INTERNATIONAL DESIGN CONFERENCE - DESIGN 2012
Dubrovnik - Croatia, May 21 - 24, 2012.

TOWARD SIMULATION-BASED MECHATRONIC DESIGN

F. Dohr and M. Vielhaber

Keywords: mechatronics, mechatronic design, simulation-based design, modeling and simulation

1. Introduction

Mechatronics describes the synergetic combination of the traditional engineering disciplines mechanics, electronics and computer science which enables innovative products. This is one of the reasons why today mechatronics has become a part of almost every technical product and is becoming increasingly important. This leads to a new level of complexity regarding both the number of parts and the different domains involved. At the same time globalization forces companies to reduce time to market and development cost while quality demands actually increase. As indicated by the growing number of product recalls, traditional engineering methods already cannot satisfy the customer needs and the specific aspects of mechatronics any more.
One promising way to close this gap is model- and simulation-based design. Of course simulation has been a part of engineering design for years and offers very powerful and mature techniques. But essentially it is used only as a supporting tool and as a consequence the full potential is not exploited. In this case simulation is part of computer-aided product development where validation and verification are mostly based on the use of physical prototypes. Yet as illustrated in figure 1 those traditional methodologies fail to cope with the anticipated increasing degree of product complexity and innovation – which is observable even today. However, when simulation is regarded not only as a tool but as a new design paradigm, it offers the possibility to reduce time and costs in design processes and thereby the possibility to handle today's challenges.

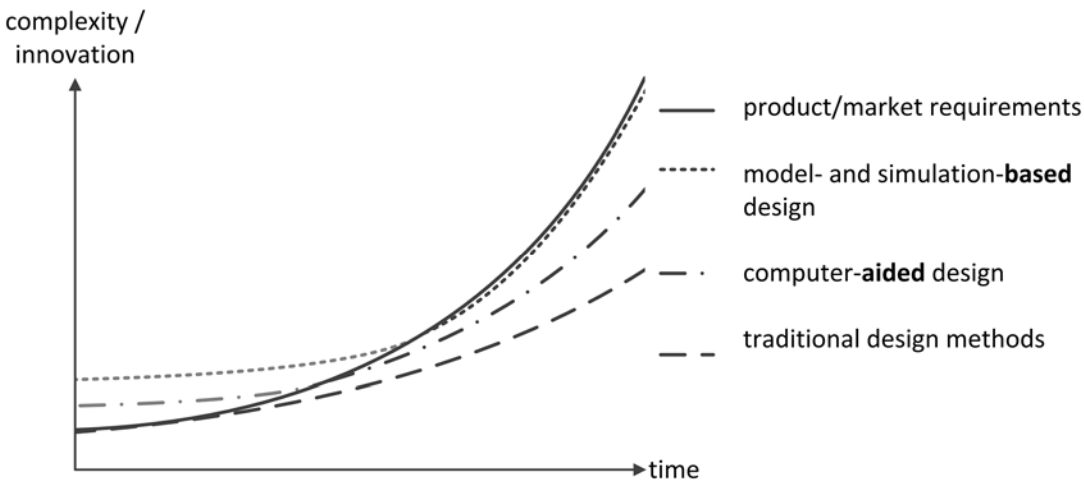

Figure 1. Fulfillment of market requirements through different design paradigms [Dohr 2011]

This means that verification of the fulfillment of requirements is not limited to a single phase at the end of the development process but rather spread throughout the process as a continuous action. As shown in figure 2 this leads to higher development effort (e.g. time, money or personnel) in early phases. Yet this is compensated for by the absence of a second peak of effort caused by late failure detection through testing in traditional design methods. Additionally, system knowledge in early phases is supported by simulation-based engineering – in contrast to traditional design where knowledge is obtained at a later stage due to prototyping. This leads to more reasoned design decisions and thereby reduces system failures. As a consequence the system achieves higher maturity in early phases.
In spite of this, there is currently no methodology taking full advantage of the potential of modeling and simulation.

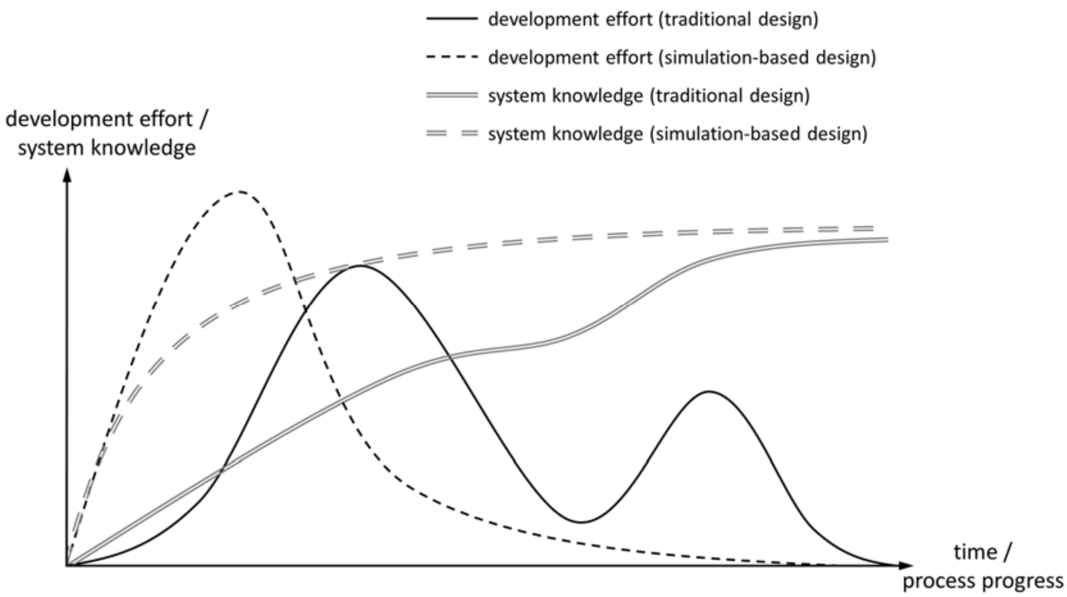

Figure 2. Comparison of development effort and system knowledge

The first part of this paper contains an analysis of current design methodologies concerning the use and integration of modeling and simulation. Based on this, questions and requirements on how to arrive at a simulation-based design process are derived in the second part. Finally, based on the previous chapter, a framework for simulation-based mechatronic design is introduced.

2. Analysis of current design methodologies

In the following section design methodologies are analyzed regarding both the use and usability of model- and simulation-based design. Since mechatronics is a multidisciplinary field, methodologies from mechanics, electronics, software and mechatronics itself are discussed. Within the scope of this paper only the most popular methodologies are described as these already highlight the current problems and challenges.

2.1 Aspects to be considered

The analysis focuses primarily on four aspects:

- Iterations
  When talking about simulation-based design, iterations become an important aspect. Traditional procedures verify specifications at the end of a process cycle through physical prototypes. This leads to macro-iterations where a complete cycle has to be repeated. By using simulation throughout the process those macro-iterations are substituted by various micro-iterations. This means that there is continuous validation and verification during the entire

process, resulting in a much more mature product at the end of a cycle. This different approach to iterations has to be considered in a methodology.

- Testing and prototyping
  Validation of concepts and verification of specifications can be done in two ways; by physical or by virtual prototypes. Testing and prototyping often are not explicitly taken into account in design methodologies although they have significant impact on process operation and product quality. Especially when it comes to simulation it is important to provide guidance where and when to test and how to handle new information.
- Use of modeling and simulation techniques
  While the preceding aspect also takes into account physical testing methods, this aspect focusses solely on the use and usability of modeling and simulation. Since there is a myriad of sophisticated tools and techniques, the analysis aspect also takes into account the use of modeling and simulation as a tool to support traditional activities.
- Interdisciplinarity
  As previously mentioned, mechatronics is an interdisciplinary field. Each of the disciplines has developed its own procedure models and utilizes them successfully. But synergy only comes into effect when different domains cooperate and share their knowledge and information. Hence a combination of individual procedure models is not sufficient for mechatronic product development – instead the integration of different interests is needed.

2.2 Mechanics

In mechanical engineering development methodologies have a long tradition. Numerous methodologies exist which often focus on only one particular topic or case. Below two of these methodologies are analyzed representatively.

2.2.1 VDI 2221

The guideline VDI 2221 [VDI 1993] deals with the development and design of technical systems and products, with special focus on mechanical engineering. It divides the development process into four phases plus "further realization". Although it is mentioned several times that the procedure is not strictly linear, the phases are often executed sequentially and iterations tend to cover all four phases. Furthermore there is no criteria when and how to return to earlier phases.

Regarding testing and prototyping, there is no discussion of these topics in the VDI guideline. Hence this part of the development process is subsumed under the point "further realization". As a consequence there is no information on how to validate a concept or how to verify the fulfillment of requirements.

Due to the date of publication of the guideline the procedure is mostly tailored for traditional manual approaches. Indeed it is stated that the methodology is suited for the use of computer-aided technologies but there is no guidance on how these are to be applied. Apart from this, it only focuses on the shape- and geometry-oriented digital mockup (DMU). But above all, mechatronic development has to take into account functional aspects which are summarized in the digital functional mockup (FMU).

Although VDI 2221 is meant to provide a generally valid procedure it focuses on mechanical engineering. Accordingly the issue of interdisciplinarity is only considered in terms of parallel but independent development tasks.

Overall, VDI 2221 has essential weaknesses with regard to the aspects discussed in chapter 2.1.

2.2.2 "Münchener Vorgehensmodell"

Lindemann [Lindemann 2007] developed a process model which distinguishes itself from common linear models by a network structure. This allows switching dynamically between different phases which in turn supports micro-iterations. Since analysis is part of a single step – i.e. "determine properties" – there is no continuous validation and verification. Within this step different methods are discussed for determining which properties to analyze, how to analyze them and how to handle analysis results. There is no concrete distinction between physical testing and simulation and since it is a generic model there is no concrete discussion of simulation technologies. Furthermore there is no guidance on how to cope with interdisciplinary teams and products.

2.3 Electronics

Contrary to mechanical engineering where design methodology has a long history and is a wide-spread field of research, explicit methodologies for the development of electronic systems are very rare.

2.3.1 Siegl

Siegl [Siegl 2005] divides the development process of electronic systems into six phases (conceptual phase, detailed design, physical design, sample production, module and system test, pilot run). Testing plays an important role in electronics. After every phase there is a testing or analysis step to verify the fulfillment of requirements. Furthermore, how verification can be done and what is needed for this is defined after the conceptual design phase. Most of the analysis steps – especially in early phases – are done by simulation. Since this procedure only deals with electronics interdisciplinarity is not covered.

2.3.2 Gausemeier

Gausemeier [Gausemeier 2006] extends the common Gajski diagram by combining the three domains (behavioral, structural and layout domain) and the abstraction levels into a phase model. Iterations are not explicitly covered but since it is a linear model, most of them encompass the complete cycle. Within the different phases there are concrete tasks including different kinds of simulations. But more concrete guidance on how to handle simulations and results is not provided. As the model focuses on the development of digital electronic devices and micro-electronics, interdisciplinarity is not covered.

2.4 Software design

In software design there is a vast number of procedure models. Often these are company- or even team-specific. One of the most common ones is the V-model. As this is the basis for the guideline VDI 2206 [VDI 2004] and thereby very similar it is discussed in chapter 2.5.

2.4.1 Spiral model

The spiral model developed by Boehm [Boehm 1986] enhances the waterfall model by enabling iterations in a spiral structure. It consist of four phases (determine objectives, identify and resolve risks, development and test, plan the next iteration). Since the spiral only allows iterations over a complete cycle, micro-iterations are not provided.
A major step in software development is testing. But since this is only considered in a single phase, there is only discrete analysis. However since the cycle in the spiral model is much shorter than in other domains the interval between two analysis steps is much shorter.
The spiral model does not deal with aspects of interdisciplinary work.

2.4.2 Test-driven development

The main aspect of test-driven development is that before implementing any functionality in the software, a test is defined and written to test this function. Only when the function-specific test passes, the development process can proceed. Since there is a test for every software unit, test-driven development offers continuous testing and hence micro-iterations.
Test-driven development is used – among others – within Extreme Programming, a software development methodology introduced by Beck [Beck 1999]. It focuses on adapting to changing customer requirements.
Interdisciplinarity only plays a role within software teams but not across domains.

2.5 Mechatronics

Since mechatronics has become a key field of engineering there has been a lot of research regarding mechatronic design, leading to various different design methodologies. While a comprehensive examination would exceed the scope of this paper, only two of these methodologies are discussed.

2.5.1 VDI 2206

The guideline VDI 2206 combines existing VDI guidelines into a framework for the development of mechatronic products. It adapts the common V-model to mechatronics while remaining intentionally flexible to suit different applications. As a result of this, the guideline is rather abstract.
Although the V-model describes a rather linear process, it is mentioned several times that the different stages do not have to follow this linearity. But as described in VDI 2221 there is no guidance on how to break this linearity. Most of the iterations are intended as a complete repetition of the macro-cycle.
In contrast to VDI 2221, in VDI 2206 there is an explicit stage for the assurance of properties. However, this only takes place at the end of a macro-cycle and is focused on physical prototypes. The question of what and how to test and how the results affect previous stages is not covered.
Since modeling and model analysis are important factors of success in mechatronics they are considered in the guideline as an accompanying activity throughout the entire process. But due to the guideline's flexibility they are not discussed in depth which means that there is no guidance on which techniques to use and how to apply them.
One focus of VDI 2206 is interdisciplinarity. While in system design and system integration this is incorporated, domain-specific design is mostly independent although cooperation and communication in this phase are very important.

2.5.2 Isermann

Isermann [Isermann 2008] developed an extended and more detailed V-model for the development of mechatronic products. In contrast to VDI 2206 the V-model is only passed through once and ends with the production of the system. Therefore the individual steps have to be completed with iterations – i.e. passing micro- instead of macro-iterations – but guidance and criteria on deciding for iterations and determining their scope is not given.
While the standard V-model provides testing activities only in the right branch of the V, Isermann includes a testing or analysis step after almost every stage. Hence there is continuous testing throughout the process. But handling of information from testing and deducing activities is not covered.
Although testing is still oriented strongly towards physical testing, simulation is provided in almost every stage. But concrete discussion of this topic is not included since it is regarded primarily as a tool.
Despite the fact that the main stages are almost the same as in VDI 2206, interdisciplinarity is not further figured out.

2.6 Conclusion

The results of the analysis are summarized in table 1.
As shown, none of the domains currently fully integrates simulation into the mechatronic development process. Some do not even take into account testing or analysis as part of the process model.
Traditionally mechanics dominate the process. But when looking at methodologies from software development and electronics it can be seen that analysis – especially modeling and simulation – plays a much more important role than in mechanics as it is integrated as nearly continuous action. So mechatronic design can benefit by giving more attention to the non-mechanical domains.

Table 1. Results of the analysis (+: covered in detail; o: covered; -: not covered)

Methodology	Iterations	Testing and prototyping	Use of modeling and simulation techniques	Interdisciplinarity
VDI 2221	-	-	-	o
"Münchener Vorgehensmodell"	+	o	-	-
Siegl	-	+	+	-
Gausemeier	-	o	+	-

Spiral model	o	o	-	-
Test-driven development	o	+	-	-
VDI 2206	o	o	o	+
Isermann	+	o	o	o

3. A basic framework for simulation-based mechatronic design

As shown in chapter 2 there are currently deficits regarding the integrative use of modeling and simulation. On the path toward a framework taking into account the features of modeling and simulation, the requirements on such a framework are derived based on the results of the analysis of chapter 2.

3.1 Requirements on a mechatronic framework

Most of the common development methodologies have a very abstract character which avoids concrete guidance on the different situations of the development process. That is one of the reasons why application of those methodologies often fails. Mechatronic development needs more than just a process model. That is why the framework consists of six different viewpoints on simulation-based mechatronic design (see figure 3):

- Process
- Methods
- Responsibilities
- Data management
- Modeling languages
- IT tools

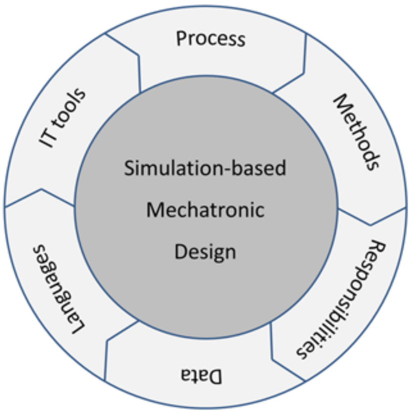

Figure 3. Different viewpoints on simulation-based mechatronic design

Regarding IT tools there are a lot of isolated applications – e.g. finite element software or multibody systems – both across domains and across the process. Since those tools are highly sophisticated and established there is no use in fully integrating them into one software tool instead. They should be connected by giving guidance on when and how to use them in the process and how to assess the influence of the analysis results on other tools.

One of the steps preceding simulation is always modeling. And as with IT tools, every domain has its own languages. But especially in early phases this frequently leads to misunderstandings during system design. That is why there is a need for a common overall modeling language to describe this system behaviour. A lot of research has been done on this topic and one already very mature and promising example is the object-oriented modeling language Modelica®. It combines a graphical user interface offering elements of all domains with a mathematical description of the system in the background. Furthermore, for almost every application a new model has to be defined which means an unnecessary effort and a great amount of data. To cope with this problem there are several attempts to

both derive new models from existing ones and developing common standards for the model exchange.
Data management has become a key factor in product development and is even more important in simulation-based design. Most of the time in simulation phases is spent collecting information and data, offering huge potential for improvement. The above-mentioned deriving of models and common standards, the use of co-simulation and a complete integration of data management into the framework are possible ways to cope with this problem.
Apart from the IT-related topics, organizational aspects play an important role in successful product development. Hence assignment of responsibilities has to be part of the framework. This has to be considered between domains – e.g. regarding system modeling and simulation – as well as within domains. In particular the boundaries between design engineers and computational engineers become blurred.
Modeling and simulation offers nearly endless possibilities of analysis which implies the danger to extend the development process instead of shorten it. Therefore engineers have to be supported with regard to what, how and when to analyze and how to handle results as well as how to derive actions.
The basis for all those aspects is provided by the process model. Simulation changes the procedure compared to traditional design methodologies. The use of simulation makes analyses feasible already in early phases. Moreover the traditional prototyping at the end of the development process is being replaced – or at least significantly shortened – by a continuous analysis throughout the process. Thus macro-iterations are being substituted by several micro-iterations supporting more reasoned design decisions and leading to higher maturity of the product.
In this paper the process model is in the focus as this provides the basis for integration of the other viewpoints.

3.2 Process model for simulation-based mechatronic design

As discussed in chapter 2 none of the traditional methods from mechanics or mechatronics consistently incorporates testing or analysis – especially not continuous testing. In contrast, electronics and software development give much more attention to that. Derived from the procedure in those two domains, the first step of the process model is a definition of analysis milestones (see figure 4). Those analysis milestones specify the analyses to be done during the process. Each of them contains information about the respective stage in process, about who is responsible for this analysis, the properties to be analyzed, the test procedure – e.g. which tools to use –, the analysis conditions – e.g. fully virtual or hardware-in-the-loop simulation – and the boundary values.
Analysis milestones are defined based on the requirements on the system or product. The first analysis scheme is derived after the generation of the requirements list. Since properties are not yet fully known this early in the process, analysis milestones are kept very simple. But – similar to the requirements list – they have to be adapted dynamically to the current process status. Especially after system design – when the system and its properties are defined – the milestones can be concretized. Those milestones are integrated in the process model.

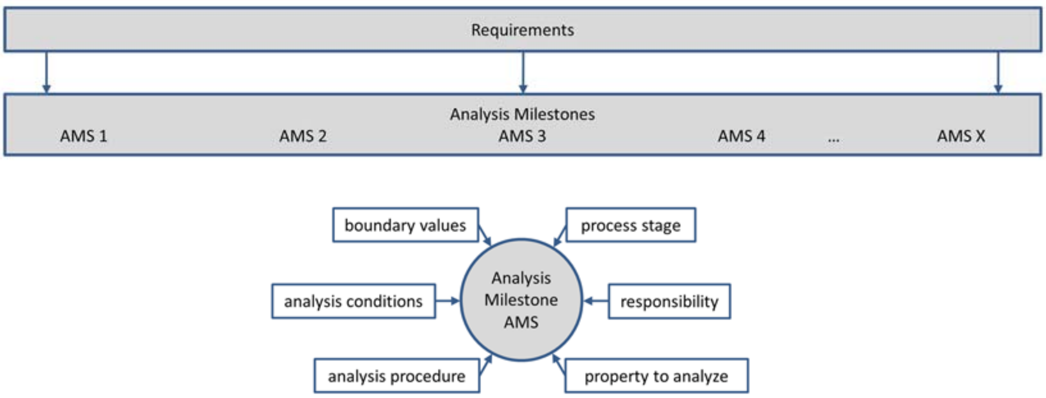

Figure 4. Defining analysis milestones

Since the guideline VDI 2206 provides the current standard for mechatronic design, the preliminary process model shown in figure 5 is based on this structure and extends the V-model in terms of simulation-based design. The input of the process model consist of the requirements and the defined analysis milestones. The first phase is – similar to VDI 2206 – system design involving all domains. After the definition of the conceptual system design, this concept is simulated during simulation phase which is explained in detail below. However, more than one concept may be – indeed, should be – simulated. In this case the simulation results support decision making by offering well-founded, objective criteria. Simulation results are compared with the criteria defined for the corresponding analysis milestone and only if these are fulfilled the system design is validated and the next phase can be started. Otherwise system design has to be revised until the milestone is reached (dashed lines in figure 5). It has to be noted that there is not necessarily just a single milestone for system design. System design can for example be divided into different grades of maturity with each of them having its own analysis milestone – i.e. in sequence – and there can be several parallel milestones for different properties to analyze.

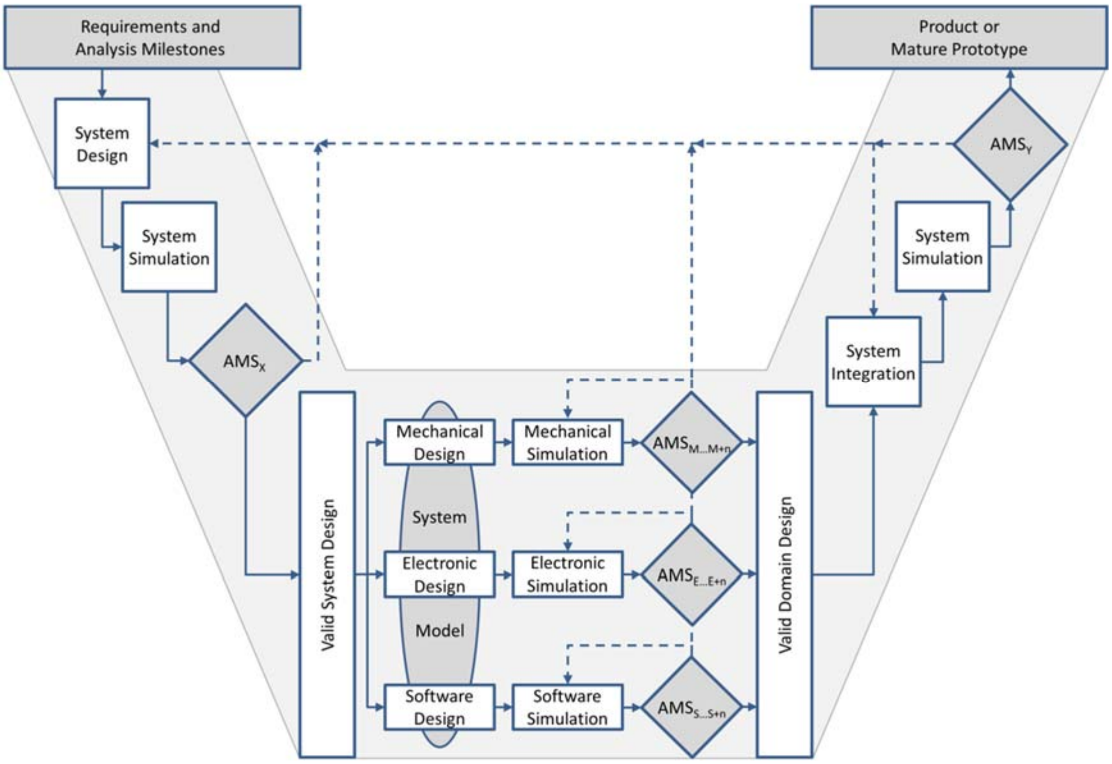

Figure 5. Process model for simulation-based mechatronic design

Based on the validated system design the system is partitioned into the different domains. Independent development between the domains would lead to a late compatibility check of the single domains' design and result in macro-iterations covering the whole domain design or even system design. In order to avoid this, the use of a system model is proposed. This model contains information about the system like interfaces or geometric values. There has already been research on these topics, for example [Stetter et al. 2011] introduced an agent-based system model for such applications based on SysML. The use of this system model allows each domain to check compatibility of its design with other domains to reduce macro-iterations. The procedure of the simulation is the same as in system design: simulation results are compared to the defined analysis milestones and domain design is either valid or it has to be revised. Additionally it can be possible that in domain design problems occur which lead to a revision of the system design. Although these iterations should mostly be eliminated by validation of system design they have to be noted. Of course there can be several milestones both in parallel and in sequence. If all domain designs are validated they are integrated into the complete system. The procedure is the same as in the two phases preceding. However as the maturity of the system is already of a high level there are – apart from the pure virtual simulation – hybrid forms of analysis consisting of both virtual and physical parts. This means that certain parts of the system already physically exist and are tested in a virtually simulated environment. This technique is commonly known as hardware-in-the-loop. In this case, the system model from system design phase ideally can be reused for the simulation of non-physical parts.

After every design phase the process model provides a simulation phase which has still to be elaborated on. Figure 6 shows the procedure within such a phase. The input is always a conceptual design which has to be validated. Based on this concept the system has to be modeled. Since a model is always just a representation of an object, the challenge is to find the right balance between precision and operationality. On the one hand a high-precision model can lead to non-interpretable results because of high complexity, on the other hand an inaccurate model leads to incorrect result – possibly without the engineer being aware of it. So a model has to be chosen to suit the specific purpose which often requires experience. In general, modeling depth increases with the course of development as design maturity increases. Thus every model has to be validated before simulation and potentially be revised. This validation can be performed for example by simple experiments or with test data of other products. If the model is validated the simulation can be started. To validate the simulation plausibility checks or comparison with results of other products can be used. If the simulation is not valid it has to be revised, for example by choosing a different solver. Once model and simulation have been validated the analysis results have to be evaluated and compared to the criteria of the corresponding analysis milestone. If those criteria are fulfilled the output of the simulation phase is a validated design. Otherwise the input design has to be revised.

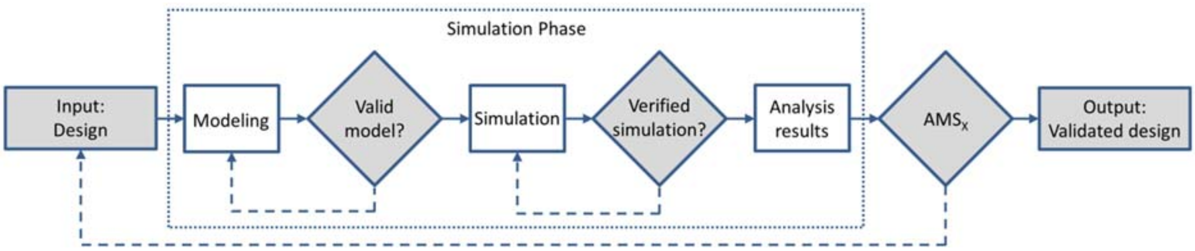

Figure 6. Concretization of a simulation-phase

4. Conclusion and further work

Mechatronics is a quickly-growing design field. The next step towards highly innovative mechatronic products is model- and simulation-based design. Although modeling and simulation have been a part of product development for years, the analysis of common design methodologies regarding the use of modeling and simulation in this paper shows that none of them exploits the full potential, wether in mechanics, electronics, software or mechatronics. In contrast to the large amount of research concerning further development of modeling languages and simulation techniques this paper describes a framework for the integration of modeling and simulation into mechatronic design based on existing techniques.

The basis for this framework is a definition of six viewpoints on mechatronic design to enable an integrated design methodology which is suitable for practical application. Initially, based on the V-model described in the guideline VDI 2206, the process view is extended with analysis milestones and analysis phases to enable continuous testing and micro-iterations.

Further work will concentrate on the elaboration of the five other viewpoints based on the process model introduced here and their integration into a mechatronic framework. There has to be methodical support in defining analysis milestones and evaluating simulation results. Furthermore guidance for the choice as well as the use of existing tools and techniques has to be provided.

This work is part of an international collaboration project, with other partnering universities focusing on the creation of holistic mechatronic models both across domains and along the process. A German automobile manufacturer provides industrial use cases to ensure the practical applicability of the mechatronic framework developed and to transfer the results to the company's PLM landscape.

References

Beck, K., "Extreme Programming Explained", Addison-Wesley Longman, Amsterdam, 1999

Boehm, B., "A spiral model of software development and enhancement", ACM SIGSOFT Software Engineering Notes, Vol. 11,No. 4, 1986, pp 21-42

Dohr, F., Vielhaber, M., "Modellierung und Simulation im mechatronischen Produktentwicklungsprozess", Design for X - Beiträge zum 22. DfX-Symposium, Krause, D., Paetzold, K., Wartzack, S., TuTech Innovation, Hamburg, 2011, pp 129-140

Gausemeier, J., Feldmanm, K., "Integrative Entwicklung räumlich elektronischer Baugruppen", Carl Hanser Verlag München, 2006

Isermann, R., "Mechatronische Systeme", Springer-Verlag Berlin, 2008

Lindemann, U., "Methodische Entwicklung technischer Produkte", Springer-Verlag Berlin, 2007

Siegl, J.,"Schaltungstechnik – Analog und gemischt analog/digital", Springer-Verlag Berlin, 2005

Stetter, R., Seemüller, H., Chami, M., Voos, H., "Interdisciplinary System Model for Agent-Supported Mechatronic Design", Proceedings of the 18th International Conference on Engineering Design (ICED11), Vol. 4, Culley, S. J., Hicks, B. J., McAloone, T. C., Howard, T. J., Lindemann, U., Copenhagen, 2011

Verein Deutscher Ingenieure, VDI guideline 2206, "Design methodology for mechatronic systems", VDI-Verlag Düsseldorf, 2004

Verein Deutscher Ingenieure, VDI guideline 2221, "Systematic approach to the development and design of technical systems and products", VDI-Verlag, Düsseldorf, 1993

Dipl.-Ing. Fabio Dohr
Research associate
Saarland University, Institute of Engineering Design
Campus E2 9, 66123 Saarbrücken, Germany
Telephone:+49 681 302 71328
Telefax: +49 681 302 71317
Email: dohr@lkt.uni-saarland.de
URL: http://www.lkt.uni-saarland.de