This is an author-created version.

The final publication is available at link.springer.com.

DOI 10.1007/978-81-322-1050-4_25

Meta-Design Catalogs for Cognitive Products

Torsten Metzler, Michael Mosch,
Udo Lindemann

Institute of Product Development
Technische Universität München
Munich, Germany
Tel: +49-(0)89-289 15154
Fax: +49-(0)89-289 15144
{metzler | lindemann}@pe.mw.tum.de

Abstract: This paper presents a concept of meta-design catalogs for cognitive products. Mechatronic and cognitive products usually demand multidisciplinary hardware and software solutions and while catalogs exist to support domain specific-design, to date there is no support for finding non-obvious and alternative solutions for cognitive products. The meta-design catalogs for cognitive functions proposed in this paper provide a link between abstract functions and hardware/software making it possible to find non-obvious and alternative solutions. They also make re-use of existing solutions possible by abstracting from the specific to an abstract pattern.

Keywords: Design Catalog, Design Pattern, Cognitive Product, Cognitive Product Development

Introduction

Design catalogs and catalogs of design patterns, such as [1, 3, 4] were created to support engineers in finding non-obvious solutions, alternative solutions and to avoid inapplicable solutions or repeated solutions. They exist for different domains but address mainly domain-specific problems.

This paper presents a concept of meta-design catalogs for cognitive products that is appropriate to classify all types of solution patterns and thereby helps to transfer cognitive functions into cognitive products. Meta-design catalogs for cognitive products are necessary because cognitive functions currently are realized only through interdisciplinary solutions and can not be realized by domain-specific solutions alone. A complete decomposition of cognitive functions into elementary functions is not possible yet due to a lack of understanding of cognitive processes. Instead, solution patterns systematically stored in design catalogs provide the possibility to include knowledge of different domains and support the conceptual design of cognitive products because they can be re-used. By directly addressing cognitive functions a further decomposition into elementary functions is usually unnecessary.

After a brief definition of terminology, domain-specific patterns are compared with the aim to identify how design catalogs from different domains are structured, what they have in common and how they differ. Based on this, a holistic framework of meta-design catalogs for mechatronic and cognitive products and systems is derived. This structure includes different meta-design catalogs differentiated by the type of design catalogs, complexity and granularity. They are appropriate to classify all types of solution patterns and thereby help to transfer cognitive functions into cognitive products. To demonstrate how solution patterns can be identified in existing cognitive products and generalized for re-use in conceptual design of future cognitive products an example is presented. One solution pattern is explained at different levels of abstraction and allocated to the framework of meta-design catalogs for cognitive products.

Background

Design catalogs and catalogs of design patterns are information sources supporting the conceptual design process by re-using solutions. They help to find patterns that realize certain functions used in function structures and models describing a system- or product-concept. They exist in several domains and vary mainly by the functions they provide patterns for and their level of abstraction. Alexander et al. [1] say that “each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice”.

In mechanical engineering, design catalogs are tailored to systematically support the conceptual product development process and have to fulfil the following requirements: quick accessibility of information, ease of use, customizability, integrity within given boundaries, validity, upgradeability and consistency [2]. They are subdivided in three categories according to [2, 3 and 4]: object catalogs, process catalogs and solution catalogs. Object catalogs contain available objects, e.g. bearings and screws, and are independent from specific design problems. They do not contain all principle solutions for a design problem but describe objects and their characteristics generically. Process catalogs contain processes, rules and process steps and are related to objects. Each solution catalog contains a variety of patterns for specific design problems and constitutes a source for alternative solutions [3]. Obviously, the three catalog types are related to each other, e.g. objects of an object catalog can help to generate new patterns for a solution catalog according to a process described in a process catalog. However, one object may be included in several patterns in different solution catalogs and adaption may be possible using different processes.

Design catalogs in mechanical engineering have mainly been developed for elementary functions, e.g. to convert, to increase/decrease, to mix/separate, etc. because it is assumed that all functions can be decomposed into them [3, 4]. However, they also include frequently re-appearing functions [4]. Homogeneous internal catalog structures are important with respect to convenience and clarity [1]. Therefore, most catalogs are structured in a similar way. They consist of an index structuring the content, a main part describing the solutions and an access part explaining the properties of the solutions. Recent research extends the access part by adding disturbances and robustness ratios to physical effect catalogs in order to consider them in the conceptual design phase of a product while avoiding additional effort and cost [5].

The development of electric and electronic systems and products is, when compared to mechanical engineering, more object-oriented. In electronic systems, objects, e.g. resistors, transistors and integrated circuits, are stored in object catalogs. Solution catalogs in this domain contain for example adaptable circuit diagrams for operational amplifiers that amplify a differential input voltage to a much higher output voltage. General circuit diagrams in solution catalogs are adapted according to the particular problem by defining properties of basic hardware components, e.g. resistors and capacitors. According to design rules and processes that are stored in process catalogs (or even computer tools), for example the width of conductor lines or the spacing between lines can be calculated. These rules, processes and especially the tools allow inexperienced electrical engineers to design electronic components, for example "Very-Large Scale Integration Systems", because the design methodology is based on the electrical behaviour of circuit elements [6] and universal design rules. It is assumed that design catalogs for electrical engineering can be structured like those for mechanical engineering due to the above mentioned characteristics even though only object catalogs were found.

In computer science, design patterns are considered re-usable elements of software supporting the conceptual software design. Patterns in object-oriented software provide generic solutions in terms of objects and interfaces. They vary in granularity and level of abstraction. Similar to design catalogs design patterns are characterized by the "pattern name", the "problem" they address, the "solution" they provide and the "consequences" of their application. Further, they are classified by their purpose and their scope and grouped in families of related patterns [7].

Experts re-use successful solutions and base new designs on prior experience without having to re-discover the whole problem [7]. Design patterns help inexperienced software developers to do the same based on the experts sharing their knowledge.

Looking at the status-quo of design catalogs and catalogs of design patterns several limitations exist. First, design catalogs in [4] are still mainly paper-based even though Derhake [8] proposed a system to computationally create and represent design catalogs. Computational search for patterns is limited and synthesis is not adequately supported. For this reason, the ease of use is low and the speed of search is slow. The authors assume a negative impact for existing paper-based design catalogs regarding the acceptability and usability due to these issues. In addition, design catalogs in the mechanical engineering domain are predominantly available in German language hindering broad usage. Koller et al. [3], Roth [4] and VDI 2222 [2] demand a consistent and conceptually accurate catalog structure. This demand must be extended to design catalogs and catalogs of design patterns of all domains in mechatronics to enable a search for multidisciplinary patterns. The use of solution patterns was already proposed for mechatronic systems in [9].

Terminology

In this section the terms used throughout the paper are explained. This is important since every domain uses their own terminology with respect to re-usable solutions and the paper intends to be understandable for interdisciplinary product development teams.

A *design catalog* contains knowledge about engineering design. It is systematically developed and makes knowledge available for everyone, independent of the experience and knowledge of individuals [4]. A difference between a collection of solutions and a design catalog is drawn regarding the level of completeness, the systematic structure and the accessibility within the design process, all of which are more restrictive for design catalogs [4, 3]. The term design catalog is predominantly used in the electro-mechanical domain with a strong focus on mechanical design.

Design patterns describe general design problems and their solutions [1, 7]. According to Gamma et al [7] a design pattern "names, abstracts, and identifies the key aspects of a common design structure". They can vary in granularity and the level of abstraction that is influencing the reuse of every design pattern. Design patterns with a high level of abstraction are likelier to be reused. The term design pattern is mainly used in the domains of computer science and architecture with increasing interest in mechatronic design.

Catalogs of design patterns organize design patterns to make systematic and efficient retrieval possible. They correspond to design catalogs in the electro-mechanical domain.

The term *solution patterns* is used in this paper to describe a generic solution of a problem that is solved in a domain-spanning way and is appropriate for re-use. Typically, the implementation of cognitive functions requires a domain-spanning solution. The new term "solution pattern" is introduced to avoid confusion with terms that have different meanings in different domains and to point out when solution patterns for cognitive functions are discussed. Solution patterns can vary in complexity, granularity and type of solution.

Solution patterns are stored in *meta-design catalogs for cognitive products*. The meta-design catalogs provide a framework for all solution patterns, similar to design catalogs and catalogs for design patterns. In addition they are linked with domain-specific design catalogs and catalogs for design patterns allowing a breakdown into domain-specific subfunctions.

In contrast to all types of above mentioned patterns a *solution* is one problem-specific occurrence of a pattern.

Meta-Design Catalogs for Cognitive Products

Section four first compares design catalogs available in engineering disciplines and catalogs of design patterns from architecture and software design. Based on this comparison, functions in meta-design catalogs are described. Finally the framework of meta-design catalogs for cognitive products is presented.

Comparison of Design Catalogs and Catalogs of Design Patterns

A comparison of design catalogs and catalogs of design patterns from different domains leads to the following assumptions. They all serve one main purpose: to find solutions for general design problems and avoid repeating the same work. In general, they use a semi-formal description of the initial situation, e.g. elementary functions in [3], electrical behavior in [6] or pattern name in combination with problem description in [7]. The solution pattern usually is broken down into a description including elements like name, problem, solution and consequences.

The common goal of design catalogs and catalogs of design patterns is to make the re-use of successful designs and architectures easier and so help designers to find design alternatives quickly [7, 3].

Graphical notations solely are not sufficient to represent solution patterns; neither in engineering design nor in object-oriented software design [7, 3]. Nevertheless, they are important and useful to foster understanding of an abstract textual description in all domains and can provide concrete examples.

Functions in Design Catalogs

To date, design catalogs and catalogs of design patterns address mainly design problems related to single domains. Nevertheless, element design and system design are inseparable [6] and systems engineers manage the development process of complex engineering projects. This paper is about meta-design catalogs for cognitive products. These are tangible and durable things consisting of a physical carrier system with embedded mechanics, electronics, microprocessors and software [10]. The surplus value is created through cognitive functions, e.g. to perceive, to learn and to act [11], enabled by flexible control loops and cognitive algorithms. Cognitive functions are the elementary functions enabling cognition as a whole and heavily rely on a software component but nevertheless are regularly realized through the combination of solution-elements from different domains. This already indicates that common, domain-specific design catalogs and catalogs of design patterns are not appropriate to search for high-level solution patterns of cognitive functions. First order cognitive functions, e.g. perceive, learn and think, are very abstract and neither support a straightforward search for solution patterns nor allow an easy decomposition into subfunctions with known solution patterns. Nevertheless, solution patterns are needed for each cognitive function that can be adapted to specific design problems in the conceptual design of cognitive products.

Koller et al. [3] and Gausemeier et al. [9] show that product functions can be decomposed into elementary functions, e.g. according to Pahl et al. [12]. Is it possible to similarly decompose cognitive functions into elementary functions? If so do these correlate with common elementary functions that can be found in existing design catalogs of different domains? If such a decomposition is possible, it is not intuitive and even cognitive scientists or neuroscientists can not precisely tell what the elementary functions are that are involved in cognitive processes in human beings. For example, Rees [13] says that "seeing is not perceiving" but can not name the extra "bit" required for perception. The authors assume that a full decomposition of human cognition into elementary functions is, with the current knowledge of human cognitive processes, not possible and for this reason not realized in a cognitive technical system to the same extent yet. However, cognitive functions are imitated and realized in CTSs and cognitive products.

Functional decomposition of imitated cognitive functions always points to solution patterns in catalogs of different domains without considering interrelations among them. By developing meta-design catalogs for cognitive functions interrelations can be considered among corresponding sections of the catalogs. It is also expected that, in future, objects will be available off the shelf that conduct cognitive functions instead of elementary functions. Therefore new object catalogs are required, capable of linking abstract solution patterns to real solutions.

Framework of Meta-Design Catalogs for Cognitive Functions

In this paper a pragmatic way to support the design of cognitive products is proposed. The approach is to develop a framework of meta-design catalogs for cognitive products that links generic solutions to related subsets of solutions in domain-specific design catalogs and catalogs for design patterns. Relating the meta-design catalog to domain-specific design catalogs is important because it allows product developers to use design catalogs they are familiar with, they do not have to be updated independent from the domain-specific catalogs and it is less demanding computationally.

The catalog types used in mechanical engineering already provide a meaningful catalog classification for design catalogs of cognitive products as well as other domains. The above mentioned catalog types constitute the first dimension of Figure 1: type of catalog. Process catalogs for cognitive products contain processes, rules and process steps describing for example how to develop a cognitive product, how to connect cognitive functions or how to decompose cognitive functions into cognitive subfunctions. Object catalogs for cognitive products are empty at the moment because integral objects accomplishing cognitive functions independent from specific design problems are not available. For this reason process catalogs are not yet linked to object catalogs targeted at cognitive products. In the future, integral objects accomplishing cognitive functions may exist and will be integrated in the object catalogs. Solution catalogs for cognitive products contain solution patterns for specific design problems related to artificial cognition and constitute a source for alternative solution patterns. The authors currently work on a modeling approach for cognitive products using the systems modeling language (SysML) and expect the identification of patterns realizing different cognitive functions. Instead of linking process catalogs and solution catalogs for cognitive products with empty object catalogs directly, they are linked to domain-specific design catalogs including object catalogs.

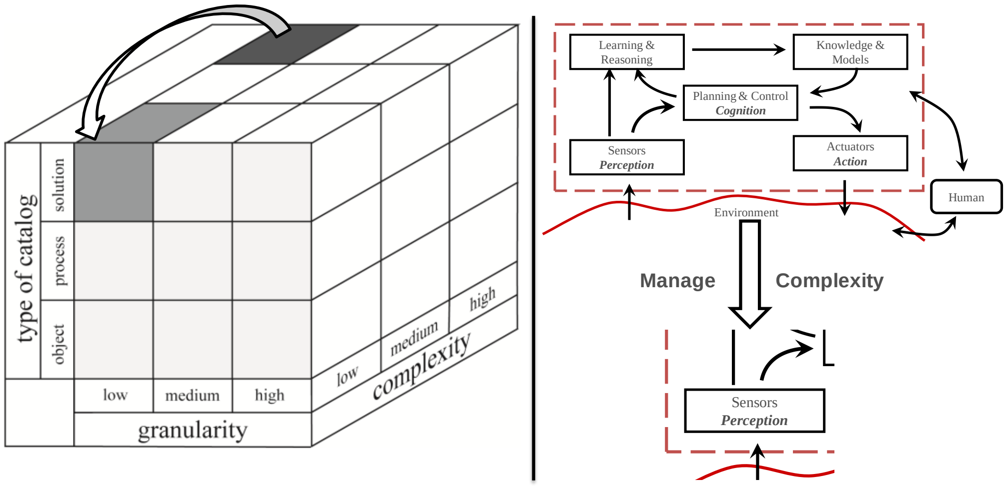

Figure 1 Framework of all Design Catalogs for Cognitive Products (left) and the Management of Complexity (right).

Further, [4] and [2] distinguish design catalogs regarding their complexity whereas [7] use the term granularity. The authors consider both differentiations important for the following reasons. [4] and [2] determine complexity by the number of relations among elements in a system or product. This means that a cognitive product with several interlinked cognitive functions is usually more complex than a system with an isolated cognitive function. Granularity describes how detailed a system is broken down or decomposed into subfunctions and therefore a higher granularity is characterized by an increase of interlinked functions in the model. In conclusion it is assumed that a higher granularity increases the model complexity but not the inherent system complexity because only the level of abstraction is changed. In order to create holistic design catalogs for cognitive functions different levels of complexity as well as different levels of granularity have to be covered. This helps to manage system and product complexity and to increase model granularity by breaking down functions into elementary functions that can then be linked to single-domain design catalogs.

Figure 1 (left) shows all possible design catalogs for cognitive products structured according to catalog types, complexity and granularity. Solution patterns with high complexity and low granularity, e.g. in solution catalogs, are linked to design catalogs with a lower complexity and low granularity. This is visualized in Figure 1 (left) with the black arrow pointing from the left side of the building block in the back to the left side of the building block in the front. An example for a solution pattern with high complexity and low granularity is for example a generic system architecture for CTS as proposed in [14], see Figure 1 (right).

Beyond the classification of meta-design catalogs for cognitive products their internal structure is of great importance because they need to cover solutions including elements of different domains. By comparing the internal structure of design catalogs from different domains, the following issues have to be considered to create holistic and unambiguous design catalogs for cognitive products that are valid independent of complexity, granularity and catalog type:

- a (formal) description of the problem/function and the solution is given
- solutions are accessible through a kind of index
- limitations of the solution space through parameters

Next, the advantages of existing design catalogs and catalogs of design pattern are combined to create a suitable catalog structure for cognitive products. Thus, at first a universally valid index of design catalogs for cognitive products is required and proposed according the taxonomy of cognitive functions [11]. The taxonomy of cognitive functions was created by analysing scientific publications from different domains with regard to cognitive capabilities and cognitive functions and structuring the found terms in an unambiguous way in a taxonomy. Using cognitive functions as an index for design catalogs seems appropriate because all kind of cognition can be traced back to them. Depending on the hierarchical level of the cognitive function that has to be realized technically, different links point to different solution patterns in the main part of the catalogs with different levels of abstraction. "to act", a very abstract cognitive function, points to a very generic solution pattern generally describing fundamental requirements as well as to sub-patterns providing more specific solutions for subfunctions of "to act", e.g. for "to interact" or "to move" (Figure 2). The main part of the catalog contains the solution patterns for the cognitive functions. Every solution pattern is first characterized by a name according to the cognitive function and a description of the problem that can be solved with it to avoid wrong applications of the pattern. Then the solution provided in the pattern is described. In addition, links to sub-patterns are included, in case the user is looking for something more detailed. Finally, the solution is broken down into domain-specific components of the solution, either objects in object catalogs, processes in process catalogs or tailored solutions in solution catalogs. A description of consequences concludes the every pattern including e.g. limitations of the pattern or possible disturbances influencing the solution.

Index

Primary Pattern Name	Link	Secondary Pattern Name	Link	…
to perceive	*link a*			
to learn	*link b*			
to understand	*link c*	to interpret (to construe, to see)	*link c.1*	…
		to solve (to work out, to figure out)	*link c.2*	
		to appreciate (to take account)	*link c.3*	
		…	*link c.n*	
to know	*link d*			
to think (to cogitate, to cerebrate)	*link e*	to reflect (to think over)	*link e.1*	…
		to reason	*link e.2*	
		to plan	*link e.3*	
		…	*link e.n*	
to decide	*link f*	to judge (to adjucate, to try)	*link f.1*	…
to act (to move)	*link g*	to react	*link g.1*	…
		to interact	*link g.2*	
		to move	*link g.3*	
		…	*link g.n*	
…	…	…	…	…

Generic Link
"link n"

Main Part

Pattern Name	"name"				
Problem Description	description of the problems that pattern "name" solves				
Solution Description	**Description**	**Sub-Pattern**	**Domain**	**Link to Domain-Specific Catalog**	**Description**
	general description of the solution for pattern "name"	link to sub-patterns (if no further sub-pattern is available continue with domain specific solutions/patterns)	**Mechanical Engineering**	link me	
			Electrical Engineering	link ee	
			Computer Science	link cs	
			…	…	
Consequence Description	description of consequences when applying pattern "name"				

Figure 2 Internal Structure of Design Catalogs for Cognitive Functions with Index and Main Part.

Application

This section shows how solution patterns of design catalogs for cognitive products are identified, using as an example cognitive product developed by an interdisciplinary student team, and how to allocate the solution patterns to the design catalogs. Afterwards, the identified solution pattern can be used to support the search for solutions in the conceptual design phase of future cognitive products.

The cognitive product from which solution patterns are identified is a toy called "Virtual Opponent Slot Car" and was developed in a class on the theme "Cognitive Toys" by a team of four students from ME and EE. The goal of the students was to develop a slot car toy that is fun, even if no human opponent is available. To achieve human like behavior and skills it was decided that the virtual opponent learns to drive around the track from the human player. For every segment the speed of the human-driven slot car is measured and the maximum speed is stored in order to recall it during a race. Since a game with optimal performance would become boring rather quickly, a tactic function was implemented that makes the virtual opponent act according to the human driver's performance. In case the human-driven slot car is behind the virtual opponent, it drives slower depending on the distance and in case the virtual opponent is behind the human-driven slot car it drives as fast as possible. This way, the two slot cars usually stay close together and a close finish situation is generated keeping the game exciting all the time.

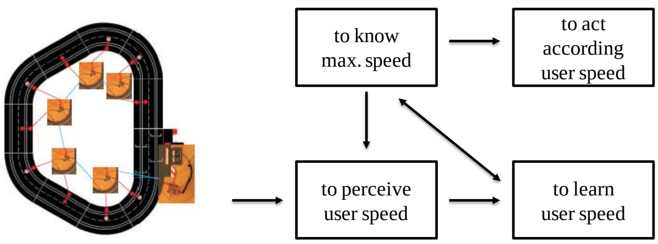

Figure 3 "Virtual Opponent Slot Car" (left) and cognitive function structure of the product (right).

A picture of the prototype is shown in Figure 3 on the left and the high level cognitive function structure on the right. The virtual opponent perceives positions of the two slot cars and the average user speed for every section of the track and compares it with the known maximum speed for that section. In case the actual speed is higher than the maximum known speed it learns the new maximum speed from the user. The virtual opponent acts according to its knowledge about the maximum user speed.

Pattern Name	**"to perceive"**				
Problem Description	Provide a solution about how a technical system becomes aware of something through senses. This pattern is relevant when a system is not only meant to do signal processing but has to consider context as well.				
Solution Description	**Description**	**Sub-Patterns**	**Domain**	**Link to Domain-Specific Catalog**	**Description**
	the action "to perceive" requires the sensing of signals in the environment and internally and process them according to the existing knowledge of the system and the context	*-to know* *-to sense* *-to process data*	**Mechanical Engineering**	-	-
			Electrical Engineering	*-sensor-catalogs* *-processor catalogs*	find appropriate sensor for the desired perception
			Computer Science	*- data processing algorithms*	find appropriate algorithm for the perception task
Consequence Description	- depending of what needs to be perceived different sensors are required - depending on what sensor is used different software is required - …				

Figure 4 Solution Pattern of "to perceive".

A frequently reoccurring cognitive function while developing cognitive products is "to perceive". Therefore, a solution pattern for "to perceive" has been abstracted from the virtual opponent describing the general problem, the related solution and consequences exemplarily. All parts of the solution pattern are included in Figure 4. The pattern name is *"to perceive"* according to the cognitive function and it is appropriate to solve problems related to "becoming aware of something through senses". The generic solution pattern describes how a technical system can perceive by identifying subfunctions that have been found in prototypes, e.g. the virtual opponent. The action to perceive requires the sensing of signals in the environment and internally and the processing of them according to the existing knowledge of the system and the context [15]. Sub-patterns are assumed for other functions, e.g. to sense or to process and other cognitive functions, e.g. to know. The consequences on this pattern are that, depending on what needs to be perceived, different sensors, processors and software, etc. are required. Linking this solution pattern with a domain specific pattern is partially possible, e.g. the link to object catalogs in electrical engineering containing sensors and processors is possible. The allocation of the solution pattern in the structure of meta-design catalogs for cognitive products has to consider the pattern type, complexity and granularity. The granularity in "to perceive" is low as well as the complexity. The solution pattern fits well into a solution catalog because it describes schematically how to make a technical system perceive something through senses. It neither describes an object nor a process. It is allocated to the light grey building block in the front of Figure 1.

Because the textual description above and Figure 4 are very abstract, an additional graphical representation of the pattern is considered helpful. This graphical representation has been developed in SysML (Figure 5). The model includes operators and flows. The flows define the inputs and outputs of the operators and the operators conduct an activity on the flow. The combination of flow(s) and one operator is considered here as one function. In the example several functions together accumulate to the cognitive function "to perceive". Cognitive functions and flows in the activity diagram shown in Figure 5 are further explained in [11].

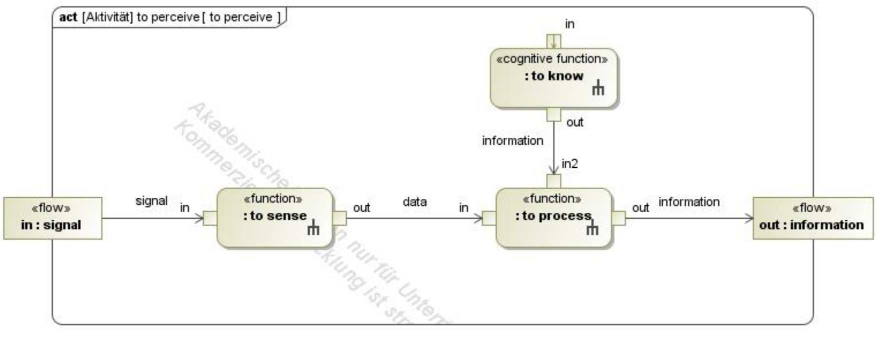

Figure 5 Graphical Representation of the Solution Pattern "to perceive" including Functions.

In Figure 6 components have been allocated to the subfunctions presented in Figure 5. These components are capable of realizing the functions they are allocated to. In the case of the solution pattern "to perceive" the components belong to other domains, e.g. sensors are electronic objects and can be found in object catalogs of the electrical engineering domain.

By adding components to the solution pattern the granularity of the system model is increased but the model complexity increases likewise. For this reason it is good to start with an abstract pattern and model and successively detail it throughout the development process.

Coming back to the sample application of the virtual opponent perception is realized through the functions and components described below. The function "to sense" is realized through Hall Effect sensors that are placed around the racing track and detect magnetic fields (“signals). They output data every time a magnetic field is within their range. A magnet is attached to the bottom to the slot car already to increase the traction. Therefore, no adaptation of the slot car was necessary. The processing of the sensor data is accomplished by a combination of the micro controller AT90USB162 mounted on an AVR-USB-162 development board from OLIMEX, a laptop and software. The software requires the sensor data and knowledge about the previous position as well as a definition about how the change in sensor data and position has to be interpreted. Therefore, knowledge about different sensor data has to be stored in the memory of the micro controller.

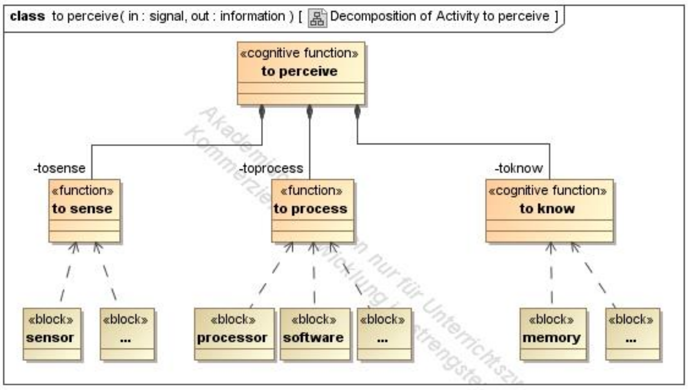

Figure 6 Graphical Representation of the Solution Pattern “to perceive” where Components are Attached to Functions.

Discussion

Meta-design catalogs for cognitive products support their conceptual design phase by providing generic solution patterns that can be reused. They close the current gap of missing solution patterns tailored to cognitive functions that can not be broken down to elementary functions yet and therefore are realized using interdisciplinary solutions. By directly addressing cognitive functions a further decomposition into elementary functions is usually unnecessary.

Instead of creating meta-design catalogs for cognitive products that contain domain-specific solution patterns or even objects at component level they are interlinked with domain-specific catalogs avoiding ambiguity. That is possible because domain-specific design catalogs and catalogs of design patterns are structured similar and are partially identical to the proposed meta-design catalogs.

To date, the use of meta-design catalogs for cognitive products is limited by the number of solution patterns stored inside and the paper-based structure. The number of solution patterns must be increased significantly in order to benefit from the meta-design catalogs. As a starting point the authors are going to analyze the cognitive products they already developed, extract solution patterns and integrate them in the meta-design catalogs. It is expected that enough solution patterns will be found to do some basic evaluation of the design catalog by testing it during the development of a new cognitive product.

As proposed earlier in the paper a software implementation of the meta-design catalog structure is necessary, supporting: the systematic integration of new solution patterns, the search for solutions according to the problem description at the required level of complexity, granularity and type of catalog, and an effective representation of the solutions with links to other, domain-specific catalogs. Ideally, files containing a model of a solution pattern can be included in the meta-design catalogs, e.g. SysML models, and an interface to the modeling tool is available enabling an easy integration of the solution into the own model.

Conclusion

This paper presents a framework of meta-design catalogs for cognitive products supporting the conceptual development. They are structured according to catalog type, complexity and granularity from an external viewpoint and each catalog has the same internal structure consisting of an index and a main part. The main part itself consists of a pattern name, a problem description, the solution pattern and consequences arising through the application of the solution pattern. Solution patterns managed in meta-design catalogs for cognitive products are linked to other solution patterns containing cognitive functions as well as to domain specific design patterns.

Acknowledgments

This research is part of the Cluster of Excellence, Cognition for Technical Systems – CoTeSys (www.cotesys.org), founded by the Deutsche Forschungsgemeinschaft (DFG).

References

1 Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S., "A Pattern Language", Oxford university Press, 1977.

2 VDI 2222 Page 2, "Design Engineering Methodics – Setting up and use of design catalogs", Beuth, 1982.

3 Koller, R., Kastrup, N., "Prinziplösungen zur Konstruktion technischer Produkte", Springer, 1998.

4 Roth, K., "Konstruieren mit Konstruktionskatalogen – Band II", Springer, 2001.

5 Mathias, J., Eifler, T., Engelhardt, R., Kloberdanz, H., Birkhofer, H., Bohn, A., "Selection of Physical Effects Based on Disturbances and Robustness Ratios in the Early Phases of Robust Design", International Conference on Engineering Design (ICED11), Vol. 5, pp. 324-335, 2011.

6 Whitney, D. E., "Why Mechanical Design Cannot be like VLSI Design", Research in Engineering Design, Vol. 8, pp. 125-138, 1996.

7 Gamma, E., Helm, R., Johnson, R., Vlissides, J., "Design Patterns – Elements of Reusable Object-Oriented Software", Addison-Wesley, 1995.

8 Derhake, T., "Methodik für das rechnerunterstützte Erstellen und anwenden flexibler Konstruktionskataloge", PhD Thesis. TU Braunschweig, 1990.

9 Gausemeier, J., Frank, U., Donoth, J., Kahl, S., "Specification technique for the description of self-optimizing mechatronic systems", Research in Engineering Design, Vol. 20, pp. 201-223, 2009.

10 Metzler, T., Shea, K., "Cognitive Products: Definition and Framework", International Design Conference – DESIGN 2010, pp. 865-874, 2010.

11 Metzler, T., Shea, K., "Taxonomy of Cognitive Functions", International Conference on Engineering Design (ICED11), Vol. 7, pp.330-341, 2011.

12 Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H., "Engineering Design – A Systematic Approach", Springer, 2007.

13 Rees, G., "Seeing is not perceiving", Nature Neuroscience, Vol. 4-7, pp. 678-680, 2001.

14 Beetz, M., Buss, M., Wollherr, D., „Cognitive Technical Systems – What is the Role of AI?", Lecture Notes in Computer Science, Volume 4667, pp. 19-42, 2007.

15 Fellbaum, C., "WordNet: An Electronic Lexical Database". MIT Press, 1998.