep 2007.
- Mattsson S., Gullander P. and Davidsson A. (2011), "Method for Measuring Production Complexity" presented at the *International Manufacturing Conference IMC 28 - Manufacturing Sustainability*, Dublin City University, Ireland, 2011.
- Mattsson S. et al. (2014), "Comparing quantifiable methods to measure complexity in assembly", *Int. J. Manufacturing Research*, Vol. 9, No. 1, pp. 112-130. <https://dx.doi.org/10.1504/IJMR.2014.059602>
- Novak, S. and Eppinger S.D. (2001), "Sourcing By Design: Product Complexity and the Supply Chain", *Management Science*, Vol. 47, No. 1, pp. 189-204.
- Oliviera R. (2019), "Motion Planning for Heavy-Duty Vehicles", Lic. Thesis, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden.
- Patzak, G. (1982), "Systemtechnik – Planung komplexer innovativer Systeme, Grundlagen, Methoden, Techniken", Springer Verlag.
- Pugh, S. (1990), "Total design", Addison-Wesley.
- Rodriguez-Toro, C.A., Jared, G. and Swift, K. (2004), "Product-development complexity measures: a framework for proactive-dfa implementation", *DS32: Proceedings of DESIGN 2004, the 8th International Design Conference*, Dubrovnik, Croatia, pp. 483-490. ISBN 953-6313-59-6
- Sosa, M.E., Eppinger, S.D. and Rowles, C.M. (2003), "Identifying Modular and Integrative Systems and their Impact on Design Team Interactions", *ASME, Journal of Mechanical Design*, <https://dx.doi.org/10.1115/1.156074>
- Stone, R.B., Wood, K.L. and Crawford, R.H. (2000), "A heuristic method for identifying modules in product architectures", *Design Studies*, Vol. 21 No. 1, pp. 5-31.
- Suh N.P. (1999), "A theory of complexity, periodicity and the design axioms", *Research in Engineering Design*, Vol. 11 No. 2, pp. 116-132.
- Törngren M. and Sellgren U. (2019), "Complexity Challenges in Development of Cyber-Physical Systems", In: Lohstroh M., Derler P. and Sirjani M. (Eds.), *Principles of Modeling*, Springer International Publishing AG, pp. 478-503, <https://doi.org/10.1007/978-3-319-95246-8>
- Ulrich, K.T. (1995), "The role of product architecture in the manufacturing firm", *Research Policy*, Vol. 24 No. 3, pp. 419-440, [https://dx.doi.org/10.1016/0048-7333\(94\)00775-3](https://dx.doi.org/10.1016/0048-7333(94)00775-3), ISSN 0048-7333.
- Ulrich K.T. and Eppinger S.D. (2000), "Product Design and Development", second edition, McGraw Hill, New York, USA.
- VDI (2004), "Design methodology for mechatronic systems", VDI 2206, VDI Guidelines, Beuth Berlin, 2004
- Williamsson, D. and Sellgren, U. (2016), "An approach to integrated modularization", *26th CIRP Design Conf., Stockholm, Sweden*.
- Williamsson D., Sellgren U. and Söderberg A. (2018), "Product Architecture Transition in an Evolving Multi-Brand Organisation", *15th International Design Conference*, <https://doi.org/10.21278/idc.2018-0366>
- Williamsson D. and Sellgren U. (2019), "Introducing implementation dependent behavior into integrated product architecture clustering", *Proceedings of the 21th International Dependency and Structure Modeling Conference, DSM 2019*, Monterey, CA, USA, <https://doi.org/10.35199/dsm2019.9>