INTERNATIONAL DESIGN CONFERENCE - DESIGN 2010
Dubrovnik - Croatia, May 17 - 20, 2010.

MECHATRONIC SYSTEMS ENGINEERING – THEORY AND AUTOMOTIVE PRACTICE

M. Vielhaber, D. Bergsjö and A. Ćatić

Keywords: mechatronics, systems engineering, automotive industry, PLM

1. Motivation

Automotive engineering is looking back on more than 120 years of history. Most of this time was marked by mechanical design improvements. In recent years, electronics is more and more dominating the formerly mechanical cars, for now more in the sense of optimizing the mechanical functions than replacing them by new mechatronic solution principles. This is however foreseeable for the future, with completely new concepts coming up based on the ongoing electrification of the drive system. Mechanical engineering as such has a much longer history, with engineering design methodology research entering the scene around 150 years ago. Also in design methodology, mechanical thinking has been predominant over most of the time. Later developments have led to standardized mechatronics-focused process models, which are however not consistently applied in industrial practice, yet.

This paper investigates how state-of-the-art systems engineering methodology can be brought closer together with engineering practice, focusing on the example of automotive engineering. It identifies gaps and proposes steps towards a better theory/practice fit of engineering methodology. In a first step, chapters 2 and 3 analyze mechatronic systems engineering in theory and automotive engineering in practice, respectively. This is done based on observations at German and Swedish car manufacturers, workshops and discussions with scientific partners and system suppliers, and research and advanced engineering projects on engineering methodology, engineering systems, and Product Lifecycle Management (PLM). Chapter 4 describes major challenges arising from the application of mechatronic systems engineering approaches on automotive engineering. The discussion in chapter 5 will lead to the question, if future mechatronic systems engineering will require more evolutionary or more revolutionary steps in engineering methodology, followed by conclusions in chapter 6.

2. Mechatronic Systems Engineering

This chapter describes and analyses mainly theoretical approaches to engineering design of mechatronic products. Taking a look at the history of a discipline, first, may explain, why it is as it is today, and where paradigms may be hardened and therefore to be addressed with special attention.

Mechanical engineering is looking back on thousands of years of history. First theoretical works on engineering methodology can be found in the 19th century, leading Central Europe to a more scientific approach to engineering design compared to more applied and experienced-based concepts in Anglo-American countries. The scientification also gave the starting signal for a dispute between theory and practice, which is still ongoing today, being regularly discussed both in design departments and on scientific conferences. Multiple process models have been developed to describe engineering design in an applicable way, and still are. Common aspects have generally been the proposal of step-by-step guidelines along the process and a special attention on early, conceptual process phases. The German standard VDI 2221 [VDI, 1993] offered an attempt to deliver a consolidated view. In the 1960's, systems engineering [e.g., Daenzer, 1997] came up as a holistic approach to problem solving. In recent years, this paradigm got more concretized and applied on mechatronic products with VDI 2206 [VDI 2004], which is today often used as a basis for applied engineering design methodologies.

Looking back on this history, mechatronics is a comparably new aspect of engineering design. The electronification of formerly mechanical product solutions is however growing rapidly, with penetration levels varying by product, reaching around 40% added value in automotive applications, and the proportion is increasing. Engineering methodology couldn't keep pace with this development. VDI 2206 also tried to bring closer together the independently developed engineering methodologies for electronic, often software-based products and the traditional mechanical ones, both domains however bringing along long traditions of own and uncoordinated processes, methods, and tools. Figure 4 shows the macroscopic process model of VDI 2206, which is supplemented by problem solving methods on a microscopic level. On a first glance, the presented process is geared to the so-called V-model of software engineering. It applies this inflationary-used visualization format on the sequential process steps of VDI 2221. Both the earlier conceptual phases and the later validation phases are advanced to an integrated system level, leaving the detailed design on unintegrated domain-specific process streams. Another difference to its predecessors is that VDI 2206 dissociates itself from the former rigidity and sequentiality of process steps and associated methods. It doesn't raise the claim anymore to provide a detailed course of action, but refers to method toolkits for on-demand application, instead, thoroughly detailed however only for the mechanical part. Thereby it delivers more a frame of methods than an integrated methodology. It is herewith in-sync with other contemporary approaches, such as [Lindemann 2009]. Above points could be objected to the model of VDI 2206 – nevertheless it offers a first systematic shot for an application-near, mechatronics-focused engineering methodology, at least. In engineering practice, VDI 2206-like V-models are referred to quite often. Their real application is however still fragmentary. Gaps can mainly be found in the early system design phase, where an integrated mechatronic functional view is still lacking, in the later system integration and validation phase, where the domain results should be systematically brought back together, and in the interlinking of the domain-specific detailing steps, where independent and partially incompatible processes, methods, product data models and IT solutions still dominate.

Major trends currently to be seen in engineering methodology are related to enhancing IT support across all phases of the V-model, advancements on mechatronic integration on process and method level, and the above mentioned modularization and thereby flexibilization of the process models.

3. Automotive Engineering

This chapter will provide an overview of automotive engineering and the ongoing mechatronic penetration of the products, setting it also in relation to other kinds of industries, see figure 3.

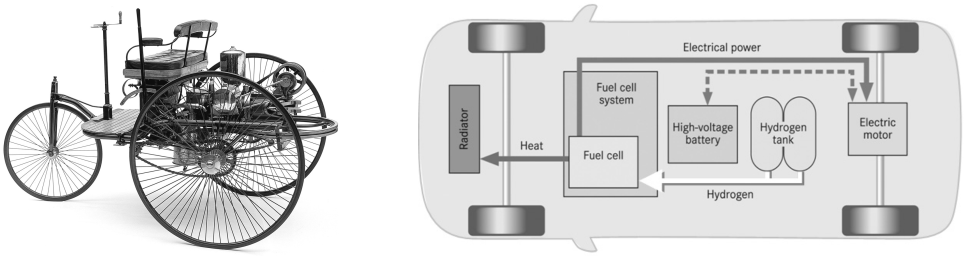

Figure 1. Car concepts 120 years ago and today [www.media.daimler.com]

Automotive engineering took its beginning in 1886 with the first patented vehicle of Gottlieb Daimler (figure 1). A majority of revolutional automotive key developments on a conceptual level go back on Daimler and his contemporaries and happened within the first 20 years of automotive history. Generally, these inventions were more driven by practical demands than by methodical derivation. Since then, automotive engineering is marked by mainly evolutionary concept improvements – VDI 2206 gives an example with the evolution of the brake system. What is also pointed up by this example is the entering of electronic support, control and optimization functions in recent years. Real replacements of mechanical main functions by mechatronic solution principles in a broader application are however still to come. They can be foreseen up to the complete car concept level based on the ongoing electrification of the drive system (see Figure 1 right – with mechanics now just in shaded gray).

3.1 State-of-the-art & trends

Going from the product level to the processes and methods applied in automotive development, no fundamental adaptations have happened to the traditional mechanics-based approaches, yet. Practical automotive engineering is often still executed as the development and later assembly of a bunch of physical parts. In fact, electronic and software development steps have entered the process models of the overall development systems, and of course single dedicated electronic and software engineering methods are applied. But still, no integrated mechatronic development organization has been established, yet. No integrated mechatronic product data models and IT tools have been implemented. And domain-spanning coordination mainly happens on a people basis, still. In recent years, quality concerns originating from weak cross-domain processes and a missing method integration of electronics, software and mechanics have lead to major initiatives to manage and control this heterogeneous process and method landscape. Although successful, it may be questioned if this may be just an intermediate step on the way towards a real mechatronic development setup. Major trends to be seen in automotive engineering are the further ongoing electrification of components, functions and at the end the complete car concept, asking for adequate support from the method and process side. Increasing complexity and multiplicity of variants have been a topic for a while, and still are. Last but not least it has to be recognized, that cars are no longer bought just because of them fulfilling basic technical functions, as this was for long over automotive history. Today's customer expectations go far beyond with design and reliability aspects often being mentioned first in market surveys.

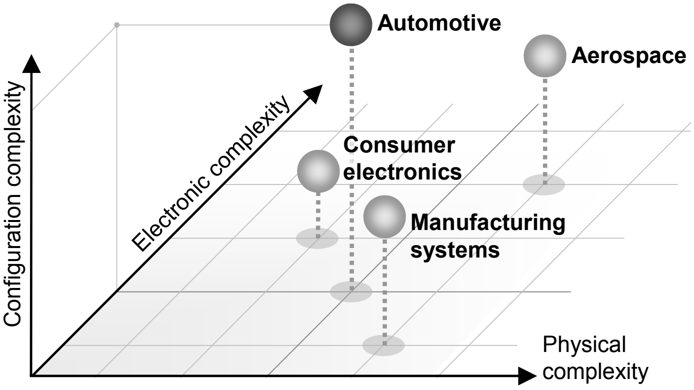

Figure 2. Complexity comparison by industry

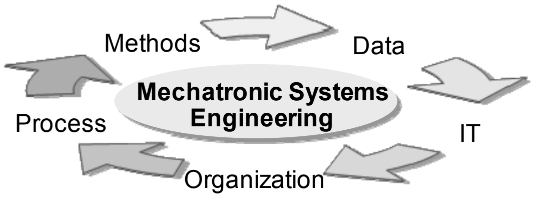

Figure 3. Integration aspects of mechatronic systems engineering

3.2 Comparison with other industries

One may ask why mechatronics engineering is still to be discussed as a vast variety of well-engineered mechatronic products is well established on the market. In fact, this may be the case for some industries and products – digital cameras or industrial robots being just two examples. A generalization however requires a more distinctive view. Figure 2 categorizes products by three dimensions of complexity. First, their physical structure may be more or less complex. Second, their electronic layout may be more or less sophisticated. And third, what is often not adequately considered, their configuration and variance level could be more or less high. Looking at automotive products, compared with others, both physical and electronic complexity range in the midfield – for sure more complex solutions can be found both on the mechanical and on the electronic side. But, what makes automotive products distinct is the combination with an extraordinary configuration complexity. The number of configurations per product is still increasing based on market demands, with both mechanical and electrical components along with their version and variant multiplicity to be kept in-sync. This combination makes mechatronics an outstanding challenge in automotive engineering, making concepts potentially already applied in other industries not directly transferable.

4. Challenges for Automotive Systems Engineering

This chapter will analyze how mechatronic systems engineering methodology from a theoretical viewpoint can be brought closer together with automotive engineering application. From an overall perspective, automotive systems engineering still has some way to go from its former mechanical part oriented setup towards an integrated mechatronic system oriented concept. In [Vielhaber 2004], one step on this way was described with the move from the traditional part orientation towards an assembly oriented engineering setup, built on a combination of process, methods, data, IT system and organization oriented building blocks. This concept has now to be carried a step further to the systems level. To be successful in concept and application, automotive systems engineering will have to consider similar aspects of process, method, data, IT system and organization integration, see Figure 3.

4.1 Process Integration

To be successful, all building blocks have to be addressed in combination. In the following, process integration will be dealt with first, as it appears to having developed to the highest maturity level. At least, this is true from a macroscopic view. Stage-gate process models which reflect the general V-model process are common practice in automotive engineering. Their phases are structured by quality gates to be passed with defined maturity criteria. Figure 4 relates these gates to the respective steps of VDI 2206. It is obvious in this example that the V-model's brake down from system to component level is well reflected. The first conceptual steps are treated at an integrated, domain-spanning level. Going to components, work splits up into domain-specific traditional paths. In the later validation phases, the milestones show the successive re-integration towards the full system level, again.
This theoretical process is more or less stuck to depending on company specifics and product characteristics. For 'new development'-style tasks like car (body) projects manufacturers focus heavily on the descriptions and fulfillment of requirements, following the V-model. For projects of more 'evolutionary development'-character such as engine projects tasks are more focused on gradually refining the product or system over time. From a mechatronic perspective evolutionary development has resulted in electrical and software functions having been added successively to former mechanical products (e.g. the engine). In this case, the process is less rigid and less prescriptive, and testing the product and the entire system becomes more important and time consuming.

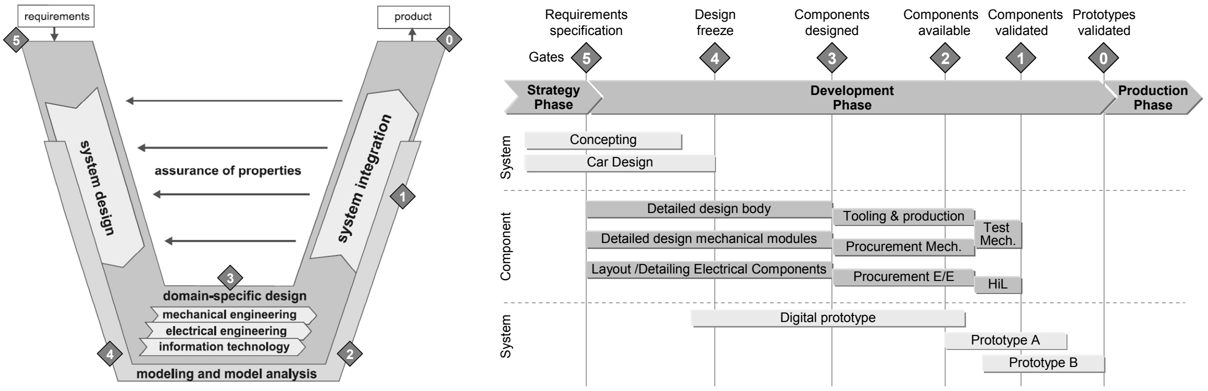

Figure 4. Stage-gate process model for automotive engineering (left: [VDI 2004])

Significant gaps in process integration are not obvious on this high level of the process models. They get more eminent coming down to the methods, tools and data supporting the respective steps. As long as these are not on an adequate integration level, cross-domain integration will only happen at the respective milestones to be passed together, if at all. In the meantime, inconsistencies may develop which lead to increased integration efforts in the following and thereby potential losses in time, quality and money. As electronic components (with embedded control software) are to a larger extent supplied by suppliers with wide system solution responsibility while mechanical parts are supplied 'as designed by OEM', this poses a barrier in the possibility to integrate the processes not only due to cross-domain issues but also due to inter-company issues. Such issues not only include asynchronization between supplier processes and OEM processes but also among supplied components which require suppliers to be synchronized with each other.

4.2 Methods Integration

A multiplicity of methods is applied all over the automotive product creation process. This is the case for all engineering domains involved, with each of them bringing along its own, independently developed and logical-as-such set of methods. Really integrated mechatronic methods however are only rarely to be found. Even if common IT systems are applied, domain-specific methods still dominate. And also similar intended purposes may be dealt with differently and potentially inconsistently across domains, leading to significant problems in re-integration.

In the following, this will be elaborated on two examples from automotive engineering, which reside in different phases of the V-model process. First, in the conceptual phases, requirement and function driven design is a topic in focus in current automotive projects, with mechatronic integration however being just at the beginning. And second, most prominently for automotive engineering based on the complexity analysis in figure 2, configuration management across domains poses an elementary challenge all along the V-model, with special emphasis on the detailing and validation phases.

Looking at contemporary design methodologies, requirements are the key inputs driving and controlling the complete engineering process. Furthermore, an initial functional concept phase is seen as eminent for the finding of optimal solution principles. Both points are also recognized in automotive engineering. The requirements management process is dominating the initial phases; it is however only loosely integrated with the method and tool landscape of the successive steps in design and validation. Efforts are on the way to improve this integration. Linking requirements to the physical products via PLM solutions is one common solution approach. Questions still to be solved as preconditions are however the optimized fragmentation level of requirements formerly collected within text-based specifications, their consistent quantification to make them interpretable for design tools, and their integration depth into tools to keep their complexity level manageable. A functional modeling step is well-established for the electronics and software domains, but it is not yet consistently thought through for the mechanics part. At least, functional modeling is seen as an enabler for an integrated, domain-spanning conceptual product view, and thereby as a basis for optimized function-fulfilling designs as well as function-based validation tests. Also for functional structures, PLM systems seem to be the solution of choice, and functional modeling tools are available on the market, with the respective system implementations not posing big technical issues. Questions still to be solved are however even more fundamental than in the requirements area: what should be the appropriate level of functional descriptions, will it be possible or even desirable to describe the complete mechanical product part on a functional level, and what should be the functional modeling depth to keep the complexity level manageable.

Once these fundamental issues are solved, requirements, functions, and also links between them and the resulting physical products can be modeled in an explicit way (figure 5 left). Based on the thinking that just making requirements and functions explicit does not increase the complexity existing anyway, an all-integrated system model may be compiled to be used all along the process. Such approaches are promoted by PLM system suppliers, and tried to be adapted for automotive application [Lamberti 2009]. They may look tempting on a pure methodical or single IT systems level. Modeling the entire universe of requirements, functions, geometries and all their interrelationships will however fail in reality due to an overwhelming complexity of the resulting product model. Furthermore, major challenges such approaches pose lay in the organization and overall IT concept adaptations they imply. Also, making complexity explicit may make it also rigid and prevent the flexibility necessary for an inventive product creation process. Thus the focus has to be on methodically keeping a product model flexible, e.g. through providing more modeling tool kits than sophisticated modeling templates, and thereby keeping the complexity on a level manageable by the user.

Allocation of functions in a function oriented development process is a major challenge for mechatronics development. The relationship between the function and the component or system needs to be clearly defined. As electrical functions are abstract and realized first when initiated by the operator they are not easily modeled and connected to the physical world. An operator function could in this context be defined as an electrical-related service that creates explicit benefit for the operator, whereas a physical function is closely related to the physical system (see Figure 5 right).

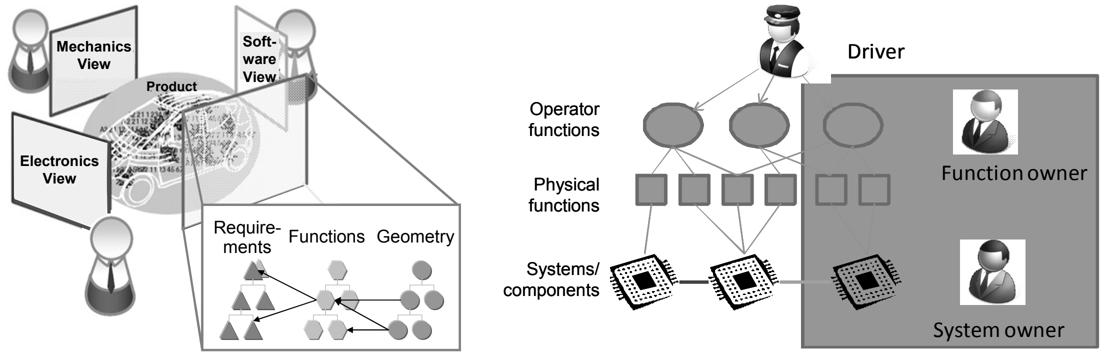

Figure 5. Mechatronic product model mapping requirements, functions, and components

Configuration management covers the handling of both variants and versions of product information all along the product lifecycle. It is a complex process even in a single-domain environment. The design cycle of a car is 3-5 years. In mechanical engineering, over this period of time, parts typically go through several iterations until they reach a sufficient maturity. In each prototype built, different versions of the same part may be used. Designers have to be able to manage all kinds of such configurations to create and validate their designs. With thousands of parts comprising a car, it is obvious how sophisticated the configuration mechanisms must be to cope with these needs. In a multi-domain environment, differences in the processes between the disciplines are additionally problematic – e.g. in software and electronics development, different lifecycles, prototyping mechanisms, configuration logics, and data schemes are used than in the mechanics domain. The lifecycle for an information appliance is generally less than a year. Bringing this together with the mechanics domain, in order to, e. g., ensure correct combinations of software, control unit, and mechanical part versions, all domains' configuration mechanisms have to be enabled to communicate with each other.

What's required to improve on these issues is in a first step a common understanding of configuration management terms and basic concepts across domains. Second, common base methods have to be defined on this, thereby building the base for an integrated system solution. And third, the domain-specific implementations – which will still have to be different to support the different process demands described above – have to be kept in-sync, at least at commonly defined gates, see figure 6 (left). This sounding easy, it has to be considered that domain-specific methods have long traditions and are applied in every-day practice. A transition concept towards an integrated configuration setup has therefore to be an elementary concept component. An important aspect when implementing such a synchronization model is that it has to play well with the vast amount of suppliers involved in this work. A common issue encountered in practice is that as change issues result after such synchronization points, especially during verification and integration, the times required for a redesign effort vary quite heavily depending on each supplier's prioritization and internal plans.

Looking at both examples presented, it's obvious that a solid foundation on the data and successively the IT systems level is required to support the methodical issues presented.

4.3 Data Integration

Integration on the data level has to be an integral part of mechatronic systems engineering. The methodical concepts described above require a common system-spanning basis on the data level. Such an integrated data concept has to comprise both conceptual product data objects, such as requirements, functions and the links between them and the physical product, and configuration information such as object variants and versions, at least. More comprehensively, an integrated, however flexible and extendible core data model for all relevant system data needs to be established. Such a data model may be taken over from IT system suppliers for the scopes of their respective systems. An all-integrated PLM system is however just fiction. Current potentially domain-specific data objects all have their own legacy, thereby making a consolidation difficult and elaborate. Projects having tried to establish such an integrated data model on a complex product level (such as an automobile) prove the complexity of this undertaking, and the challenge will be multiplied by the described extension to a systems level. What makes systems data management especially difficult is the accentuated role of links between the respective data objects. These links have to ensure the consistency and traceability of all relations between requirements, functions and geometries as well as across all variants and versions. Again, IT system suppliers are promoting such integrated data models, but incorporation of linked objects residing out of the limited application scopes is not sufficiently considered, this however being the reality – it's an academic standpoint neglecting the traditionally established as-is systems and therefore claiming to apply an all-in-one data model in an all-in-one IT system. From a scientific viewpoint, the topic of link management has been addressed, already. E.g., [Zimmermann, 2005] and [Burr, 2006] describe approaches to extract links from the domains and manage them by separate objects in a distinguished, domain-independent location, see figure 6. Significant productive applications have however not developed out of this, yet.

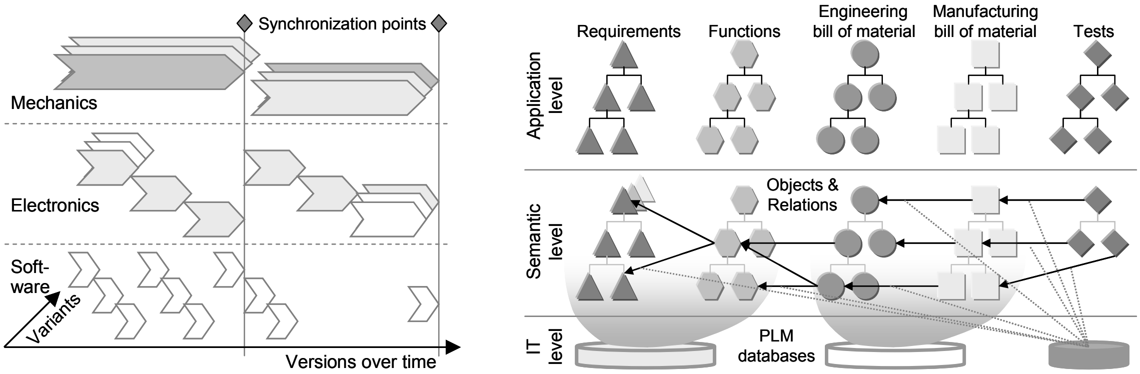

Figure 6. Synchronization of configurations across domains (left) as one aspect of an integrated data model for mechatronic systems engineering (right, similar to [Burr 2006])

To conclude, a system-wide core data model seems inevitable for system-level engineering, however difficult to establish. As a consequence, it has to be kept as comprehensive as necessary, but as lean as possible. It has to be examined, what the information objects are required for inter-domain cooperation and integration. This especially includes the definition of the relations between these objects. Then, syntax, semantics and behavior of these objects and relations have to be agreed upon together with the integration patterns, how they are to be shared or synchronized across domains. The strive towards increased product modularization may facilitate such an approach by allowing the creation of data models locally for each module and thereby making it easier and more manageable to identify objects and integration patterns needed. This would also support a subsequent integration of necessary IT systems and tools, relieving the pressure from having to integrate everything with everything, and allowing focusing on relevant information flows and important interfaces.

With all this as a basis, it can then be decided how this data can be implemented in an IT landscape, what will be in focus of the following chapter.

4.4 IT Systems Integration

IT concepts and solutions are being discussed ever since the introduction of the first supporting IT tools in engineering, both in design departments and on scientific conferences. They often get in focus as they are pushed by IT vendors, promoting IT as the main enabler for efficient engineering. Both mechatronics and systems engineering are buzzwords stressed quite often in this context. This chapter will set IT systems in relation to the other dimensions previously discussed. Looking at the methodical and data issues presented, it looks tempting to 'just' implement an all-in-one IT system to cover the entire scope of engineering, an approach often focused by IT system suppliers. This would mean putting everything from requirements over geometry till validation and production data into one single application, at least on the data management level. This might be realistic in academic setups or for small engineering startups. In reality, diversity and flexibility demands of the domains involved force the use of best-in-class applications best-supporting the domain-specific process and method landscape. This situation is also reflected in typical legacy IT landscapes of automotive companies, with different domains supported by different IT systems with different underlying methods.

The solution for mechatronic systems engineering in automotive application has to be somewhere in between. Various such approaches have been discussed over the previous years, often under the flag of PLM (e.g. [Bergsjö, 2007]). It's foreseeable today that the overall IT support for the product creation process will have to follow a flexible and modular layout with clearly defined system layers and roles – not meaning that on a domain basis different system roles may not be aggregated in one actual system. In addition, it has to be kept in mind that cross-domain mechatronics integration builds just one axis of the general three-dimensional PLM framework of cross-domain, cross-lifecycle, and cross-enterprise integration [ibid.]. Such solutions are often envisioned to be realized via service-oriented architectures (SOA). Thereby, optimal domain-spanning interlinking and domain-specific flexibility could be combined. Several preconditions have however to be fulfilled: the common understanding of basic methods based on the common mechatronic core data model as already discussed above, and the definition of interaction patterns and suitable services between the disciplines concerned. Following the reasoning about optimizing the information flows according to product modules, creating local clusters of integrated IT-tools could facilitate an implementation effort which otherwise would strive towards the utopia of an all-embracing product wide, or even business wide, information model before any systems integration could be realized.

Finally, cross-discipline technical solutions like SOA can only be made work based on suitable organizational setups, which will be discussed in the next chapter.

4.5 Organizational Aspects

The organizational dimension may appear fuzzier than the other ones and therefore less important to be addressed. For success in application, it is however at least equally relevant: none of the formerly discussed approaches can be successful without being adequately reflected on the organizational level. When it comes to integration questions, one general solution often mentioned is 'frontloading', meaning to aggregate all relevant work in the early phases of the process. This also sounds tempting for mechatronics integration. The challenge is however to keep the resulting early-phase complexity manageable by adequate organizational setups.

Organizational aspects will have to be addressed on both a macroscopic (e.g., department) and a microscopic (e.g., role) level. On the macro level, the mechatronic disciplines generally reside in independently grown, still unintegrated organization units. Coordination between them happens on a workgroup level – VDI2206 proposes interdisciplinary teams as a key organizational measure. For higher-level mechatronic integration, it cannot be the solution to just merge departments together, rather new organizational layouts may have to be thought of. This could be, according to the V-model process, a split into common conceptual and additional detailing organization units, thereby separating between the system level and the component breakdown, see figure 7(a). This thinking is reflected by modularization strategies currently followed in the automotive industry. The system (i.e. car project) level organization unit stays on an overall product level, making use of highly mature component modules provided by to a large extent independent, component level organizations units. Focus of the former organization unit is the concepting, component adaption, and validation of the overall product, whereas the latter runs through the complete development cycle for a limited component scope, thereby acting as a supplier to the system level. As a drawback, this may force a component-oriented, bottom-up approach, which would contradict the intended system-orientation.

A different option would be to split the development work along maturity levels, see figure 7 (b) – also VDI 2206 proposes recursive run-throughs of the V-model process. Thus, a first V-cycle covers the earlier, pre-development stages, then being of a 'new development' character, and leads to a medium, still flexible maturity level, represented e.g. by system-level product templates. The second V would be of a more 'variation development' character, adapting the input from the first one and bringing it to a production-ready level of maturity. A combination of both splits is proposed in the lean product development paradigm originating from Toyota [Kennedy 2008]. This approach separates product development into two value streams – a product value stream (i.e. car projects) and a knowledge value stream (i.e. pre-development of sub-systems). The project organization prioritizes a set of key product characteristics, e.g. 'silence', and points out the car sub-systems to be focused, e.g. engine, exhaust and body. These sub-systems are then responsible to find their major contributors to noise, for the engine it may be vibration. Not all sub-systems are chosen due to the project budget, so the rest will take whatever they have that is mature enough for detail design. The selected sub-systems will make an advancement of their particular scope within the project. The knowledge gained during the advancement is captured and documented as result of the knowledge value stream. The solution can then be reused as mature in other projects where some other sub-system is chosen to be advanced. Sub-systems could also be advanced outside projects, but this would put high demands on the governance in order to direct these efforts towards characteristics which are sought for by customers. Whatever the organizational split will be, it will require strong top-down governance to coordinate roles, methods, functions and finally IT solutions across the domains and resulting organization units.

Similar thoughts have to be made on the micro level of the individual designer and the respective development roles. Also on this level, pure concentration of work and responsibility on a single role through frontloading may quickly lead to overloading and overstraining it. A solution will have to redefine and split roles, away from traditional component responsibilities and towards integration responsibilities, as already proposed in [Vielhaber 2004].

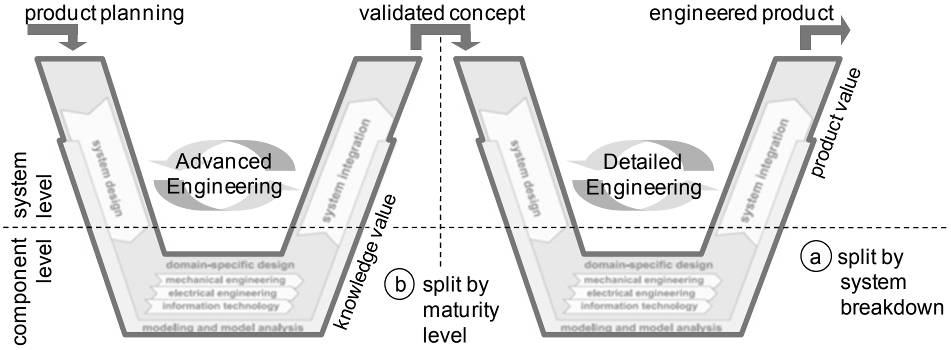

Figure 7. Organizational split options within the V-model process (V-models: [VDI 2004])

Also the proposed separation of function and system level has to be reflected in the organization, as function owners are appointed apart from the already existing component order. From an organizational perspective the function owner generally has a weaker position, since the function itself is abstract and difficult to assess. A transfer of 'power' from the component towards the function owner is needed; since the function is closer to the end user this could potentially improve his experience and satisfaction.

5. Discussion

The analysis of the five dimensions of automotive systems engineering – process, method, data, IT system and organization integration – show that evolutionary further development of each dimension alone will not lead to optimal mechatronic engineering solutions. For all dimensions, such single solution approaches have been presented. To deliver applicative value, they have to be addressed in combination. Where applicable, main interdependencies have been pointed out.

Approaches currently found however often show a unidirectional focus, only: VDI 2206 focuses on the overall methodology (process dimension); contemporary engineering literature such as [Lindemann 2009] often concentrates on the provision of method toolkits (method dimension); and IT system suppliers tend by nature to focus on the data and IT dimensions. Finally, within the five dimensions presented, organization is the one omitted the most, although it often shows to be the most significant one when it comes to implementations within any of the other four dimensions.

Another aspect not addressed in this paper comes to the fore when looking at mechatronics literature. This is highly dominated by a product-oriented view, focusing on the description of mechatronic components like sensors, actors and control elements. This view is however neither integrated with the traditional mechanical machine element view nor with the other dimensions discussed above.

One may therefore conclude that omission of the integrated view promoted in this paper is at least one reason for many contemporary approaches not to make it to a high acceptance level in application. A successful (in the sense of applicable) future mechatronic systems engineering methodology will have to deliver such an integrated view. It will have to combine evolutionary further developments in all relevant areas to a new and consistent overall approach in order to optimize quality, effort and costs through optimized mechatronic integration.

Further conclusions can be drawn for engineering education. Here, the integrated view on the process model (or methodology), on methods, data, IT and organizational aspects as well as on 'mechatronic machine elements' has to be reflected in the curriculums for future mechatronic engineers.

6. Conclusion

Mechatronics may be the biggest challenge so far in the history of automotive engineering. Its significance is assessed similarly across companies; realization approaches may however differ – requiring great prudence regarding company culture and legacy. Although widely discussed, no comprehensive mechatronic systems engineering methodology, which has proven to be successful in theory, education and, at least from an industrial standpoint most important, application, has been described, yet.

In this paper, a consolidated approach was proposed with aspects of process, method, data, IT and organization integration, capable to bring state-of-the-art systems engineering approaches closer together with engineering practice. Open questions however remain. Follow-up work will therefore have to develop a methodology description both teachable and applicable. In a next step, theses for a future mechatronic systems engineering have to be detailed, generalized from the automotive to a universal level and validated on examples from industrial practice. This will then help to answer, how far a just evolutionary redesign of traditional approaches can be capable to cope with the challenges of mechatronic systems engineering, or if a more revolutionary approach may be still to come.

References

Bergsjö, D. et al., "Product Lifecycle Management for Cross-x Engineering Design", proceedings of International Conference on Engineering Design - ICED 07, Paris, 28.-31.08.2007, pp. 469-470, No. 452.

Burr, H., Eigner, M., Vielhaber, M., "Product Structuring for Cross-x PDM", Proceedings of International Design Conference - DESIGN 2006, Dubrovnik, 15.-18.05.2006, pp. 655-664.

Daenzer, W. F., Huber, F (Ed.), "Systems Engineering", 9th ed., Verlag Industrielle Organisation, Zurich, 1997.

Kennedy, M., Harmon, K., Minnock, E., "Ready, Set, Dominate", The Oaklea Press, Richmond, VA, USA, 2008.

Lamberti, R., Winterstein, R., Sappin, O., "Characteristics of a Future Mechatronic Product Creation Process in the Automobile Industy", ProSTEP iViP Symposium 2009, Berlin, 12.-13.05.2009.

Lindemann, U., "Methodische Entwicklung technischer Produkte", 3d edition, Springer, Berlin, 2009.

VDI guideline 2206: "Design Methodology for Mechatronic Systems", VDI-Verlag, Düsseldorf, 06/2004.

VDI guideline 2221: "Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte", VDI-Verlag, Düsseldorf, 05/1993.

Vielhaber, M. et al, "Assembly-oriented Design in Automotive Engineering", proceedings of International Design Conference - DESIGN 2004, Dubrovnik, 18.-21.05.2004, pp. 539-546.

Zimmermann, J. U., "Informational Integration of Product Development Software in the Automotive Industry - The ULEO Approach", PhD thesis, University of Twente, 2005.

Dr. Michael Vielhaber
Daimler AG – Group Research and Advanced Engineering
89013 Ulm, Germany
Telephone: +49-160-862 8153
Email: michael.vielhaber@daimler.com