ICED13
INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

SOLUTION PATTERNS TO SUPPORT THE KNOWLEDGE INTENSIVE DESIGN PROCESS OF INTELLIGENT TECHNICAL SYSTEMS

Harald ANACKER (1), Thomas SCHIERBAUM (1), Roman DUMITRESCU (2), Jürgen GAUSEMEIER (1)
1: Heinz Nixdorf Institute, Germany; 2: Fraunhofer Institute for Production Technology IPT, Germany

ABSTRACT
Recently, mechatronics as a self-contained discipline has undoubtedly shaped the development of technical systems. Mechatronics stands for the close interaction of mechanics, electronics, control engineering and software engineering. Due to the advancement of information and communication technologies, the functionality of mechatronic systems will go far beyond current standards. The increasing complexity requires a consistent comprehension of the tasks between all the developers involved. Especially during the early design phases, the communication and cooperation between the engineers is necessary to design a first overall system model. In addition, reusing of once successfully implemented solution knowledge is becoming increasingly important related to the overall context of the triangle of tension formed by time, cost and quality. Today the reuse is well established within the different disciplines. But usually, the sum of separate single solutions is rarely the best solution on a systems-level. This requires an approach, to support the knowledge intensive design process of intelligent technical systems by a new form of multidisciplinary abstract described solution patterns.

Keywords: design methodology, mechatronics, solution pattern, solution knowledge

Contact:
Harald Anacker
Heinz Nixdorf Institute
Product Engineering
Paderborn
33102
Germany
harald.anacker@hni.upb.de

1 INTRODUCTION

Today, mechanical engineering products are characterized by the close interaction of mechanics, electronics, control engineering and software engineering. This interaction is expressed by the term mechatronics. In order to categorize the variety of applications, the diversity of mechatronic systems can be centralized and expressed by the three categories “Spatial Integration of Mechanics and Electronics”, “Multibody-Systems” and “Intelligent, networked Systems” (advanced mechatronic systems). Figure 1 shows an overview of the wide variety of mechatronic systems.

The first category is based on the integration of mechanics and electronics. The aim is to reach a high density of mechanical and electronic functions within a limited amount of space. The second category deals with the controlled movements of multibody systems. The objective is to improve the system’s behavioral movement. Mechatronic systems are always part of the two categories.

The conceivable development of information technology opens up fascinating perspectives which have the potential to go far beyond current standards. Keywords as “Things that Think”, “Cyber-Physical Systems” or “Industry 4.0” express this perspective on Intelligent Technical Systems (see figure 1, category three). Intelligent technical systems have inherent partial intelligence. They can independently modify their behavior during operation by following the recurring actions: Analyzing the current situation, determining the system’s objectives and adapting the system’s behavior (Mehrabian & Russell, 1974). “Things that Think” means the integration of self-x properties to enable more adaptive, robust, foresighted and user-friendly systems in general (Gausemeier et al., 2009). “Cyber-Physical Systems” characterize the rising complexity of technical systems based on the decomposition of information processing in subsystems and the associated necessary combination, coordination and communication between the subsystems (Broy, 2010); examples are embedded systems. The most important platform of communication will be the Internet. However, cyber-physical systems do not only solve problems in cyber space in a collaborative way, but also have a direct impact on real physical processes by the use of their actuators. “Industry 4.0” means the fourth industrial revolution based on the integration of intelligence for monitoring and controlling processes in production systems that go even beyond company boundaries (Broy, 2010).

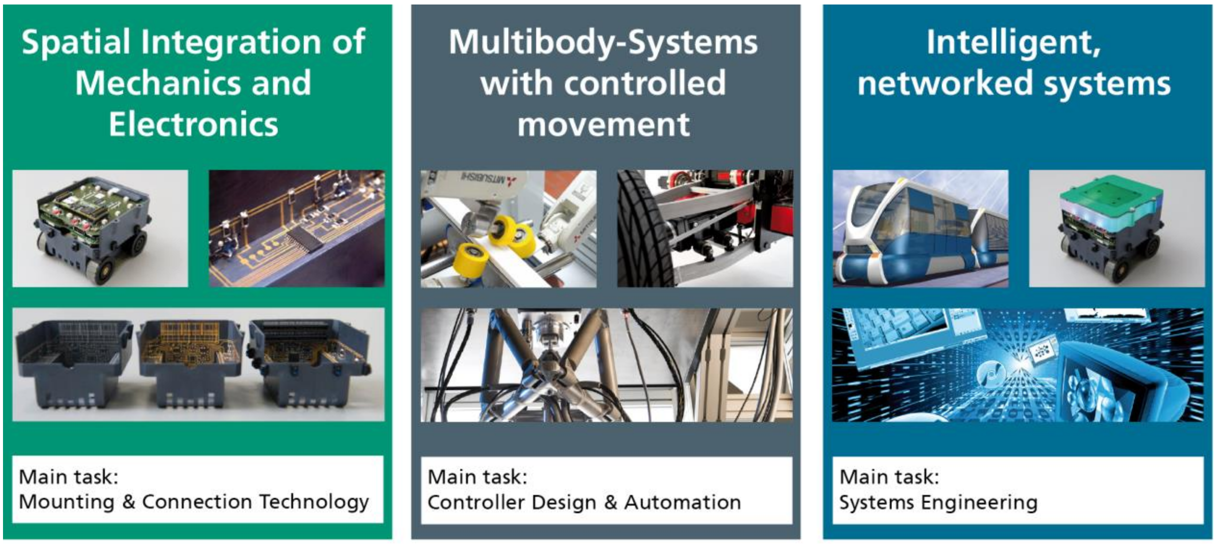

Figure 1: Categories of Mechatronic Systems

In section two, we will explain the initial situation and our field of action in context system design. Afterwards we will carry out our developed approach in detail. Significant parts are a specification technique to describe the principle solution of mechatronic systems, a uniform type of solution patterns and the formalization of the central aspects requirements and functions. In section 4 we will evaluate our results over the course of the design of two cooperating delta robots. To conclude, we will sum up the major points and give a short outlook on our future work.

2 INITIAL SITUATION AND FIELD OF ACTION

Even though mechatronics as a self-contained domain is rather new in comparison to other disciplines, several design methodologies in both academia and within the industry exist. Most of them focus on one aspect or discipline; a process-related view on systems-level is still missing. The core of a design-methodology for intelligent technical systems must be a procedure model that focuses on the integration of all participating disciplines during the early design-phases (Rzevski, 2003). On the one hand the procedure model has to include all necessary activities and tools. On the other hand the procedure model has to be primarily oriented in the well-grounded problem-solving cycle (Hall, 1962; Haberfellner, et al., 2012). The minimal consensus of all approaches is covered in the VDI-guideline 2206 (VDI, 2004). The main concept is the V-model, which gives an overview over the general procedure of designing mechatronic systems. For this purpose, the guideline distinguishes the process into three basic phases. During the system design, a cross-domain principle solution is being developed. The domain-specific design follows, in which the different participating disciplines work in a parallel manner. In the system integration, the developers merge the discipline-specific results, verify and validate them.

In addition solution knowledge and its reuse in context product engineering is of major significance. Usually the solution knowledge is at least implicitly provided in the form of person-specific actions or the knowledge is undetected in different knowledge sources like documents, product catalogues or computer-internal models (e.g. software-code). In order to efficiently develop a suitable externalization of the implicit knowledge is necessary. A high number of different approaches exist that focus such knowledge management processes (e.g. expert systems, knowledge maps, data basis…). All approaches are more or less based on the SECI-model of Nonaka and Takeuchi (1997). The main reasons for the necessity of a knowledge management within a company are: sustainable competitiveness, continuous improvement process, retaining knowledge despite employee turnover (Probst et al., 2006). The changing characteristics of engineering/manufacturing processes of intelligent technical systems and especially the close interdisciplinarity within the design require new approaches for the knowledge management. The significant requirements are: taking all involved disciplines into account, reaching an abstract level at the very beginning to support the design process in a solution-neutral way, creating a dynamic approach that grows with every passing development process, reducing the workload for the developer by the process activities of 'saving' / 'searching' suitable solution knowledge. Nowadays a great variety of different approaches exist like e.g. knowledge maps or topic maps etc. (Maurer, 2011; Weber, 2011). The major disadvantage is saving the solution representation by textual description. Against the background of model based systems engineering (system design) a new type is necessary. Therefore we use of solution patterns. A pattern is subdivided in different categories which describe a specific problem, the core of the solution and context (Alexander et al., 1977). We propose solution patterns as a success-promising approach (Cloutier, 2008), because the different categories of each pattern can be filled with generic and abstract models. These characteristic offers a direct integration into the design process.

The initial situation in context product engineering is shown in figure 2. At the early stages an interdisciplinary team of engineers work together during the system design. The goal of this phase is to conceive the first concept of the system, which is called the principle solution. Afterwards during the concretization the engineers of the involved disciplines work in a parallel manner. The system specification is necessary to consolidate changes of different disciplines which are relevant on a system's level. The cycle diagrams symbolize the reuse of solutions within the different disciplines. Well established discipline-specific approaches exist, which will be described in detail in section 3. At the end of the process stand the developed product and the related production system. The description is simplified and is sequenced to show the basic relations. In reality the design process is characterized by iteration loops and detailed activities.

Our field of action is the identification and externalization of solution knowledge to support the system design by interdisciplinary knowledge artifacts. In general potential knowledge for the externalization must be validated. Therefore we propose the detailed computer-internal models (software code, cad-model etc.) and their interactions as the starting point for the identification- and externalization-process. In addition we will take into consideration existing discipline specific approaches of the concretization phase. Our overall aim is to provide multidisciplinary solutions because generally speaking the sum of all discipline-specific solutions is rarely the best solution on the systems-level.

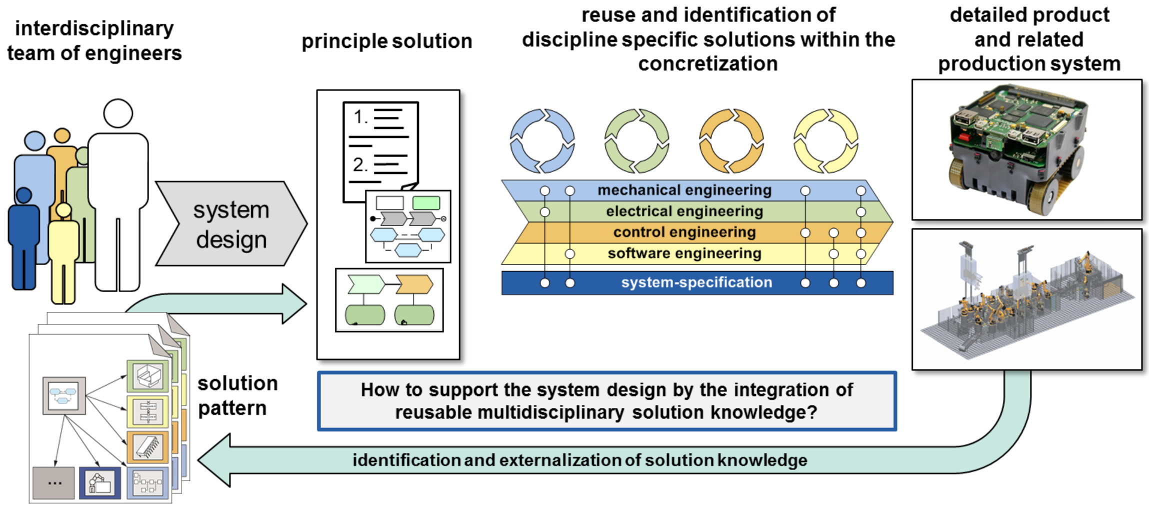

Figure 2: Solution patterns in context product engineering

3 Approach for the identification and generation of Solution Patterns

As already mentioned the design of intelligent technical systems is a highly knowledge-based process – especially due to the large number of different experts and their interaction during the early design phases. The systematization of solution knowledge in context product engineering is pictured in figure 3. At the top of the figure the design process is shown in a highly abstract way. The systematization is presented in the lower half. The connection of the different knowledge artifacts and the design process is symbolized by the colored cycle diagrams which represent a discipline. We basically distinguish two kinds of reusable knowledge which are relevant sources for the definition of multidisciplinary solution patterns: working principles and design patterns.

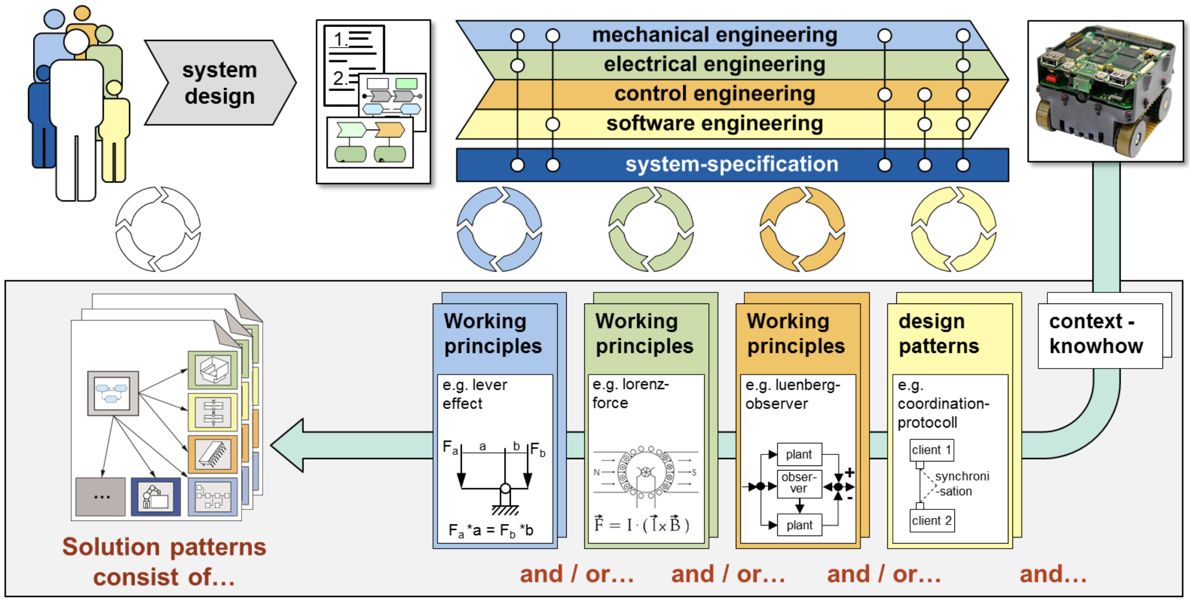

Figure 3: Systematization of Solution Knowledge

Working principles are well established in mechanical (e.g. lever effect), electrical (e.g. lorenz-force) und control engineering (e.g. luenberg-observer). These principles describe the relationship between the physical, biological or chemical effect and the geometrical and material characteristics of system elements. In order to integrate working principles during the design process, an essential step has to be carried out: specifying the functional description. According to Pahl et al. (2007), a function describes the desired relation between the input and output of a system with the objective to fulfill a certain task (Pahl et al., 2007). Specifying the functional description of the system helps developers abstract from former prefixed solutions from which they may have in mind. Thus, a function has to depict the solution in a neutral and abstract way as well as the volitional relation between the input and the output of a system. The functional description is done by building a function hierarchy. The overall function of the system is at the top of this hierarchy. It is subdivided in subfunctions up to a point when suitable working principles for these subfunctions can be found. To name the functions of technical systems, different noun-verb-catalogues for mechanical engineering have been developed (Koller & Kastrup, 1998; Roth, 2001). Possible appendages to create reusable solutions for control engineering are mathematic relations that are explained by Föllinger (2008): e.g. catalogues of elementary transfer elements with related functions or complex approaches like observers e.g. luenberg-observer or kalman-filter0. In addition the so-called "design patterns" are a description of a reusable combination of elements for object-oriented software. Well-known approaches are design patterns from the "gang of four" (Gamma et al., 1994).

Nevertheless solution patterns for the system design have to possess different characteristics. The characteristics are: a uniform pattern representation to combine and compare patterns and to fix the engineering design-knowledge, the consideration of the described working principles and design patterns, providing essential context know-how. As shown in figure 3, a solution pattern consists of working principles and/or design patterns and general context-specific knowledge. Therefore, to create a new solution space for the system design and to enable an interdisciplinary team of developers to get a benefit, we have to solve many challenges:

1) As explained before, all involved experts from different disciplines have to cooperate closely especially during the design of the system. Hence, a superior system model is necessary, which combines all the essential aspects of mechanical, electrical and software engineering (INCOSE, 2007). This system model is the basis of the first analysis, verification and validation on the systems level (VDI, 2004).

2) A uniform specification of solution patterns is needed, that allows the common syntheses of abstractly described solutions of the participating disciplines during the system design. In addition, a new classification of such multidisciplinary solutions is necessary.

3) The main focus of our approach is to search efficiently for suitable solution patterns. Therefore we have to develop a new approach in context product engineering which can go far beyond current standards like keyword search option. The perspectives of such search methods are expressed by "semantic web" or "web 2.0". The basic prerequisites are a formal representation of the search query and the counterparts in the solution space: requirements and functions.

3.1 Specification technique to describe the principle solution

Within the conceptual design, the basic structure and the operation modes of the system are defined. The result is called the principle solution which does not only describe the physical characteristics, but also the logical operating characteristics. For this purpose a new specification technique was developed within the CRC 614 (Gausemeier et al., 2009). The description of the principle solution is divided into several aspects: requirements, environment, application scenarios, functions, active structure, system objectives, shape and behavior. The aspect behavior consists of different kinds of behavior, e.g. logic behavior, the dynamic behavior of multi-body systems, the cooperative behavior of system components.

The aspects shown are computer-internal represented by partial models. The relations are modeled between the various constructs of the relating partial models and amount to a coherent system. By using this specification technique, the system is modeled in a holistic and domain-spanning way. The specification of the principle solution forms the basis for the communication and cooperation of the developers from different disciplines during the complete development process.

3.2 Solution Knowledge

According to ALEXANDER et al. a pattern describes a recurring problem and the core of a solution (Alexander et al., 1977). Therefore we developed a uniform specification of solution patterns for the development of intelligent technical systems (Dumitrescu et al., 2010) (Fig. 5). This specification is significant for all the involved disciplines. Therefore it is structured into six aspects:

The aspect *characteristics* describe the properties of the pattern. They allow inferences on which requirements the pattern can meet. Some examples for the relevant characteristics of the basic system can be geometrical or material. Some examples for characteristics of pattern for information processing are processing speed or the type of calculation. The aspect *functions* lists and describes the functions, which can be implemented by the solution pattern. Thus this aspect expresses the problem description. The aspect *active structure* is the core of the solution description. This aspect specifies which system elements are necessary in order to implement the functions of the pattern and how those system elements are interrelated. With the aspect *behavior* the description of the solution is completed. For this purpose the behavior of the system elements or of logical groups of several system elements is described. The *solution principles* are the basis for the implementation of the solution pattern after the conceptual design. In general there are physical effects (e.g. lorentz force) on the physical system parts and methods (e.g. planning and learning algorithms) for the information processing as it is the focus of this contribution. Methods serve to implement the self-optimization process, in particular to adapt objectives and the system's behavior. Within the aspect *context* applications, in which the solution pattern was implemented successfully, are getting specified.

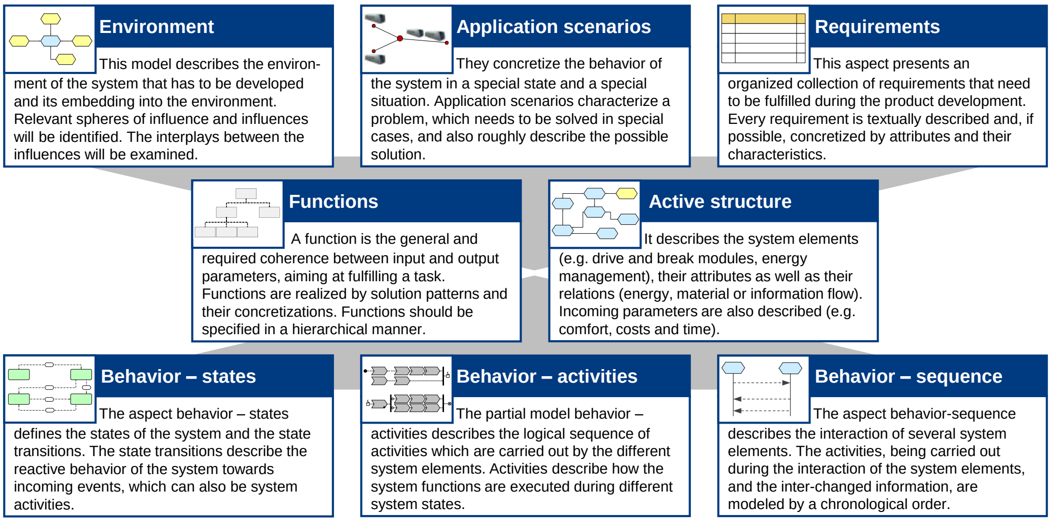

Figure 4: Short description of the used specification technique

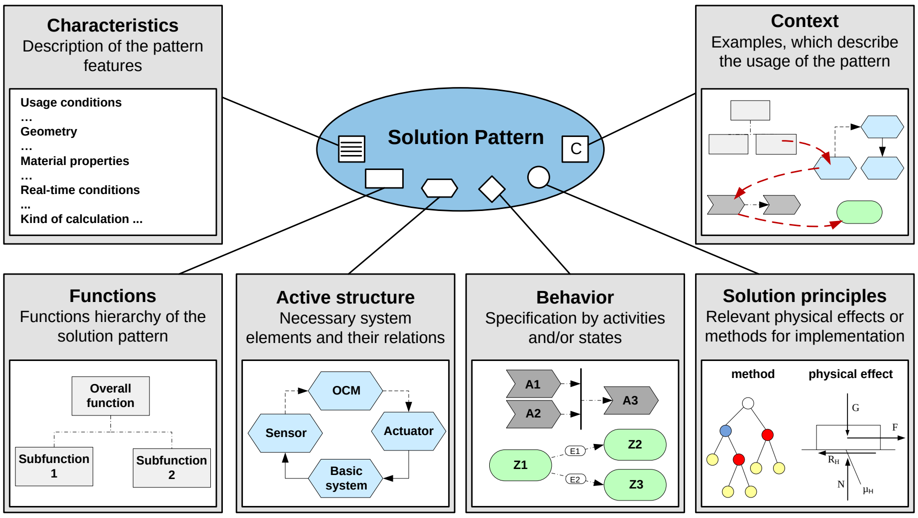

Figure 5: Uniform specification of solution pattern for the conceptual design

3.3 Formalization of requirements and functions

Formal representation of requirements and functions helps to support developers during the design phase (Rupp, 2009). Furthermore it relieves developers from time-consuming processes like searching and finding suitable solutions for their given problem and also allows the direct integration of solution patterns into the developing process. Formal requirements can also be useful to identify changes in the concept changes between the involved disciplines. As shown in figure 6 we present the selection of suitable solution patterns based on requirements and functions. Formal requirements can be mapped with the aspect characteristics from any solution pattern. This also occurs in the aspect functions.

Therefore a formal computer-internal representation is needed. Our approach and the differences between the requirements and functions will be explained in the following.

Requirements: Nowadays different approaches for requirements engineering and -management exist. The overall aim is the handling of the increasing complexity based on the wide range of requirements (Hood et al., 2007). Against the background of our field of action “efficient search for suitable solution patterns” today’s approaches insufficiently encounter the outlined challenge. They often focus on one discipline so that a holistic consideration of multidisciplinary systems is still missing (Birkhofer, 2001). To overcome this challenge we developed a new approach for formalized requirements description. It mainly consists of three aspects: requirements documentation, parser and taxonomies (see figure 6). They will be clarified in more detail in the following. Requirements are documented with the help of requirements lists as they are mentioned in well-known literature like Pahl & Beitz. The focus is to reduce restrictions for developers by drafting requirements. There are only a few restrictions for drafting requirements: Each requirement may include a characteristic and has to be referenced to the system itself or part of the system (e.g. electrical power has to be used by the system); each requirement may only describe one characteristic (e.g. height or length). In the next step the requirement is divided into sentence stems (single words) by a parser (see fig. 6 (1)) and allocated to word-classes (e.g. verbs, nouns, adjectives). Sentence stems and allocated word-classes are used afterwards by means of taxonomies to create formal requirements. Different taxonomies (e.g. verb-taxonomy, unit-taxonomy, characterizes-taxonomy), allow to create semantics in requirements (see fig. 6 (2)). Furthermore taxonomies help to avoid misinterpretation by ambiguity and discipline-specific use. The characteristics-taxonomy also includes categories (e.g. geometry, security) according to Pahl et al. (2007). This enables the allocation of requirements to those requirement-categories (e.g. height to category geometry). Thus, requirements can be linked to solution patterns (see fig. 6 (3)).

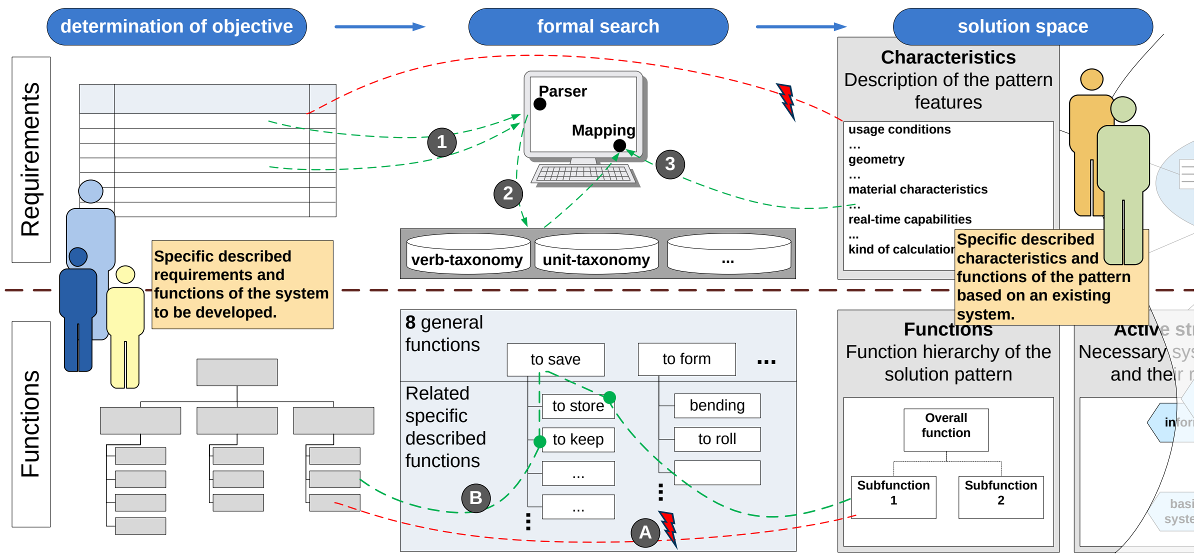

Figure 6: Search for solution patterns based on formalized requirements and functions

Functions: To find possible solution patterns the relation between patterns and systems functions are quite important. The functional description of a solution pattern as well as the definition of the desired system has to be comparable. Without an appropriate instrument a search query on the direct path is not successful (see fig. 6 (A)). The description must include all the specific information of development intention which the developer has in mind. To consider the semantic of the function description a first approach of ontology-based search was implemented. The core is ontology with general functions for the description of mechatronic systems. We analyzed established catalogues of functions which are mainly focused on mechanical engineering and built a catalogue of mechatronic functions (Koller and Kastrup, 1998; Roth, 2001). It contains only simple substantive-verb combinations and assigns the verbs of the three system-flows: the energy, material and signal flow. These build a common subset for all mechatronic systems and consequently they are not very detailed. Depending on the specific application solved by a solution pattern the functional description has to be more specific in order to depict the solved problem precisely. Although this information is interpretable by other developers, a successful mapping is still missing. Hence, the specific functions will be transferred to general functions (see fig. 6 (B)). Within the ontology all verbs and system-flows are structured and valid combinations are modeled using object properties.

4 SYSTEM DESIGN OF TWO COOPERATIVE DELTA ROBOTS BASED ON SOLUTION PATTERNS

To explain our results in this chapter, we will use the development of a cooperating robotic system as an example (figure 7). The system consists of two identical robots with delta kinematics and three electric drives. The device's center point is a solid circular surface that allows a ball to be rallied between two robots. An external sensor to determine the trajectory and point of impact on the surface is not present. Each robot acts as an independent player and calculates the expected trajectory of the ball. To explain our results we subdivide this chapter into the paragraphs "Multi-criteria choice of solution patterns" and "Principle Solution".

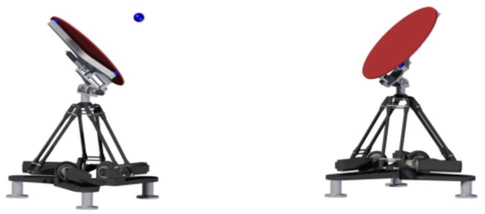

Figure 7: Application Example "two cooperative delta-robots"

Multi-criteria choice of solution patterns: Based on the partial model requirements and functions the choice of suitable solution patterns can be realized (see figure 8). In order to search for suitable solution patterns, requirements e.g. that "electric energy has to be used by the actuators", are parsed and formalized by means of a parser and taxonomies. In the next step the part of the requirement which describes the characteristics of the system ("electric energy has to be used") is used to be mapped with characteristics of solution patterns. As shown in figure 8, both solution patterns fulfill the part "electric energy has to be used". Thus a solution space with suitable solution patterns is created. In the second step functions of the function hierarchy are used. As every developer describes functions in his own way, a catalogue of functions and ontology are used for a formal search. Functions like "movement in x-direction" are indirectly linked to the solution pattern "delta kinematics" by "move within 3d space". In such cases our approach supports finding suitable solution patterns, although the functions are not named in the same way. For all suitable solution patterns another solution space is created. Therefore we developed a classification that is based on the basic structure of a mechatronic system – actuators, sensors, basic system and information processing. In the third step the classification is used to map the referenced parts of the system with solution patterns by checking if they belong to the class which is referred to within the requirement. A third solution space is created. By combining all three solution spaces, feasible solution patterns for all aspects are presented. Developers are now able to make their choice out of a pool of suitable solutions for the specific problem. In our example the presented solution patterns are "synchronization" and "delta kinematics".

Principle Solution: After selecting suitable solution patterns the synthesis of the partial models active structure and behavior shall follow. Figure 9 shows the results of our developmental task. The initial active structure was built up by dragging and dropping of selected solution patterns. Afterwards the team of developers has to modify the different aspects depending on the different design task. In our considered case, all transitions and active elements in frames are highlighted. The extraction of the active structure points out the interactions between physical and non-physical system elements (software components) within robot 1 as well as the interaction with the user and robot 2. In general, it is possible that one specific information flow in the active structure includes several distinct sequence diagrams. The used solution patterns are highlighted in figure 9 and will be briefly explained.

The highlighted sequence diagram shows the complex interaction between the coordination module of robot 1 and the environment element of robot 2 before starting with juggling the ball. In this case, the coordination module sends a proposal "playing?" to robot 2. After checking the proposal, robot 2 sends the acceptance message "playing!". After receiving the acceptance message, robot 1 switches to the state „active" and subsequently the robots start the game. The sequence of synchronization is finished, if either robot 1 or robot 2 sends the command "deactivate". This pattern was identified after the successfully completed development of the miniature robot shown in figure 8. The source of the second solution pattern “delta kinematics” was a pick and place application. The general physical structure of the kinematics is usually the same. Some examples of the differences in the case specific detailing are the size of the bars and eccentric drives, the type of the center and the parameter setting of the local axis controller.

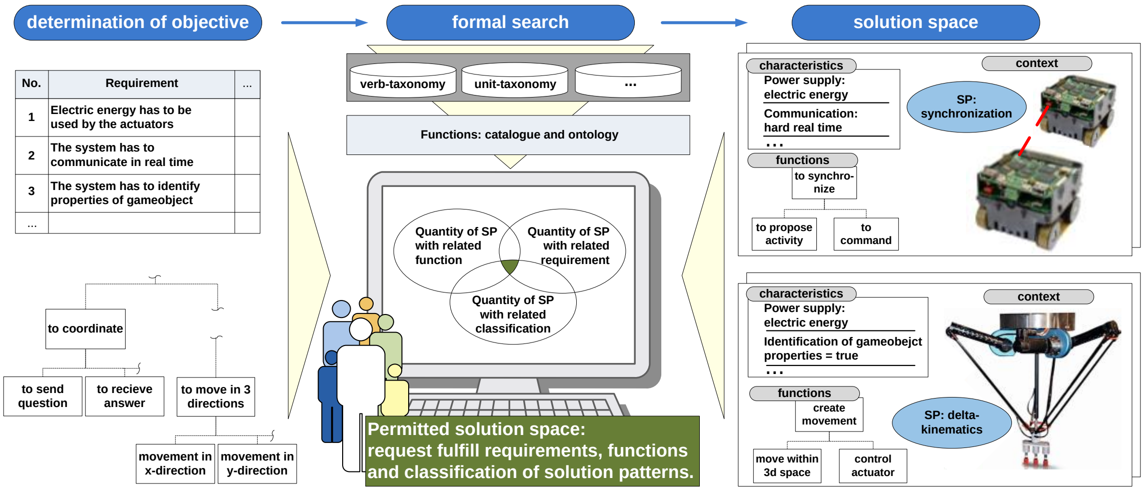

Figure 8:Multi-criteria choice of solution pattern

As shown in figure 2 and figure 3 a key benefit of solution patterns is the recurring cycle of using a pattern and the generation of new ones. We have presented the identification of the two used solution patterns, identified in previous product developments. Now the developed principle solution of the cooperate delta robots (figure 9) is the basis for a new complex pattern consisting of the synchronization, the local control and the delta kinematics.

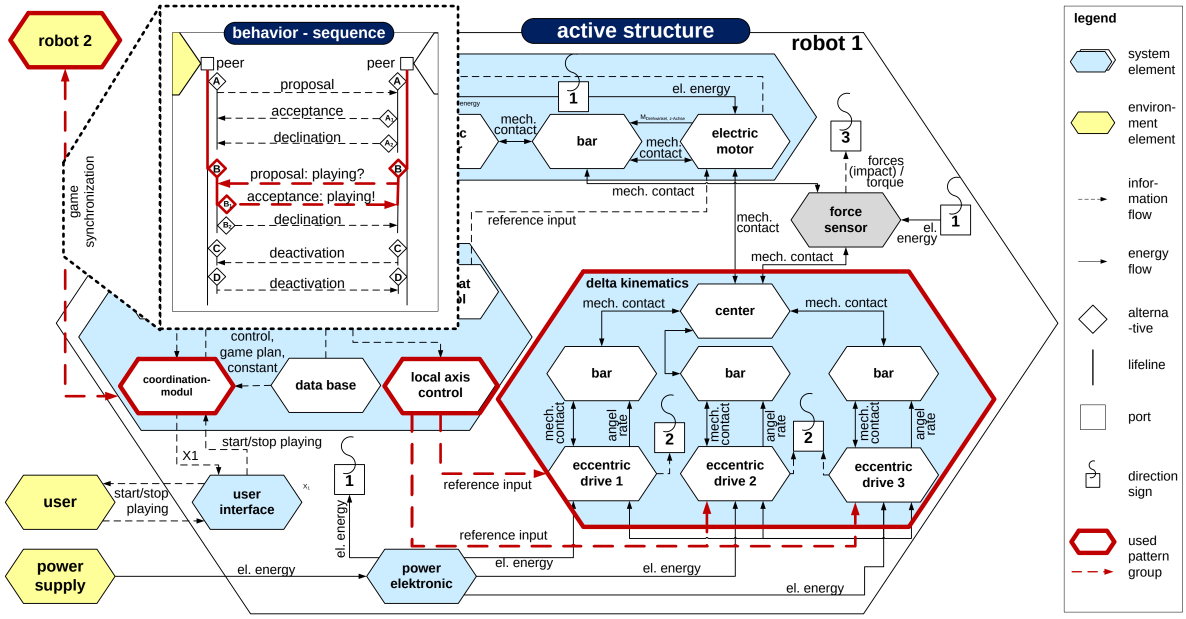

Figure 9: Principle Solution of a delta-robot (highlighted: use of two selected SP’s)

4 CONCLUSION

The design of intelligent technical systems is a challenge, not only because of the increasing complexity of the system, but also because of its design process. Nowadays it is clear that the reuse of once successfully implemented technical know-how is becoming more and more important. To support the knowledge intensive design process we have developed and presented a new approach. We introduced a specification technique to describe the principle solution of intelligent technical systems as well as a uniform specification of “Solution Patterns”. This uniform specification will enable all participating developers to reuse multidisciplinary solutions. We explained the necessity of the formalization of requirements and functions and our related approach. To conclude we presented the use of solution patterns within the system design. Therefore we showed the multi-criteria choice of solution patterns based on formalized requirements and functions and our synthesized principle solution of the two co-operative delta robots. In our future work, we are going to enhance and enlarge the catalogue of solution patterns. In addition we want to define an overriding system to define solution patterns by reverse-engineering a developed product and its production system. In this regard we will expand the use of solution patterns for the integrative design of the product and related production system.

ACKNOWLEDGMENTS

This contribution was developed in the course of the Collaborative Research Centre 614 "Self-Optimizing Concepts and Structures in Mechanical Engineering" funded by the German Research Foundation (DFG) under the grant number SFB 614. This work was also developed in the project "ENTIME: Entwurfstechnik Intelligente Mechatronik" (Design Methods for Intelligent Mechatronic Systems). The project ENTIME is funded by the state of North Rhine-Westphalia (NRW), Germany and the EUROPEAN UNION, European Regional Development Fund, "Investing in your future".

REFERENCES

Alexander, C.; Ishikawa, S.; Silverstein, M.; Jacobson, M.; Fiksdahl-King, I.; Angel, A. (1977), 'A Pattern Language', *Oxford University Press,* New York.

Birkhofer, H. (2001), 'The Future of Design Methodology', *Springer,* London

Broy, M., ed. (2010) 'Cyber-Physical Systems – Innovation durch softwareintensive eingebettete Systeme', *acatech DISKUTIERT,* Springer Verlag, Berlin.

Cloutier, R. (2008), 'Applicability of Patterns to Architecting Complex Systems – Making Implicit Knowledge Explicit', *VDM Verlag*, Saarbrücken.

Dumitrescu, R., Anacker, H., and Gausemeier, J. (2010), 'Specification of Solution Patterns for the Conceptual Design of Advanced Mechatronic Systems', In Proceedings of: International Conference on Advances in Mechanical Engineering, Shah Alam, Malaysia.

Föllinger, O. (2008): 'Regelungstechnik – Einführung in die Methoden und ihre Anwendung' *Hüthig Verlag,* Heidelberg.

Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. (1994) 'Design-Patterns – Elements of Reusable Object-Oriented Software' *Addison-Wesley Verlag,* München

Gausemeier, J., Frank, U., Donoth, J., and Kahl, S. (2009), 'Specification Technique for the Description of Self-Optimizing Mechatronic Systems', *Research in Engineering Design,* 20(4), pp. 201-223.

Gausemeier, J.; Rammig, F. J.; Schäfer, W. (2009), 'Selbstoptimierende Systeme des Maschinenbaus – Definitionen, Anwendungen, Konzepte', *HNI-Verlagsschriftenreihe,* Band 234, Paderborn.

Haberfellner, R.; Fricke, E; Weck, O.; Vössner, S. (2012) 'Systems Engineering – Grundlagen und Anwendung'. *Orell Füssli Verlag,* Zürich

Hall, A.D. (1962) 'A Methodology for Systems Engineering', *Princeton University Press*

Hood, C.; Wiedemann, S.; Fichtinger, S.; Pautz, U. (2007) 'Requirements Management: The Interface Between Requirements Development and All Other Systems Enginnering Processes', *Springer,* Berlin

International Council on Systems Engineering (INCOSE) (2007), 'Systems Engineering Vision 2020'. *Incose-TP-2004-004-02*.

Koller, R., and Kastrup, N., (1998), 'Prinziplösungen zur Konstruktion technischer Produkte.' *Springer Verlag,* Berlin.

Maurer, M. (2011), 'Knowledge Transfer Applying the structural Complexity Management Approach', in: *Journal of Information Retrieval and Knowledge Management,* Vol. 1

Mehrabian, A and Russell, J.A. (1974) 'An Approach to Environmental Psychology', *MIT Press,* Cambridge.

Nonaka, I.; Tackeuchi, H. (1997), 'Die Organisation des Wissens – Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen', *Campus Verlag,* Frankfurt am Main.

Pahl, G., Beitz, W., Feldhusen, J., and Grote, K.-H. (2007), 'Engineering Design – A Systematic Approach', *3rd ed. Springer Verlag,* London.

Probst, G.; Raub, S.; Romhardt, K., (2006) 'Wissen managen – Wie Unternehmen ihre wertvollste Ressource optimal nutzen', *Gabler Verlag,* Wiesbaden.

Roth, K.-H. (2001), 'Konstruieren mit Konstruktionskatalogen, Band 2, Kataloge. ' *Springer Verlag,* Berlin.

Rupp, C. (2009), Requirements-Engineering und -management, *Hanser*, München, Wien
Rzevski, G. (2003), ‘On conceptual design of intelligent mechatronic systems‘, in: *Mechatronics*, Volume 13
Verein Deutscher Ingenieure (VDI) (2004), ‘VDI-Guideline 2206 – Design methodology for mechatronics systems‘, *Beuth Verlag*, Berlin.
Weber, H. (2011), ‘Erstellung nutzerindividueller Dokumente für die Vermittlung von Produktentwicklungswissen durch den Einsatz von Topic Maps‘, Phd-Thesis, Fachgebiet Produktentwicklung, TU Darmstadt, VDI-Verlag, Düsseldorf