INTERNATIONAL DESIGN CONFERENCE - DESIGN 2006
Dubrovnik - Croatia, May 15 - 18, 2006.

DESIGN 2006

ON THE IMPORTANCE OF A FUNCTIONAL DESCRIPTION FOR THE DEVELOPMENT OF COGNITIVE TECHNICAL SYSTEMS

K. Paetzold

Keywords: mechatronics, cognitive abilities, product development

1. Introduction

Today, higher and higher expectations and demands are made on mechatronic products in respect of their functionality. The higher level of intelligence of such multi-disciplinary systems results from the opportunities coming along with further developments within informatics. This opens up new opportunities to develop systems, which independently improve and optimize their own performance. With it however, also a number of new challenges for product development are resulting. Cognitive abilities in technical systems are always tied to their physicalness, which means, their acting in terms of the system's intended purpose is based on data collected by the system from its environment or provided by the system itself.

In the context of this article, cognitive systems in medical technology applications are mainly considered. Here, the intended objective is to reestablish quality of life lost due to disease by means of cognitive systems (e.g. prostheses in cases of dismemberment or compensation of abilities in cases of dementia syndromes).

Cognitive abilities in technical systems result from the close interplay of individual components and thereby also from the close cooperation between domains. In order to support the product development process of cognitive technical systems, it is therefore first of all necessary to characterize these systems more precisely. The objective of considerations regarding structuring the product development process is primarily to make the complexity of such systems manageable.

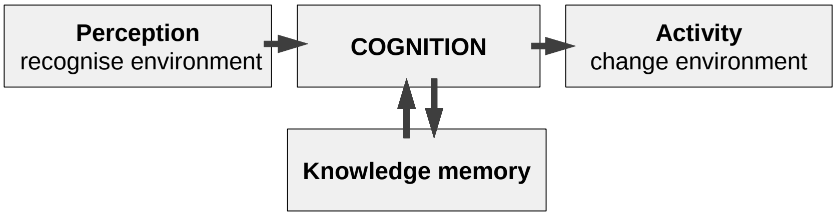
2. Characterization of cognitive technical systems

To enable implementing cognitive abilities in technical systems, it is first of all necessary to describe those abilities more precisely from the viewpoint of cognition science.

2.1 Cognitive abilities from the viewpoint of cognition science

The term cognition derives from the Latin term *cognoscere* and means something like to recognize, to perceive, or to know (Figure 1) [Strube, 96]. Very generally within psychology, first of all the capacities of recognition, perception and knowledge are associated with it. This means, a system is able to perceive its ambience, to recognize relevant data, to represent those in the system, and to associate them with knowledge already available. Then regarding the objectives and tasks, problem-specific actions are derived. Simultaneously, it is essential to evaluate and to memorize these actions and resulting reactions in connection with the triggering data. Experiences by the system are resulting thereof, which can be utilized in terms of behavioral rules. An actoric component is necessary to enable active influence on the environment and/or to operate actively in it. Furthermore, some means of communication between the user of the system, the system itself, and the environment is required.

Developing of such behavioral rules by comparing with former actions and their context is synonymous with learning by the system. Cognitive systems are always adaptive. Therefore, it is essential to have mechanisms for learning available to such systems.



- perceive and cognize
- encode
- memorise and remember
- think and solve problems
- motion control
- use of language

Figure 1. Description of cognitive abilities

Interesting thereby is that cognitive abilities are described by stimulus-reaction mechanisms and therefore, are comparable to processes, which for instance are described by automats. So cognitive processes are computation processes. At the same time, the control loop as reference architecture can also be applied to cognitive systems to describe these stimulus-reaction mechanisms. Thereby, a basis exists for conferring these to technical circumstances.

2.2 Conferring cognitive abilities to technical systems

Cognitive processes may be understood as control processes. The description as control loop thereby reverts to proven cognizances from cybernetics. It already serves as a reference structure for mechatronic systems.

Cognitive technical systems can thus be understood as a consequent further development of mechatronics. The above-described cognitive abilities go beyond the abilities of adaptive systems known so far. For their realization it is necessary to break up rigid sensor-actor chains. Due to the learning aptitude of the system, new behavioral rules can originate, which under certain circumstances connect a defined sensor pattern with an adapted actor operation. At the same time however, reactions must be assured that guarantee the existence of the technical system and its safety and reliability. So cognitive abilities can only be integrated in a technical system that would be capable of acting and survivable also without them.

Resulting thereof is a multi-layer model for cognitive technical systems. (Figure 2), which is redolent of the automation hierarchy for technical processes. On the lowest step, reactive structures are found as non-cognitive regulation. These are needed to maintain the basic functionality of the system. On a higher level stands the associative regulation. In this step, learning mechanisms to generate adapted stimulus-reaction mechanisms (sensor-actor patterns) are already implemented. Conferred to a technical system, such functionality can be ensured by neuronal networks. The basic approach associated with it, to optimize the system via training data, is in essence equivalent to the classical conditioning known from psychology. These networks need training data to optimize the processes. Training mechanisms known from neuronal networks (monitored learning, corroborative learning, unsupervised learning) again can be considered as further subdivisions, whereas unsupervised learning can already be compared to learning by comprehension. A next higher step realizes the actual planning and controlling of actions; here is cognitive learning, i.e. learning by comprehension indispensable.

Keeping the multi-layer model in mind, it be comes clear that the reactive basic structure of the mechatronic system constitutes a fundamental prerequisite for the realization of cognitive technical systems. Thereby it also becomes quickly apparent that the realization of cognitive abilities primarily has to be supported by respectively efficient data processing. The system's physicalness however must also not be neglected as finally, the parameters forming the basis for data processing need to be provided through it. Also the system's learning aptitude is bound to such physicalness as the results of the "thinking process" finally have to be converted into concrete actions by the system's motor function. By rough motor function (e.g. design of kinematics), even if knowing better how it should be, finer motion sequences cannot be realized.

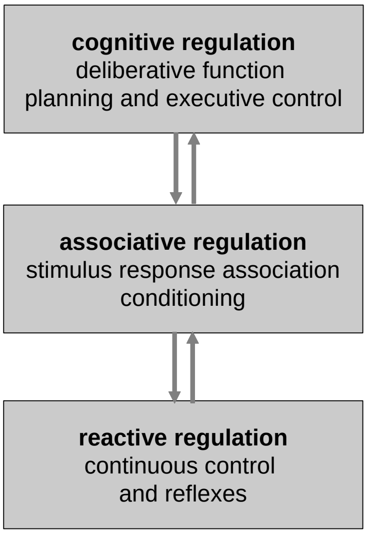


Figure 2. Multi-layer model for cognitive technical systems

Through cognitive abilities, a technical system obtains a high degree of autonomy. In reverse, by the technical system's ability to react flexible in different situations, it is able to act comparatively autonomously. Resulting thereof is a series of product requirements and with it, requirements for the product development process, in addition to the cognitive abilities.

3. Effects on the development of cognitive technical systems

3.1 Regarding product description and structuring

The reference architecture of a control loop, on which the product is based, is by its content of information firstly comparable to the black box model. This means, it firstly represents a rough approach for its description that needs to be improved step by step in the course of development. In this connection it shows that not only hierarchization is of essence during improvement, but also modularization. In this respect, modularization should not relate to the partial components in the control loop. Such separation finally leads to distinct domain-specific development that doesn't treat interfaces to other domains equal for fulfilling function.

Here, modularization results from the fact that individual sub-functions are to be understood in terms of a control loop and are developed as such. Basis for such an approach is the thought that functionally articulated modules normally work relatively autarkic, and so have only few interfaces to other modules.

Eventually, the modules need to be brought into an overall context again, also here being describable and modelable in respect of system characteristics. The connection between modules there is characterized by functional aspects, so that a comprehensive survey brings the system's function and performance come to the fore. Considering that a system's functions primarily act on the assumption

of given input parameters, on the basis of which then defined output values are expected, this may be considered as a customer's or user's view on the technical system. That view is by all means different from a developer's view of a technical system. For the developer it is interesting what happens inside the black box, what conditions are required to ensure functionality. So the developer is interested in the system behavior, which of course, also operates with the given input and output parameters. The objective of both vested interests is the same. Both views can be converted into each other through the automat description. Thereof results for product development a functional description of the system as a binding component, which depending on the perspective, can also be defined as system behavior.
According to cybernetic and system-theoretic approaches, principle considerations on the integration of cognitive abilities into a technical system require different approaches: the classical top-down approach applied in product development and the bottom-up approach in order to enable better understanding of system functions.

3.2 Bottom-up approach

Starting from the three-layer model (Figure 2), it becomes obvious that a cognitive technical system consists of a number of control loops, which however differ considerably from each other in their characterization. While some of the control loops provoke by an action on basis of totalized sensor data that may be respectively flexible, also a number of control loops is required, which always in case of critical sensor data cause appropriate actions to guarantee the system's existence or user's safety. In order to assure the system's flexibility in line with associative regulation, forms of learning aptitude are required, which enable classical as well as operative conditioning [diPrimio, 1998].
Classical conditioning known from Pawlow's experiments with a dog, activate on basis of "learned sensor patterns" (light is on) regulative reactions (causing salivation) or stimulus-reaction mechanisms. Conferred to a technical system, sensor-actor reactions are activated that reflect the basic performance of the system. Operative conditioning is based on a higher level of learning aptitude. No predetermined reactions are used as a reaction to sensor data (see classical conditioning) but a completely new behavior is derived from the recognized sensor patterns. The latent learning process behind it requires respective consideration in system design. Existential sensor-actor reactions, as they finally are provided by the regulative basic system, must not be circumvented or abolished.
According to Strube can cognitive abilities only be implemented in systems that also would exist without such abilities. For this reason, the cognitions from mechatronics on one hand serve as a basis for the development of cognitive technical systems. So first of all, a functioning regulative system with clearly defined actor-sensor interconnection is required for the system structure that ensures the basic functionality. Appropriate modeling and simulation can produce evidence of functionality respectively of system behavior, as it is known in the end from mechatronics. Simultaneously, the basic configuration of the technical system regarding actuator and sensor technology is determined by this regulative system. The principle purpose of the system and the target system associated with it provide the basis for it. (see chapter 3.3).
Cognitive abilities can only be integrated by having a suitable data processing system for instance cumulate sensor data and search such data for patterns or incorporate new sensors and consider the data in the process of generating the actions. From the viewpoint of actuator technology, cognitive abilities initially arise from variability in motion sequences (path-velocity relations) as reactions to different situations. An integration of additional actor elements into the functional regulative system is in principle imaginable, however requires verification with regard to its influence on the purpose of the system. As such additional actors must be recognizable from the definition of the purpose of the system, they are also to be considered in the regulative system.. Otherwise they would make significantly higher demands on the system design, since the technical system's basic structure needs to be prepared for it.
When a process-accompanying simulation is consequentially pursued to enable a derivation of conclusions regarding system behavior, the integration of cognitive technical systems represents a great challenge. The question is how learning aptitude can be modeled, whether symbolic modeling would be sufficient for depicting the environment, or if that environment needs to be represented in its entire complexity in form of differential equations. Additionally, modeling of a memory unit is

necessary that acts as knowledge database for the system. Here it is essential to check for the future, whether that problem statement, the representation of cognitive characteristics, can be integrated respectively depicted in established modeling approaches.

3.3 Top-down approach

The top-down approach describes the usual procedure in the product development process. Starting point for system development is the desired purpose of the system to be defined, at first representing only a qualitative assertion by conferring it into an overall function. A technical system's abilities however are not representable just by the purpose of the system. For a full definition of abilities, additional boundary conditions and constraints are necessary as they usually become describable in the requirement specification. Only in knowledge of both of these aspects (purpose of the system respectively function and requirement specification) can the developer proceed purposefully with the implementation into a technical system.

In cognitive technical systems with their high degree of autonomy and learning aptitude, the overall function now includes a series of sub-functions, which cannot be hierarchically coupled anymore, but stand independently side by side, forming a target system (see below). These sub-functions are characterized by the fact that not only the actual purpose of the system is ensured, but also their own existence. For the definition of the latter aspect, the guide values according to Bossel [Bossel, 94]: safety, effectiveness, freedom of action, mutability, considerateness, and existence, are used.

The functions defined for the purpose of system development again must be suitable to cover or to reach specific targets within the application. The targets arising in the course of using the cognitive technical system are therefore to be considered as a subset of the purpose of the system. Herefrom, some essential aspects become immediately identifiable, which complicate the development of such systems:

- Due to the high degree of autonomy, the cognitive technical system must be prepared for operating in an unstructured or semi-structured environment;
- Existence-endangering conditions must not only be recognized but also mastered by respective appropriate strategies;
- Possible sub-ordinate targets need to be prioritized respectively need be able to be situation-specifically prioritized to ensure that the system really reacts appropriately.

The first point addresses a very substantial circumstance, which cannot be overcome by conventional methods of resolution. The problems in describing the environment result not only from the impossibility to structure that environment, but also from the fact that user and developer have different perspectives on the system. Concealing behind this is last but not least that both work with different logics. While a developer for instance says monitoring is possible (logic assertion: true), the user assumes that monitoring is e.g. ethically not justifiable (logic assertion: false). From this thought follows that not only different definitions of the environment may result, but also that targets respectively the purpose finally are not hierarchically structured anymore but stand side by side to each other in terms of a heterarchy. A consideration of the environment from different perspectives without considering the logics concealed behind it brings about circularities, thus facts or assertions, which finally have to serve for their own explanation. The problem of circularities can be avoided by strict separation and the associated parallelization. [vonGoldammer, 2003]

This means for the development of cognitive systems, that three perceptions on environment need to be taken into account:

- The developer's perception as an observer with its horizon of experience and the logic connected to it
- The user's perception as another observer with its horizon of experience and the logic connected to it (differing from the developer's perception)
- The perception of the cognitive technical system that requires some self-reference to enable the system to distinguish itself from the environment

Thereof resulting are several descriptions of the environment, looked upon from different standpoints. A reconciliation of these environment perceptions in terms of a summary of all recorded parameters considered as relevant finally leads to objectivity in environment description. Moreover, the problem's complexity is not reduced, but however, it becomes well-structured through considering the different perceptions together with their priorities. That already constitutes a simplification. The reduction made and the subsequent consolidation of significant parameters in the second step nevertheless guarantees a sufficient totality of description.

From such a consideration of the environment result two essential aspects for product development:

1. High demands on modeling and simulation:
   A material part of the description of a cognitive technical system's performance is an environment model incorporating the different perceptions on the product. In doing so, it is essential to model the developer and the user as observers on the cognitive technical system and to reconcile their observations for each individual step of the process. Hereof then the actual environment model is to be derived that can be integrated into behavioral modeling. The behavioral model again is to be verified in respect of the functionality to be expected.
2. Integrating the creation of an environment model in the learning process:
   A cognitive technical system always requires a learning phase so to speak to adapt itself to the user and to the specific environmental conditions. During this learning phase, also the coordination between developer, user, and cognitive technical system must be realized, which in the first place is necessary between developer and user due to their different logics. This learning phase in terms of an adaptation to the specific user situation is independent from the learning processes, which is aspired by breaking up the strict actor-sensor chains, so also independent from learning by experiences. Nevertheless is it necessary to consider this adaptation and the different perceptions on the environment connected to it in the product description, as finally self-reference is only characterized by the abilities that result from the physicalness of the technical system.

Open from the viewpoint of modeling remains also here how modeling of the environment on basis of observer models must look. Is here classical parametric modeling required, for which respective describing languages are available, or will the approaches of symbolic modeling suffice?

4. Importance of functional modeling for cognitive technical systems

Integration of cognitive abilities into a technical system implicates a high degree of complexity, for the product itself as well as for the development process associated to it. It became clear that the application of methods of modeling and simulation are indispensable for ensuring the functionality of the product. In doing so, it is by far not enough to reproduce product characteristics by an accumulation of parameters. Only by creating coherence between them one will succeed with describing the characteristics of a cognitive technical system while thereby ensuring the functionality of the product.

A possible approach for the necessary digital product description is provided by a digital mock-up. The data model the digital mock-up is based on however is in no way sufficient for reproducing the cognitive technical system, since her predominantly geometric information is archived with other semantic information playing a subordinate role. Finally, a number of parameters also only conditionally reflect the development status of the product within the development process. For this purpose, a conjunction of data to models is necessary that permits an assertion on the behavior of the system. A dataset on basis of a requirement specification may indeed be complete by a digital mock-up to describe a product. The purpose of the system however, the functionality of the product, only becomes accessible through interactions between the sub-functions defined by parameters.

A complete product description therefore includes on the one hand all parameters, so also the ones describing relevant auxiliary functions (temperature rise, electromagnetic fields, etc.). On the other hand, a digital mock-up appears to be insufficient because assertions on the functions of the product cannot be derived. Only the integration of the properties associated with parameters in function modules while simultaneously interconnecting them through input and output parameters enables a representation of the possible system behavior.

Only by such functional mock-up it will become possible to compare the system behavior with the product functionality the user expects. An assertion regarding the development status of a product will be correspondingly more objective since now, the purpose of the system (in terms of product

behavior) as well as the requirement specification (in form of the actual parameters) are taken into account [Krause, 2005].
A functional mock-up has another advantage: for the integration of domains into the development process, the functional mock-up approach seems to be promising. According to experience, the individual domains have very different perspectives on the development process and the technical system itself. While mechanical engineers act very strongly geometry-oriented, behavior is in the foreground for electronics development. Informatics starts its work on basis of functions that are to be defined. Just from that consideration follows that a geometry consideration excludes the other domains. An extension of the classical approach for product models in the digital mock-up by semantic information only represents an auxiliary framework that does not really support the integration of domains. In case of a functional mock-up, this is to be looked at differently. When geometry quantities of mechanical engineering are considered with regard to their behavior and the functions to be fulfilled, a uniform language is found with electronics development and informatics.
Moreover, there are also other fields of knowledge involved in the development of cognitive technical systems. In order to integrate cognitive properties in technical systems, it is necessary to incorporate cognitions from cognitive sciences and psychology. For applications in the medical field, it is furthermore necessary to take also that field of knowledge into account. As these fields of knowledge work with approaches of the system theory that postulates input-output relations as basis for describing behavior, function-oriented consideration in terms of a functional mock-up may be understood as a general form of description, which is able to integrate further fields of knowledge.

5. Summary

The development of cognitive technical systems may be considered as a consequent further development of mechatronic systems, in which rigid sensor-actor chains are broken up while thereby becoming flexible. Certainly, this is primarily achieved by appropriate data processing, which also enable a high degree of learning aptitude. Cognitive abilities however also have an effect on the product structure, so on the physicalness of the system. Their development again needs to be target-oriented advanced. Thereby it proves to be reasonable to represent the system's functionality by input-output coherences. Behavioral modeling not only enables reconciliation with the functions required by the user, but can simultaneously be understood as an integrating element for the domains involved. From this derived is the postulation to supplement today's prevailing digital mock-up by a functional mock-up.

References

Strube, G.: „Kognition“. In G. Strube et al. (Hrsg.): „Wörterbuch der Kognitionswissenschaft“, Verlag Klett-Cotta, 1996.

diPrimio, F.: “Roboterkognition”. GMD-Report, Nr. 16, ISSN 1435 2702, Bad Augustin, 1998.

Goldammer, E.: „Eine Poly-kontekturale Systemtheorie und deren Konsequenzen“.
ftp://gatekeeper.informatik.fh-dortmund.de/pub/professors/vongoldammer/publications/
Polykontexturale_Systemtheorie.pdf (letzter Abruf Dez. 2005)
Vortrag bei: „Kybernetische Visionen – (Re)vision der Kybernetik, veranstaltet vom Institut für Kybernetik e.V. und der Gesellschaft für Kommunikationskybernetik. Berlin 26.-28.11.99.

Bossel, H. : „Modellbildung und Simulation – Konzepte, Verfahren, und Modelle“. Verlag Vieweg, 1994.

Krause, F.-L.; Rothenburg, U.: Advanced Product Validation using FunctionalDMU. Presentation, Deutsch-israelisches Symposium, Berlin 2005.

Knoll, A.; Christaller, T.: Selbstrepräsentation, Selbstwahrnehmung und Verhaltens-steuerung von Robotern. In: Selbstrepräsentation in Natur und Kultur / Sandkühler, Hans Jörg[Hrsg.] (Philosophie und Geschichte der Wissenschaften : Studien und Quellen), S. 109 – 132, 2000.

Frank, U.; Giese, H. et al: Selbstoptimierende Systeme des Maschinenbaus-Definition und Konzepte. Heins-Nixdorf-Institut Paderborn, 2004.

Dr.-Ing Kristin Paetzold
Chief Engineer
Institute for Engineering Design, University of Erlangen-Nürnberg
Martensstr. 9; 91058 Erlangen, Germany
Tel.: +49 (9131) 85 23 222
Fax.: +49 (9131) 85 23 223
Email: paetzold@mfk.uni-erlangen.de
URL: www.mfk.uni-erlangen.de