



# PATTERN BASED SYSTEMS ENGINEERING - APPLICATION OF SOLUTION PATTERNS IN THE DESIGN OF INTELLIGENT TECHNICAL SYSTEMS

H. Anacker✉, R. Dumitrescu, A. Kharatyan and A. Lipsmeier

Fraunhofer IEM, Germany

✉ [harald.anacker@iem.fraunhofer.de](mailto:harald.anacker@iem.fraunhofer.de)

## Abstract

For the development of intelligent technical systems, Systems Engineering and Solution Patterns are the guarantee for success. In order to avoid cost-intensive iterations, the documentation and reuse of solution knowledge is addressed during the systems design. Using an interdisciplinary specification technique, a uniform structuring of Solution Patterns as well as the composition in a multidimensional knowledge space takes place. This is the basis of an associated systematics for a solution pattern-based system design of mechatronic systems, which is validated by two cooperating DeltaRobots.

*Keywords: systems engineering (SE), solution pattern, solution knowledge, knowledge-based engineering (KBE), mechatronics*

## 1. Introduction

Innovations for modern mechanical engineering are already based today on the close interaction of mechanics, electrics/electronics, control engineering and software. The result are globally distributed **intelligent technical systems** that are interconnected via the Internet and operate in a network. Central challenges in the development of these systems are the increased interdisciplinarity and the increasing complexity of the system to be developed itself. The development cannot take place from the point of view of a single discipline. Successful development must be based on thinking and acting in which the multidisciplinary overall system is at the focus of all activities. One approach that addresses these challenges is **Systems Engineering**. System Thinking is at the forefront of Systems Engineering. Following other established approaches, such as [VDI 2206 \(2004\)](#), it extends over the entire product development, starting with an initial product idea, through the requirements, to the verified overall system (cf. Figure 1; left-hand side).

The aim of Systems Engineering is to achieve a uniform and holistic understanding of the system among all stakeholders involved. An essential field of action is the interdisciplinary design or conception in the early phases of Model-Based Systems Engineering. The system to be developed is described abstractly and holistically with the help of a system model. Specific languages such as SysML or CONSENS are used. An initial creation of the system model takes place in the interaction of all disciplines involved. It forms the basis for communication and cooperation along product development from the early phase to the verified overall system. The system model is required in order to master complexity at the overall system level and to be able to manage the development.

In addition to increasing system complexity, constantly increasing market requirements and shorter product life cycles require higher efficiency in development. In this context, the reuse of existing artifacts and thus the reuse of once successfully applied **solution knowledge** plays an increasingly important role (cf. Figure 1; right-hand side). A suitable knowledge management must be part of each enterprise as basis for creativity and innovations (Albers and Gausemeier, 2012; Probst et al., 2012; Gausemeier et al., 2012). In addition to increasing system complexity, constantly increasing market requirements and shorter product life cycles require higher efficiency in development.

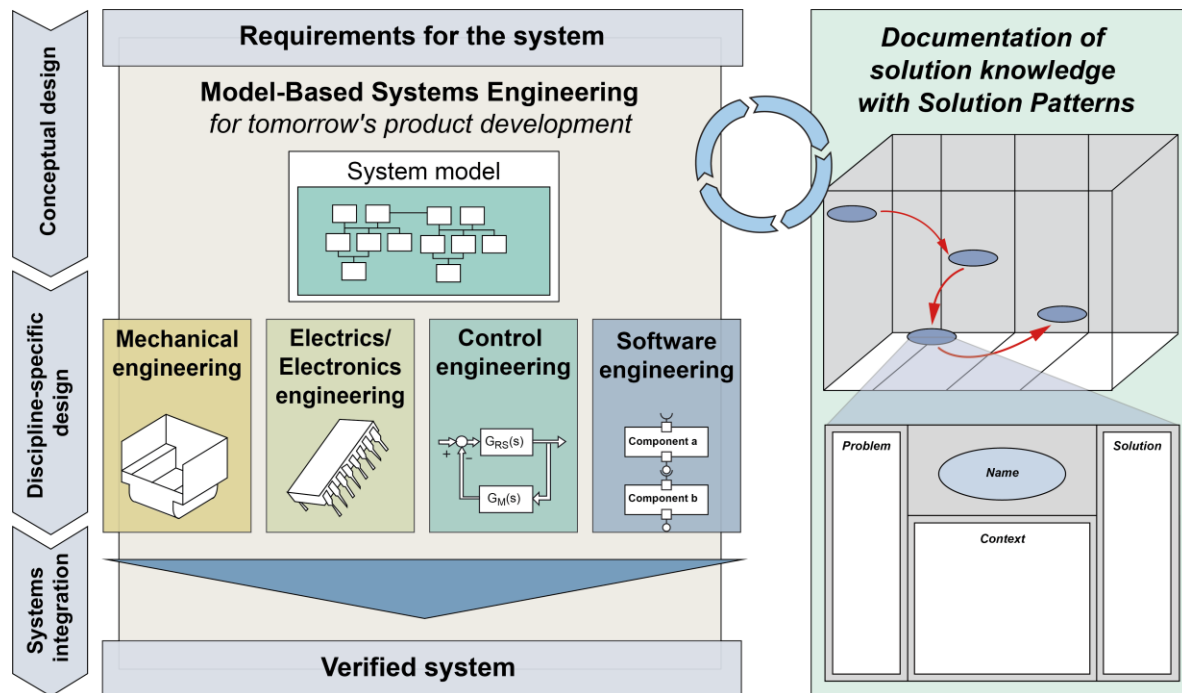


Figure 1. Procedure in the development of intelligent technical systems (left) with the help of documented solution knowledge (right)

In this context, the reuse of existing artifacts and thus the reuse of once successfully applied **solution knowledge** plays an increasingly important role (cf. Figure 1; right-hand side). A suitable knowledge management must be part of each enterprise as basis for creativity and innovations (Albers and Gausemeier, 2012; Probst et al., 2012; Gausemeier et al., 2012). An approach for the operationalization of knowledge management often described in the literature refers to so-called **Solution Patterns**. A Solution Pattern describes a problem as well as the core of the solution for this problem (Alexander et al., 1977). A multitude of documented Solution Patterns provide developers with impulses to reflect and expand their own thought patterns with the knowledge of the collective. On the basis of a comprehensive overview after Rising and further considerations by Deigendesch, Sanz/Zalewski and Suhm, the following fundamental advantages of Solution Patterns can be pointed out (Rising, 2000; Deigendesch, 2009; Sanz and Zalewski, 2003; Suhm, 1993):

- Transferability across disciplinary boundaries
- Improvement of communication through explicit knowledge representation
- Long-term documentation of solution knowledge
- Reduced complexity by breaking down extensive problems
- Increased efficiency through targeted reuse
- Promotion of creativity

In order to fully exploit the above-mentioned advantages, the use of Solution Patterns in Systems Engineering must take place at an early stage in the design of intelligent technical systems. This article describes a transfer of the Solution Pattern approach to Systems Engineering with a special focus on the interdisciplinary design of intelligent technical systems.

## 2. Fields of action and state of the art

The development of intelligent technical systems can only be successful through the interaction of different disciplines, as required by Systems Engineering. The aim is to develop concepts for the early and equal treatment of solution knowledge from the participating disciplines. In particular, the discipline-specific ways of thinking and approaches must be brought together, which is not possible at present without further ado. Against this background, there are two overarching fields of action.

### Field of action 1: Documentation of solution knowledge

A key to success lies in a language that is easy to understand and equally understood by all actors involved. Building on this, a uniform documentation of interdisciplinary relevant knowledge is necessary. This requires suitable structuring in the form of Solution Patterns. The documentation must take into account the outlined special features of the design and represent the knowledge of the disciplines involved accordingly. The goal must be a kind of construction kit consisting of various Solution Patterns for the design. Like concrete solutions, the abstract representation of the associated knowledge in the form of patterns is interdependent. In this context, the dimensions of mutual dependencies are to be emphasized and described transparently.

### Field of action 2: Application of Solution Patterns in Systems Engineering

The approaches established today in Systems Engineering do not meet the demand for cross-disciplinary knowledge management, e.g. with the help of Solution Patterns. They often only focus on a specific area or discipline; a holistic view of reusable product artefacts, some of which are interdisciplinary, is only rudimentary, if at all. The result is usually a combination of discipline-specific partial solutions, which rarely result in the best possible overall solution. A systematic approach is required to identify, document and apply Solution Patterns for systems design.

The **state of the art** analysed in the following is oriented towards the essential challenges of the two overarching fields of action.

#### Field of action 1:

- **Languages as means of expression for solution knowledge:** A multidisciplinary language for the design of mechatronic systems is, among others, SysML. The SysML (“System Modelling Language”) addresses the holistic and interdisciplinary modelling of technical systems. It is based on UML and extends the language by further aspects (Friedenthal et al., 2012; Object Management Group, 2018). An associated process model is SYSMOD [Wei08]. Within the SFB 614 the specification technique CONSENS for the description of the principle solution of self-optimizing systems was developed (Dorociak et al., 2014). Another language is PreMISE, which was developed by DLR to map an integrated product model (Schuhmann and Berres, 2011).
- **Structuring of Solution Patterns:** Cloutier (2006) presents a uniform structure. He structures a Solution Pattern in 16 categories. The application of the patterns ranges from business model development and the actual systems design to testing and validation. Weilkins (2014) also describes Solution Patterns with SysML. Further approaches from mechanical engineering, some of which are more discipline-specific, are provided by Suhm, Salustri or Diegendsch (Suhm, 1993; Salustri, 2005; Deigendsch, 2009).

#### Field of action 2:

- **Systematics for Solution Patterns in Systems Engineering:** A very generic approach for the use of Solution Patterns in the design of technical products is provided by Grabowski in the Universal Design Theory (Grabowski and Lossack, 2000; Grabowski and Leutsch, 2003). He describes a complete systematics with integrated use of object and process patterns. Pfister et al. (2011) provide an original approach from Systems Engineering with the design pattern metamodel. Further systematics describe Sanz and Salewski (2003) or Kalawski et al. (2013) with the emphasis on reusable process modules. With a special focus Amorin et al. (2017) support safety and security assurance in Automotive Systems Engineering with their pattern approach. Schindel and Peterson (2013) concentrate on the basic management process and the configuration of patterns.

**Conclusion:** An overarching approach that fully covers the addressed fields of action does not exist. Due to the large number of established languages, the solution to be developed must be language-independent in order to ensure the broadest possible applicability. The approaches to structuring are usually too detailed for the practicable application and represent the externalized knowledge often by textual descriptions instead of models (e.g. CONSENS). The existing systematics often follow the complex approaches of system theory or the approaches of classical mechanical engineering. There is no systematics that is tailored to modern Systems Engineering for the interdisciplinary design of complex technical products.

### 3. Solution knowledge in Systems Engineering

The need for action resulting from section 2 shows that model-based approaches to knowledge management are currently lacking in the interdisciplinary design of intelligent technical systems. In the following, a systematics as well as its components are presented satisfying this need.

#### 3.1. Uniform documentation in the form of Solution Patterns

According to Alexander's definition, the Solution Patterns are divided into four categories: name, problem, solution, and context (Alexander et al., 1977). Relevant solution knowledge of the individual categories is presented by aspects of the specification technique CONSENS. However, this can be replaced e.g. by the SysML or similar in order to achieve the widest possible distribution.

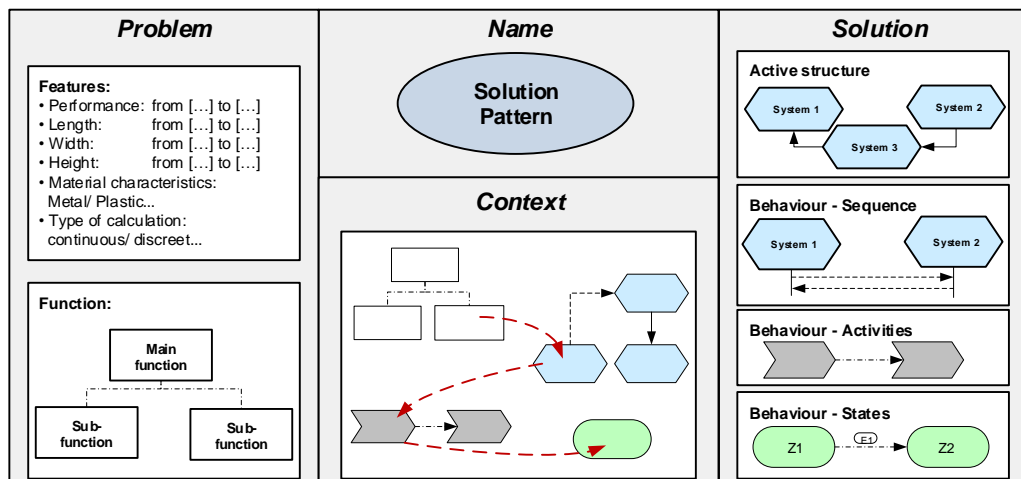


Figure 2. Uniform structuring of solution knowledge

**Name:** Each pattern is to be named unambiguously. The name already gives the developer first clues for the content when searching for a suitable pattern, if he is familiar with the corresponding area.

**Problem:** The aspects characteristics and functions are in regard to the definition according to Alexander synonymous with the problem to be solved. Both aspects should be described as solution neutral as possible. In combination with the Solution Pattern name, they form the selection criteria in the course of the systems design.

Characteristics are properties that are characteristic of the Solution Pattern. These are described both by parameter ranges in intervals and by specifications for material characteristics or for the movement type (rotatory/translatory). Functions: In the simplest case, a Solution Pattern fulfills exactly one function (cf. Principles of Construction) (Pahl et al., 2013). Depending on the complexity of the problem, a Solution Pattern can also fulfill several functions. The functions are mapped using a function hierarchy.

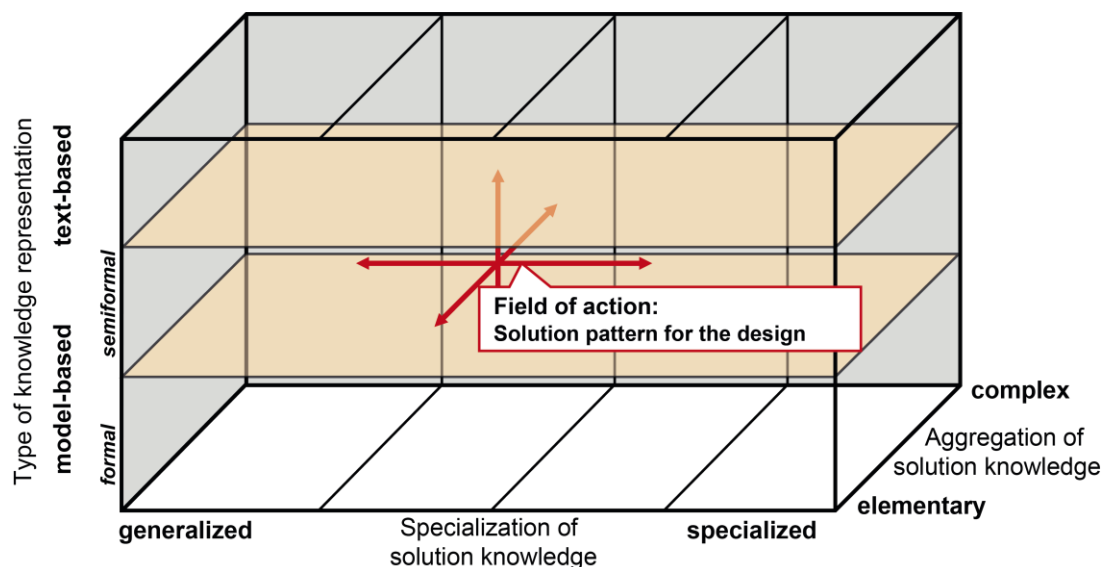
**Solution:** The description of the solution consists of the partial models for mapping the behavior and the active structure. Depending on the underlying problem, the mentioned partial models can also suffice individually. Active Structure: The active structure maps the system elements and their relationships to each other. The representation of the principle mode of action of the Solution Pattern illustrates how the previously defined functions are fulfilled. Behavior: The behavior models complete the solution description. The specification of the behavior is particularly relevant for Solution Patterns with a focus

on software engineering. Depending on the focused behavior, the information is represented by activities, states or sequence diagrams. With the help of behavior activities it can be specified in which order a system element executes its function(s). In the aspect Behavior States states and state transitions are modeled, which usually change after execution of an activity. In the model Behavior Sequence interactions between two or more system elements are mapped. The focus is placed on the temporal sequence in which, for example, messages are exchanged.

**Context:** The aspect context contains applications in which the Solution Pattern has already been used successfully. The specification of a Solution Pattern has at least one application example.

### 3.2. Multidimensional knowledge space

The underlying problems and the analysed state of the art have shown that Solution Patterns are characterized along different dimensions. This is particularly necessary in order to ensure the consistent use of Solution Patterns and the adequate mapping of interdependencies. Accordingly, Solution Patterns that are relevant for systems design as well as for upstream and downstream design steps can be described and distinguished from each other along the three dimension “**type of knowledge representation**”, “**specialization of solution knowledge**” and “**aggregation of solution knowledge**” (cf. Figure 3).



**Figure 3. Multidimensional knowledge space for the documentation of Solution Patterns and their interactions**

**Type of knowledge representation:** The basic prerequisite for an efficient integration of relevant solution knowledge into the design is the representation of the knowledge in a suitable terminology. In this case, suitable means that the knowledge is represented analogously to the language or terminology used in the corresponding design step at the time the Solution Pattern is used. In this respect, a distinction can be made at the highest level between text-based, document-based and model-based representation. Depending on the time of development and the intended use, the models differ in their degree of formalization (semiformal and formal models of the systems design).

**Specialization of solution knowledge:** Specialization, in which solution knowledge is documented, determines the creative performance of developers. The higher the degree of generalization of a Solution Pattern, the less it gets lost in unnecessary detail questions, especially at the beginning of product development. Developers often have great difficulties in deviating from known patterns of thought and in some cases in taking into account non-expert solution principles in the accomplishment of tasks. This leads to a limitation to already successfully implemented solutions. The innovation potential of existing technologies is therefore not fully exploited. Due to the high time and cost pressure, the developers leave a solution-neutral way of thinking too early and weigh up different solution concepts too little. A classification of the Solution Patterns according to the dimension of the

degree of specialization counteracts this. The more generalized a Solution Pattern is, the larger is its application context.

**Aggregation of solution knowledge:** The degree of complexity of Solution Patterns accompanies the degree of complexity of the underlying (partial) solutions, which form the starting point for the definition of Solution Patterns. Thus, more complex and elementary Solution Patterns result analogous to the structuring and aggregation of systems and subsystems. The knowledge space offers developers the possibility to record relations regarding the degree of complexity relative to each other.

### 3.3. Systematics for the application of Solution Patterns exemplified by two cooperating delta robots

In the following, a systematics for the use of Solution Patterns in the design of intelligent technical systems is presented. In the sense of VDI 2206 (2004), the approach of a function-oriented design is taken up and anchored directly in Model-Based Systems Engineering. It concentrates primarily on the reuse of established solution knowledge. The procedure model describes well-established design steps taking into account the novel, partly multidisciplinary Solution Patterns. It is divided into five successive phases (cf. Figure 4).

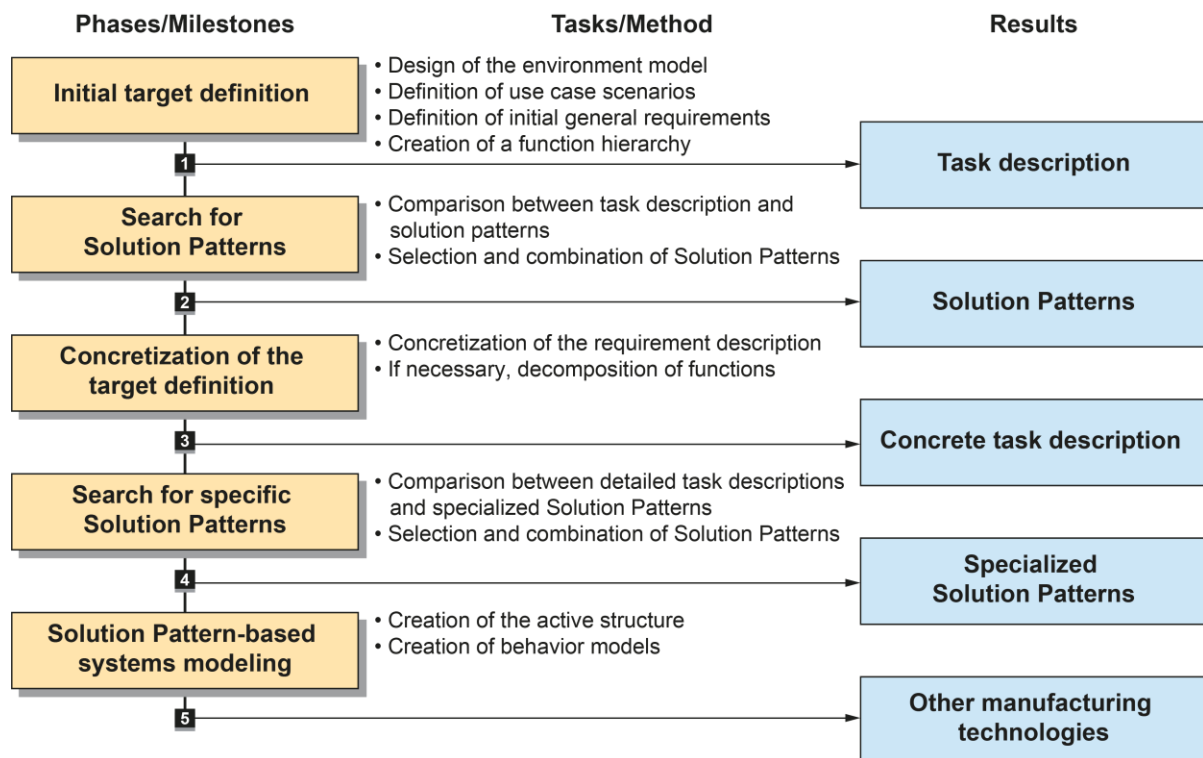


Figure 4. Procedure model for a Solution Pattern based systems design

All results are illustrated by the application example “two Cooperating Delta Robots” (cf. Figure 5). It consists of two identical robots with delta kinematics and three electric drives. The devices centre 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. At the beginning exercise, the robots do not know the parameters of the ball and which of them gets the ball from the user. Therefore, the robot that gets the ball from the user first has to juggle the ball to determine the balls parameters. After that, this robot has to ensure that the other robot is ready before hitting the ball to it and is aware of the balls parameters. As soon as they are rallying, they have to exchange information under real-time constraints, e.g. the predicted trajectory of the ball. At the end exercise, the user commands one of the robots to end the rally. This robot has to hit the ball into a box and has to inform the other robot.

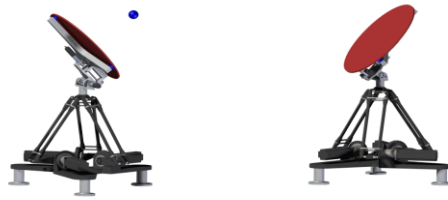


Figure 5. Application example “two cooperating delta-robots”

### Phase 1: Initial target determination

The aim of the first phase is a basis on which an interdisciplinary development team can search for potential Solution Patterns. An analysis of the environment of a system under development is followed by the definition of application scenarios. Requirements for the technical system are specified on the basis of the knowledge gained and the market and customer requirements already developed in strategic product planning. The documentation takes place in the form of a list of requirements (Pahl et al., 2013). A function hierarchy is then derived. In the first phase, attention must be paid to a solution-neutral formulation so that the developers do not unconsciously commit themselves to Solution Patterns known to them. The result is a first description of the task.

### Phase 2: Search for Solution Patterns

The aim of the second phase is to roughly restrict the available search space. The development team navigates through the multidimensional knowledge space in search of suitable Solution Patterns. The basic direction is from generalized to specialized. Depending on the strategy, the search for both elementary and complex Solution Patterns can be started. During the search, the requirements and functions specified in the first phase are compared with the problem description in the Solution Patterns. Figure 6 illustrates the interaction using the example of delta robots. A first search shows that the Solution Patterns “electric drive”, “hydraulic actuator” and “pneumatic actuator” are found for the function “generate motion”. Ultimately, each actuator fulfils the required partial function. Also the consideration of the example requirement “connected load max. 400 V” would limit the available Solution Patterns in the search space only conditionally. At this point in the procedure, a decision on the implementation of the system is necessary for the first time.

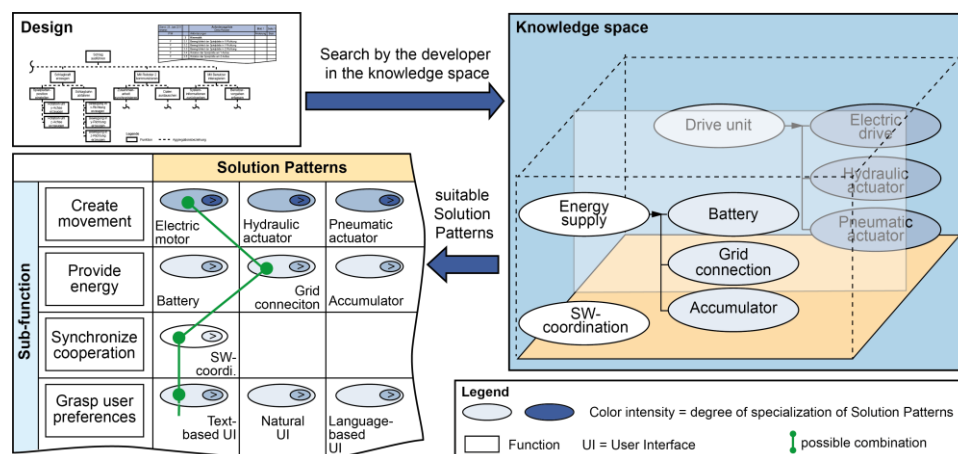


Figure 6. Preselection of possible Solution Patterns

In the final step, suitable Solution Patterns are selected in the morphological box and analyzed with regard to their combinatorial compatibility. This process still requires experience and can only be supported to a limited extent by information within a Solution Pattern.

### Phase 3: Concretization of the target definition

The selection and combination of Solution Patterns in phase 2 usually leads to new findings in the development team due to the characteristics provided. The characteristics which are assigned in the

selected Solution Patterns require a concretization of the goal definition specified in phase 1. It may be necessary to concretize existing requirements or to analyze restrictions. Furthermore, new requirements can be added by the characteristics.

Analogous to the requirements, the functional description by means of function hierarchy is decisive for the problem clarification. A concretization of the requirements is sometimes accompanied by a refinement of the functional hierarchy. By further hierarchizing the functions to be fulfilled, the desired system behavior can be concretized across disciplines. The choice of a Solution Pattern influences the number of subfunctions to be fulfilled and requires a new search.

As a result of the phase “concretization of the target definition”, a task description is available, consisting of a concretised list of requirements and function hierarchy.

#### Phase 4: Search for specialized solution patterns

The search for specialized Solution Patterns follows the same procedure as the first search in Phase 2. The starting point is the detailed task description and the already selected Solution Patterns. The developers are supported by the guided search gradually in limiting the search space so that the largest possible number of alternative Solution Patterns is available to them at any time.

During the preselection, for example, the Solution Pattern “electric drive” was selected to fulfill the function “Generate motion”. This also includes the function “convert electrical energy into mechanical energy”. The morphological box is supplemented by the extended functionalities of the Solution Patterns. Further functions have been added due to the preselection in the course of the advancing concretization. Examples are the functions “regulate current flow” and “convert rotation into translation” in Figure 7. For example, for the sub-function “Initiate game start” there is a corresponding Solution Pattern “Initiate information exchange” with the assignment of rights to ports of the type “peer to peer”. The position in the knowledge space shows that this is an aggregated Solution Pattern.

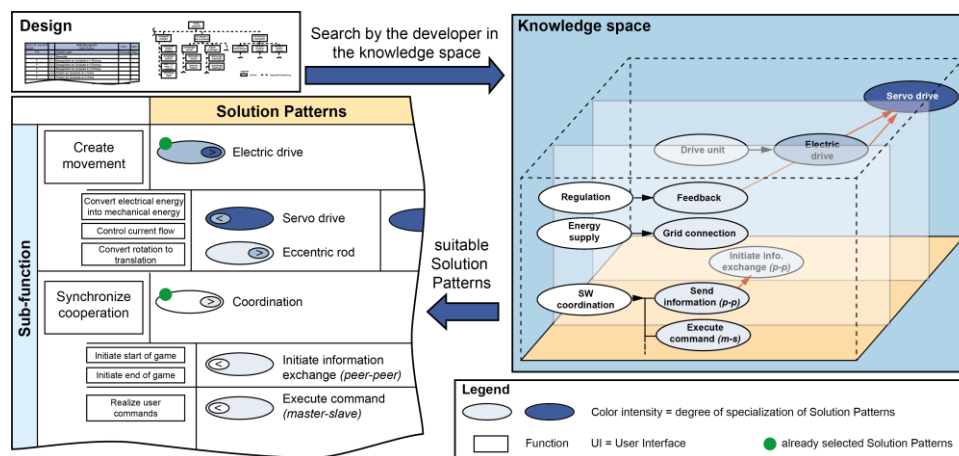


Figure 7. Detailed search for Solution Patterns

In a final step, the Solution Patterns are again combined with each other. The developers take all information about the system into account and include the results in the selection and combination.

#### Phase 5: Solution Pattern based systems modeling

The final phase of the procedure model is the Solution Pattern-based systems modeling. The combination of the Solution Patterns from phase 4 marks the starting point for the development of structural and behavior models at the interdisciplinary overall system level. Each selected Solution Pattern contains generalized information for the respective problem, which supports the developer in the synthesis. This leads to significant time savings when modeling the partial models.

When creating the effect structure, the system elements and internal relationships within a pattern are retrieved from the knowledge space for each Solution Pattern (cf. Figure 8). Provided Solution Patterns have to be adapted to the specific problem of the development task. This is done by modifying the



system elements and the flow relationships. The system elements of the Solution Patterns are to be combined to a compatible overall system. In addition and analogous to the modelling of the active structure, the developers specify the behavior on the basis of the generalised behavior models (e.g. behavior states) of the Solution Patterns. The partial models are created interactively. The final result is a multidisciplinary specification of the concept of an intelligent technical system.

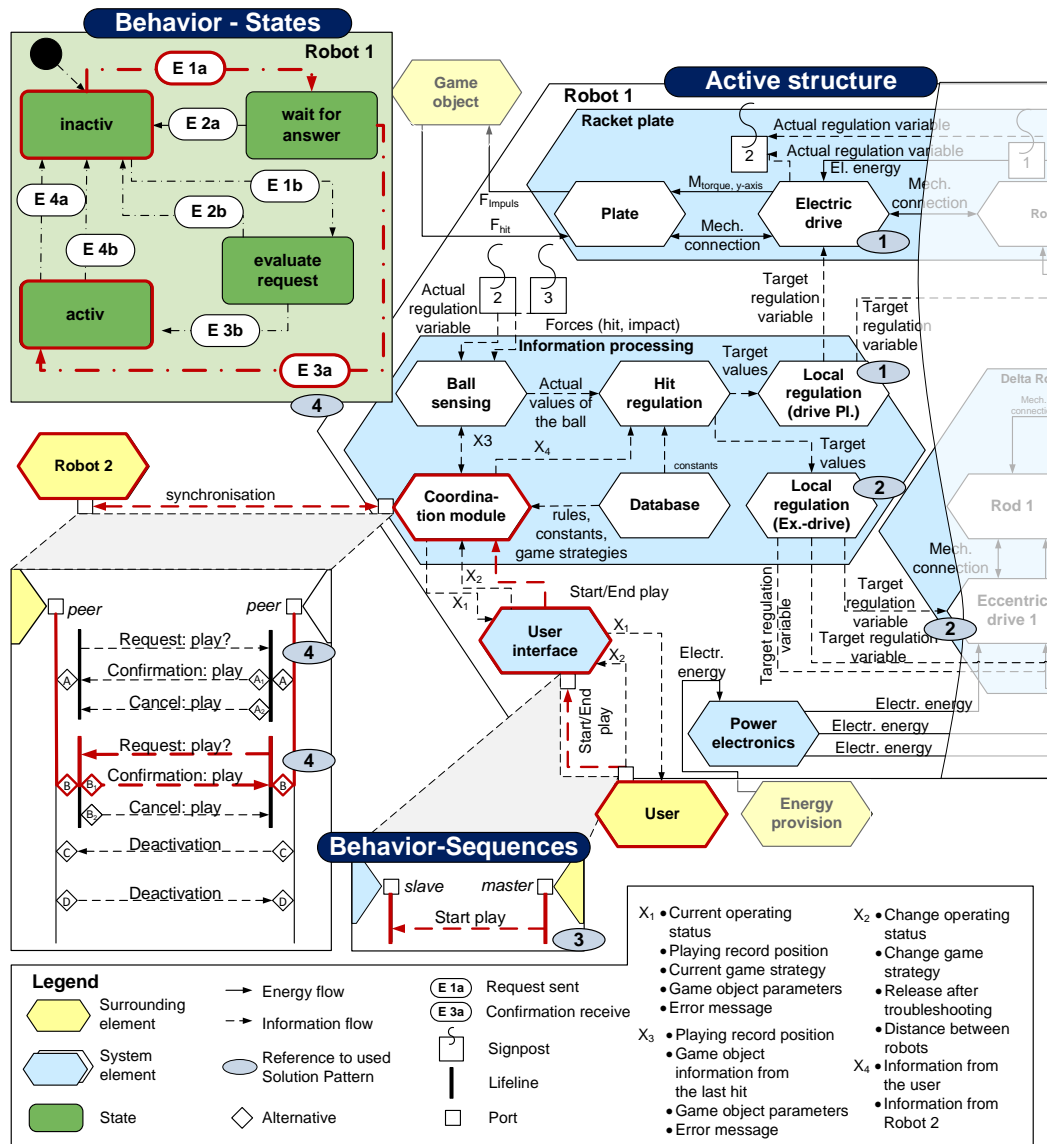


Figure 8. Excerpt of the principle solution of the example “Cooperating Delta Robots”

#### 4. Summary and outlook

For the development of intelligent technical systems, the documentation and reuse of established solution knowledge is a substantial guarantee of success. A promising approach are **Solution Patterns** that represent abstract problem-solution pairs. In order to avoid time- and cost-intensive iterations at a late stage of development, one focus lies on the early consideration of relevant solution knowledge in the context of systems design. For this purpose, the present article describes assistance for the **documentation of solution knowledge for systems design**. Using the CONSENS specification technique, a uniform structuring of Solution Patterns takes place and sets them in relation to each other in a multidimensional knowledge space along three dimensions. An associated **systematics for a Solution Pattern-based systems design** serves as a guideline, by means of which interdisciplinary development teams are enabled to develop system concepts using established solution knowledge. The

overriding goal of future research work with a longer horizon should be the Solution Pattern-based development of technical systems that includes neighboring areas such as strategic product planning.

## References

- Albers, A. and Gausemeier, J. (2012), "Von der fachdisziplinorientierten Produktentwicklung zur Vorausschauenden und Systemorientierten Produktentstehung", In: Anderl, R., Eigner, M., Sandler, U. and Stark, R. (Eds.), *Smart Engineering: Interdisziplinäre Produktentstehung, acatech DISKUSSION, April 2012*, Vol. 15, Springer, Berlin, Heidelberg, pp. 17-29.
- Alexander, C. et al. (1977), *A pattern language: Towns, buildings, construction, Center for Environmental Structure series, Vol. 2*, Oxford Univ. Press, New York.
- Amorim, T. et al. (2017), "Systematic Pattern Approach for Safety and Security Co-engineering in the Automotive Domain", In: Tonetta, S. (Ed.), *Computer Safety, Reliability, and Safety, International Conference on Computer Safety, Reliability and Security, 36th International Conference, SAFECOMP 2017*, Trento, Italy, September 13-15, 2017, pp. 329-342. [https://doi.org/10.1007/978-3-319-66266-4\\_22](https://doi.org/10.1007/978-3-319-66266-4_22)
- Cloutier, R. (2006), *Applicability of patterns to architecting complex systems, [Dissertation]*, Stevens Institute of Technology, New York.
- Deigendesch, T. (2009), *Kreativität in der Produktentwicklung und Muster als methodisches Hilfsmittel, [Dissertation]*, Institut für Produktentwicklung (IPEK), Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Dorociak, R. et al. (2014), "Specification Technique CONSENS for the Description of Self-Optimizing Systems. 4.1", In: Gausemeier, J., Ramming, F. and Schäfer, W. (Eds.), *Design Methodology for Intelligent Technical Systems*, Springer-Verlag Berlin Heidelberg, pp. 119-127.
- Friedenthal, S., Moore, A. and Steiner, R. (2012), *A practical guide to SysML: The systems modeling language*, 2nd ed, Morgan Kaufmann, Waltham, MA.
- Gausemeier, J., Lanza, G. and Lindemann, U. (2012), *Produkte und Produktionssysteme integrativ konzipieren*, Carl Hanser Verlag GmbH & Co. KG, München. <https://doi.org/10.3139/9783446429857>
- Grabowski, H. and Leutsch, M. (2003), "Integrierte Produkt- und Produktionsmodelle als Grundlage für eine wissensbasierte Produktentwicklung", In: Nagl, M. (Ed.), *Modelle, Werkzeuge und Infrastrukturen zur Unterstützung von Entwicklungsprozessen*, Wiley-VCH, Weinheim, pp. 45-58.
- Grabowski, H. and Lossack, R. (2000), "The Axiomatic Approach in the Universal Design Theory", *Proceedings of the First International Conference on Axiomatic Design (ICAD2000)*, Cambridge, 21.-23. Juni 2000, Cambridge, USA.
- Kalawsky, R.S. et al. (2013), "Using Architecture Patterns to Architect and Analyze Systems of Systems", *Procedia Computer Science*, Vol. 16, pp. 283-292.
- Object Management Group (2018), *OMG: Systems Modeling Language (OMG SysMLTM) – Version 1.6*.
- Pahl, G. et al. (2013), *Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung; Methoden und Anwendung*, Springer, Berlin, Heidelberg.
- Pfister, F. et al. (2011), "A Design Pattern meta model for Systems Engineering", *18th IFAC world congress 2011: Milan, Italy, 28 August - 2 September 2011*, Vol. 44, Curran, Red Hook, NY, pp. 11967-11972.
- Probst, G.J.B., Raub, S.P. and Romhardt, K. (2012), *Wissen managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen*, Gabler, Wiesbaden.
- Rising, L. (2000), *The pattern almanac 2000, Software patterns series, Addison-Wesley*, Mass, Boston.
- Salustri, F.A. (2005), "Using Pattern Languages in Design Engineering", In: Samuel, A.E. (Ed.), *Engineering design and the global economy: 15th International Conference on Engineering Design - ICED 05, 15-18 August 2005*, Melbourne, Australia, The Design Society, Melbourne.
- Sanz, R. and Zalewski, J. (2003), "Pattern Based Control Systems Engineering – Using Design Patterns to Document, Transfer and Exploit Design Knowledge", *IEEE Control Systems*, Vol. 23 No. 3, pp. 43-60.
- Schindel, W. and Peterson, T. (2013), "Introduction to Pattern-Based Systems Engineering (PBSE): Leveraging MBSE Techniques", *Proceedings of the 23rd Annual International Symposium of the International Council on Systems Engineering (INCOSE 2013)*, Philadelphia, USA, June 24-27, 2013, John Wiley & Sons, Hoboken, pp. 1639-1639. 10.1002/j.2334-5837.2013.tb03127.x
- Schuhmann, H. and Berres, A. (2011), "Modellbasierter Systementwurf mit dem PrEMISE-Modell", In: Maurer, M. and Schulze, S.-O. (Eds.), *Tag des Systems Engineering: Komplexe Herausforderungen meistern*, Hanser, München, pp. 35-44.
- Suhm, A. (1993), *Produktmodellierung in wissensbasierten Konstruktionssystemen auf der Basis von Lösungsmustern, [Dissertation]*, Fakultät für Maschinenbau, Universität Karlsruhe, Aachen.
- VDI (2004), *Entwicklungsmethodik für mechatronische Systeme No. 2206*, Verein Deutscher Ingenieure, Berlin.
- Weilkins, T. (2013), *Two kinds of Patterns*. [online] Model-Based Systems Engineering. Available at: <http://model-based-systems-engineering.com/2013/02/07/two-kinds-of-patterns> (accessed 15 October 2014).