We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

A Software Architecture for Cognitive Technical Systems Suitable for an Assembly Task in a Production Environment

Eckart Hauck, Arno Gramatke and Klaus Henning
ZLW/IMA, RWTH Aachen University
Germany

1. Introduction

In the last years production in low-wage countries became popular with many companies by reason of low production costs. To slow down the development of shifting production to low-wage countries, new concepts for the production in high-wage countries have to be developed.
Currently the highly automated industry is very efficient in the production of a small range of products with a large batch size. The big automotive manufacturer like GM, Daimler and BMW are examples for this. Changes in the production lines lead to high monetary investments in equipment and staff. To reach the projected throughput the production is planned completely in advance. On the other side a branch of the industry is specialized in manufacturing customized products with small batch sizes. Examples are super sports car manufacturers. Due to the customization the product is modified on a constant basis. This range of different manufacturing approaches can be aggregated in two dilemmas in which companies in high wage countries have to allocate themselves.
The first dilemma is between value orientation and planning orientation. Value orientation implies that the manufacturing process involves almost no planning before the production phase of a product. The manufacturing process is adapted and optimized during the production. The antipode to value orientation is planning orientation. Here the whole process is planned and optimized prior to manufacturing.
The second dilemma is between scale and scope. Scale means a typical mass production where scaling effects make production more efficient. The opposite is scope where small batch sizes dominate manufacturing.
The mainstream automotive industry represents scale focused production with a highly planning oriented approach due to the grade of automation involved. The super sports car manufacturer is following a scope approach which is more value oriented.
These two dilemmas span the so called polylemma of production technology (Fig. 1) (Brecher et al, 2007). The reduction of these dilemmas is the main aim of the cluster of excellence "Integrative Production Technology for High-Wage Countries" of the RWTH Aachen University. The research vision is the production of a great variety of products in

small batch sizes with costs competitive to mass production under the full exploitation of the respective benefits of value orientation and planning orientation.
To reach the vision four core research areas were identified. These areas are "Individualized Production Systems", "Virtual Production Systems", "Hybrid Production Systems" and "Self-optimizing Production Systems". Self-optimizing production systems try to realize value orientated approaches with an increase in the planning efficiency by reusing gained knowledge on new production conditions.
The research hypothesis is that only technical systems which incorporate cognitive capabilities are capable of showing self-optimizing behavior (Heide 2006). In addition to that these Cognitive Technical Systems can reduce the planning efforts required to adapt to changes in the process chain (Brecher et al. 2007). In this chapter a software architecture for such a Cognitive Technical System will be described and a use case in the context of assembly processes will be presented. Section 2 will deal with the definition of the terms "Self optimization", "Cognition" and "Cognitive Technical System". Section 3 deals with related work in the context of Cognitive Technical Systems and the involved software architectures. The fourth section describes an excerpt of the functional as well as the non-functional requirements for a Cognitive Technical System. Afterwards the software architecture will be presented. The sixth section will introduce an assembly use case and the chapter closes with a final conclusion in section 7.

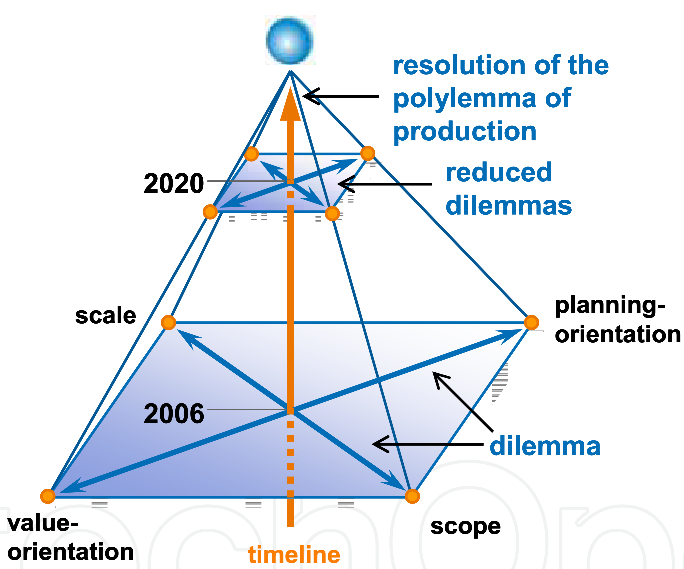

Fig. 1. Polylemma of Production Technology

2. Definition of Terms

2.1 Self-Optimization

Self-optimization in the context of artificial systems includes three joint actions. At first the current situation has to be analyzed and in a second step the objectives have to be determined. These objectives can be contradictive. In this case a tradeoff between the objectives has to be done by the system. The third step is the adaption of the system behavior. A system can be accounted for a self-optimizing system if it is capable to analyze and detect relevant modifications of the environment or the system itself, to endogenously modify its objectives in response to changing influence on the technical system from its surroundings, the user, or the system itself, and to autonomously adapt its behavior by means of parameter changes or structure changes to achieve its objectives (Gausemeier 2008). To adapt itself, the system has to incorporate cognitive abilities to be able to analyze the current situation and adjust system behavior accordingly.

2.2 Cognition

Currently, the term “Cognition” is most often thought of in a human centered way, and is not well defined in psychology, philosophy and cognitive science. In psychology Matlin (2005) defines cognition in the context of humans as the “acquisition, storage, usage and transformation of knowledge”. This involves many different processes. Zimbardo (2005) accounts, among others, “perception, reasoning, remembering, thinking decision-making and learning” as essential processes involved in cognition. These definitions cannot be easily transferred to artificial systems. A first approach to the definition of “Cognition” in the context of artificial systems is given in Strasser (2004). She based her definition on Strube (1998):

“Cognition as a relatively recent development of evolution, provides adaptive (and hence, indirect) coupling between the sensory and motor sides of an organism. This adaptive, indirect coupling provides for the ability to learn (as demonstrated by conditioning, a procedure that in its simplest variant works even in flatworms), and in higher organisms, the ability to deliberate”.

Strasser develops from this the definition that a system, either technical or biological, has to incorporate this flexible connectivity between input and output which implies the ability to learn. Thereby the term learning is to be understood as the rudimentary ability to adapt the output to the input to optimize the expected utility. For her, these are the essential requirements a system has to incorporate to account for being cognitive.

This definition is to be considered when the term “Cognition” is used. It should be stated that this definition is the lower bound for the usage of the term. Therefore this does not imply that a system which can be accounted as “Cognitive” is able the incorporate human level cognitive processes.

2.3 Cognitive Technical Systems

With the given definition for “Self-optimization” and “Cognition” the term “Cognitive Technical System” can be defined. Also a description of the intermediate steps involved in development towards a Cognitive Technical System able of incorporating cognitive processes of a higher level can be given.

The definition used in the context of this chapter is that an artificial system, which incorporates cognitive abilities and is able to adapt itself to different environmental changes, can be accounted as Cognitive Technical System.

Fig. 2 shows the different steps towards a Cognitive Technical System capable of cognition on a higher level. As cognitive processes of a higher level the communication in natural language and adaption to the mental model of the operator can be named. Also more sophisticated planning abilities in unstructured, partly observable and nondeterministic environments can be accounted as cognitive processes on a higher level.

The current situation of human machine interaction in the context of a production environment is as follows (left part of the picture): Cognitive processes occur only in humans. The technical system, which consists of an Interaction System and a Technological Application System, is not cognitive in the sense of Strasser (2004). The interaction happens in a classical way via a human-machine-interface embedded in the Interaction System. The output of the Technological Application System is evaluated and optimized by the human operator only. This also means that only the human operator can reflect about the output of the Technological Application System and improve it by adapting the parameters of the processes via the human-machine-interface.

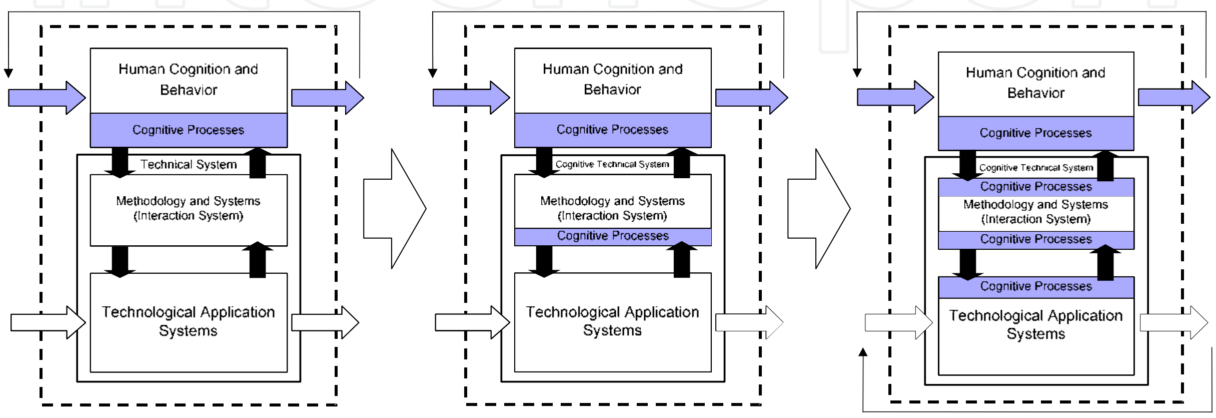

Fig. 2. Steps in the development of Cognitive Technical Systems

The intermediate step (middle of the picture) is the incorporation of basic cognitive processes in the Interaction System. The technical system could now be accounted for a Cognitive Technical System. The cognitive processes can involve reasoning and decision making. The Cognitive Technical System has to incorporate a knowledge base upon which decisions can be derived. These cognitive processes are embedded in the Interaction System which communicates with the Technological Application System but also controls it.
The right part of the picture shows a visionary Cognitive Technical System. Such a system incorporates cognitive processes on all levels, which means that the human-machine-interaction is based on multimodal communication. In addition to that the human-machine-interface adapts itself to the mental model of the human operator during the communication process. This increases the efficiency of the communication process dramatically. Therefore the human-machine-interface incorporates cognitive abilities. In addition the Interaction System incorporates cognitive processes in the communication with the Technological Application System which also embeds cognitive capabilities. This means that the communication can be alleviated to a higher level. The systems only exchange concepts of a certain kind and the subsequent tasks are derived by the system itself. A human related communication which corresponds to such an exchange of concepts would be the task of writing a report. This involves many steps to be enacted by the receiver of the task which are not communicated. Nonetheless is the result in accordance to the "intentions" of the human who gave the task.
In addition such a system would be able to evaluate the output of the process and with that the parameters which lead to it. This enables self-optimizing behavior. The evaluation process is depictured as a feedback system from the Cognitive Technical System (lower right

part). In relation to manufacturing processes, a Cognitive Technical System can control entities of the Technological Application System like robots, belt conveyors, etc. to conduct different tasks. Also a multitude of Cognitive Technical Systems can cooperate to control a process chain and optimize it as a whole in respect to the optimization objectives given to the systems.

Task descriptions can be given to the system in a more abstract way. A possible description could be the shape of the product or certain product properties. With this description the Cognitive Technical System derives the needed steps to produce the desired product. A description on such an abstract level is hugely underspecified which corresponds to a task description given from one human to another (Hägele 2008). To derive the missing information needed to solve the task a massive knowledge base is mandatory.

3. Related Work

Due to the vast research efforts in different fields like artificial intelligence, software engineering, electrical engineering, etc. this section does not intent to give a complete overview, but present a selection of related work with the focus on software architectures. As possible application fields for Cognitive Technical Systems the autonomous vehicle control, manufacturing environments as well as service robotics can be identified. There are many more fields which are not evaluated further in this context.

In the field of autonomous vehicle control the DARPA grand challenges in 2005 and 2007 showed that the control of ground vehicles in a semi-unstructured environment with the constraints of following the rules of the road is possible (Montemerlo et al. 2006). The software architectures used in these Cognitive Technical Systems followed a multi layer approach with the extensive use of state machines (Urmson et al. 2007).

In autonomous robots many architectural approaches are proposed (Karim 2006, Konolige 1998, Gat 1998 et al.). These software architectures focus on the combination of a deliberative part for the actual planning process with a reactive part for motion control (Putzer 2004).

In production technology, the cluster of excellence "Cognition for Technical Systems" (CoTeSys) is researching software architectures for Cognitive Technical Systems in production environments (Ding et al. 2008). The research focuses on the implementation of cognitive abilities in safety controllers for plant control. In this context the human machine cooperation is the main evaluation scenario.

All described approaches do not focus on the application of Cognitive Technical Systems in an assembly operation.

4. Requirements

4.1 Functional Requirements

This section describes the functional requirements for a Cognitive Technical System suitable to act in a production environment. A functional requirement is a requirement which can be noticed during the operation of the system (Sommerville 2007).

The functional requirements that a Cognitive Technical System must fulfill are the capability to process different sensor inputs (visual, tactile or electric sensors) and aggregate them to extract essential information. Based on this information, the Cognitive Technical System must process the information and find the next best action concerning the current

environmental state and the given objective. To change the environment according to the next action derived by the Cognitive Technical System, external entities like robot actuators or conveyor belts have to be controlled.
The Cognitive Technical System must interact with a human operator via a human-machine-interface. The actual design of the human-machine-interface is not part of the functional requirements (Cockburn 2003) but is specified in the non-functional requirements. To derive a decision out of the received information, the system must have a knowledge base which contains the domain knowledge. Also the procedural knowledge about the different operations it has at his disposal, for changing its environment must be stored.
The environment of a production facility adds a further functional requirement for a Cognitive Technical System. The communication via different protocols with machinery like programmable logic controllers (PLC) and multi-axis robots has to be ensured.

4.2 Non-Functional Requirements

Non-Functional requirements are defined as requirements which specify criteria that can be used to judge the operation of a system, rather than specific behaviors (Sommerville 2007). The non functional requirements for the human-machine-interface derive from DIN ISO 9355-1 and can be separated in 14 categories, which will not be described here in detail.
The requirements for the software architecture are partly derived from ISO 9126. The following categories are considered the essential ones and will be described in more detail:

- Modularity, Extendibility, Flexibility
- Robustness and Reliability
- Response times
- Information- and Datamanagement
- External communication
- User Interaction

Modularity, Extendibility, Flexibility
The software architecture of a Cognitive Technical System suitable for an assembly task in a production environment has to meet the requirements of modularity, extendibility and flexibility. Modularity in this context means, that components can be interchanged without redesigning the whole system. This concerns the user interface, the different controller components and the decision making components. This demands the encapsulation of single functionalities within components and the usage of well defined interfaces between them. The software architecture must be extendable in the sense that new components can be integrated without much effort. This satisfies also the requirement of flexibility.
Robustness and Reliability
In a production environment the requirements for the reliability and the robustness of a system are high. The technical system must have a high reliability because of the high costs of a possible production stop in case of a system failure. Because of this certain safety measures must be implemented in the Cognitive Technical System. This can be realized through redundancy of components or by fault tolerant code. This also ensures a high robustness.
Response times
In a production environment processes are optimized for high throughput. This puts further constraints on the software architecture of such a system. The response time must be low enough to react to sudden changes in the environment. The deliberative part of the

Cognitive Technical System can not derive decisions in real time due to the amount of knowledge processed. Therefore the overall response time of the system has to be ensured by a mechanism which does not depend on deliberative decision making.

Information- and Datamanagement

The information flow in the Cognitive Technical System is quite extensive. The sensory information has to be processed and routed to the concerning components. The software architecture has to incorporate an internal communication to feed the information to the components. In addition, storage of the data in different repositories has to be ensured due to the high bandwidth and the amount of accumulated data.

External communication

The Cognitive Technical System has to communicate with the different entities in a production environment. These can be physical entities like robots and programmable logic controller, but also different bus protocols (CAN-Bus and Process Field Bus (PROFIBUS)) have to be supported by the respective interfaces. Also a simple extendibility of these interfaces must be possible.

User Interaction

The Cognitive Technical System has to ensure the communication with the user of the system. The user input has to be processed and the decisions of the Cognitive Technical System have to be presented to the user.

4.3 Conclusion

The functional and non-functional requirements for the system influence the design of the software architecture. Especially the requirements of a production environment by demanding a low response time of the system define the software architecture. Furthermore the reliability is an important requirement.

5. Software Architecture

5.1 Multilayer approach

To meet the functional and non-functional requirements a software architecture for a Cognitive Technical System suitable for assembly tasks has to incorporate multiple components.

The system has to work with different levels of abstractions. This means that the deliberative mechanism cannot work on the direct sensor data received from the Technological Application System. Therefore an abstraction of the received data is necessary. This demands a component which can aggregate the received information for the deliberative mechanism. To meet the requirement of a low response time a control mechanism has to be incorporated which can act without waiting for the deliberative mechanism to respond. Also, the Cognitive Technical System has to be able to control the production facilities as well as ensure a human machine communication. Especially the concepts of modularity and reliability were the driving factors for the chosen approach. To meet these requirements a multilayer approach for the software architecture of the system was chosen (Gat 1998).

Fig. 3 shows the software architecture embedded in the human-machine-interaction. The Cognitive Technical System incorporates the Technological Application System as well as the Interaction System. The software architecture separates the Interaction System into four layers which incorporate the different mechanisms required. The Presentation Layer incorporates the human machine interface and an interface for the modification of the knowledge base. The Planning Layer is the deliberative layer in which the actual decision for the next action is made. The Coordination Layer provides services to the Planning Layer which can be invoked by the latter to start action execution. The Reactive Layer is responsible for a low response time of the whole system in case of an emergency situation. The Knowledge Module contains the necessary domain knowledge of the system.

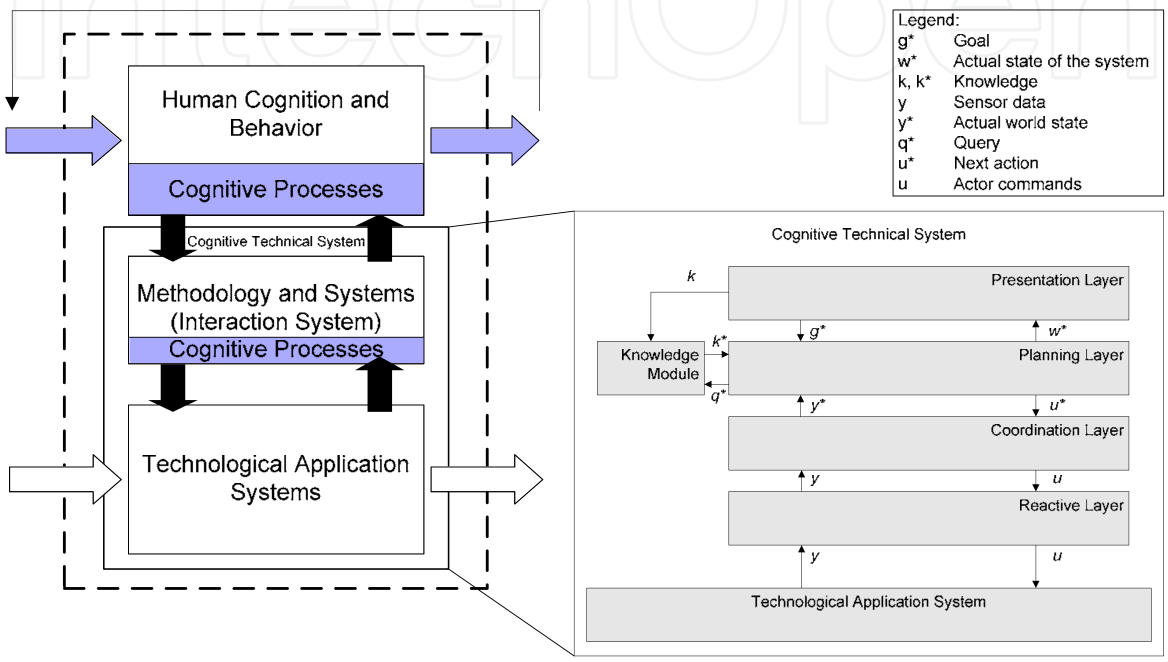

Fig. 3. Software architecture embedded in the human machine interaction

At the beginning the human operator gives the desired goal to the Cognitive Technical System via the Presentation Layer. This goal g^* is then transferred to the Planning Layer where the next action u^* is derived based on the actual world state y^* and the desired goal g^*. The actual world state is based on the measured variables y from the sensors in the Technological Application System which are transferred via the Reactive Layer. In the Coordination Layer y is then aggregated to y^*. To derive y^*, the sensor data y at a discrete time $t \in R^{\geq 0}$ is taken into account. $y(t) \in R^{n_y}$ denotes the current vector of the current measured variables at time t. This vector is then transformed in the world state $y^*(t)$. This means that the base on which all decisions in the Planning Layer are made is the actual world state y^* at a certain time t. Therefore the decision process must not take too long, because the state of the Technological Application System can have changed significantly in the meantime.

The next best action u^* derived in the Planning Layer is sent back to the Coordination Layer, where the abstract description of the next best action u^* is translated into a sequence of actor commands u, which are sent via the Reactive Layer to the Technological Application System. There, the sequence of commands is executed and the changed environmental state is measured again by the sensors. If the new measured variables y of the Technological Application System indicate an emergency situation the Reactive Layer ensures a low

response time. Then the sensor data is processed directly in the Reactive Layer and the according actor commands are executed.
Fig. 4 shows the software architecture in more detail. The different layers and their components will be described in more detail in the following section.

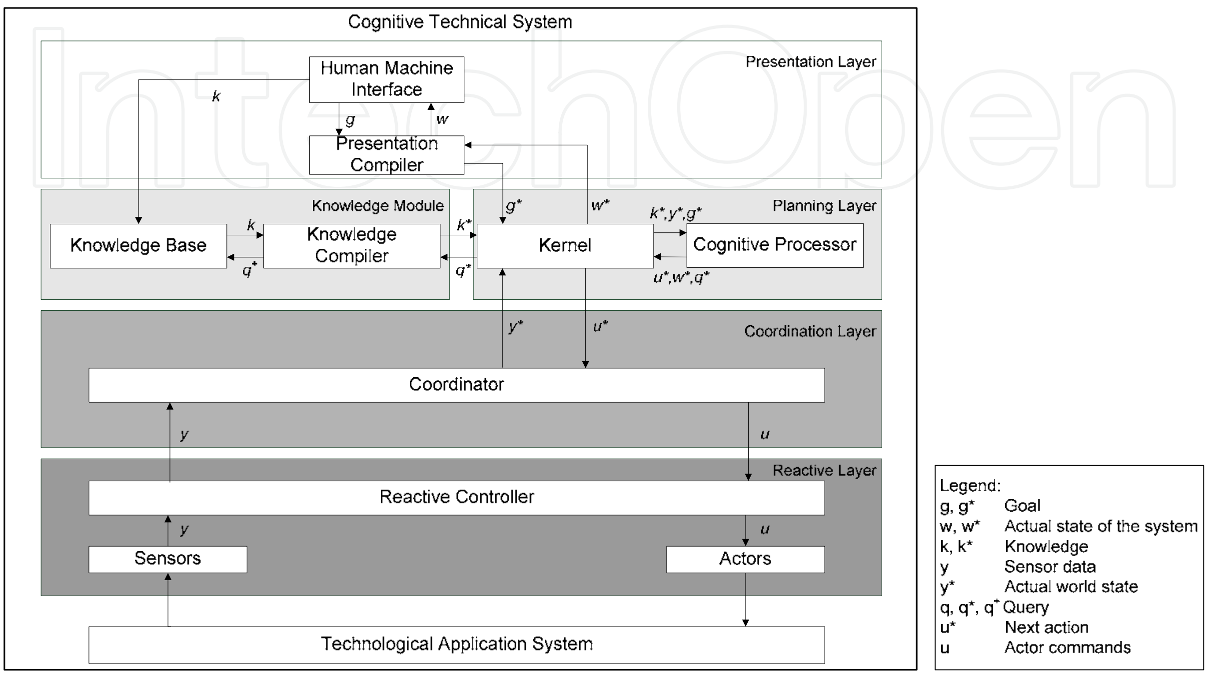

Fig. 4. Software Architecture of the Cognitive Technical System with components based on a multilayer approach

5.2 Presentation Layer

The Presentation Layer is responsible for the interaction with the user. It incorporates the human-machine-interface which is designed for the special requirements given by interacting with a technical system with cognitive capabilities.
The domain knowledge $\boldsymbol{k}$ is encoded in a fixed representational formalism. One possibility is the structuring of $\boldsymbol{k}$ in an ontology. The knowledge engineer encodes the domain knowledge specifically to the task the system has to enact. This is done prior to the system start. During the operation of the system a human operator is interacting with the system. This operator specifies a task description $\boldsymbol{g}$, which is transferred to the component Presentation Compiler. In case of an assembly task the description $\boldsymbol{g}$ can be the description of the shape of parts to be assembled and the description of the location and forms of the parts in the final assembly. This can be done, but is not restricted to, using a graphical representation, e. g. a CAD program. The Presentation Compiler has to translate this task description $\boldsymbol{g}$ into a goal state $\boldsymbol{g^*}$ which can be interpreted by the Cognitive Processor of the Planning Layer.
Due to the changing environment the behavior of a cognitive system is not perfectly predictable in advance. Therefore, the actual state of the system should always be transparent to the operator. The actual state $\boldsymbol{w^*}$ of the system is given to the Presentation Compiler, where $\boldsymbol{w^*}$ is aggregated to a human interpretable machine feedback $\boldsymbol{w}$ which is then transferred to the operator via the Human Machine Interface.

5.3 Planning Layer

The Planning Layer contains the core elements that are responsible for decision-finding. It contains the Kernel and the Cognitive Processor as components. The Kernel distributes the signal flows in the Planning Layer. The Cognitive Processor computes the next best action u^* based on the goal state g^* and the current world state y^*. If the Cognitive Processor cannot derive a next best action it can send a query q^* for more information to the Knowledge Module.

The Kernel component then invokes the action execution according to the action returned by the Cognitive Processor. In case of a request for more information, the Kernel queries the Knowledge Base for actions applicable on the objects in y^*. According to the actual processor used, the Knowledge Base returns the knowledge k^* via the Knowledge Compiler. The additional knowledge is then considered in the computation of the next best action. Fig. 5 shows the activity diagram for the Cognitive Processor. In the rare case that the Cognitive Processor could not find an action and the Knowledge Base could not return k^*, the Cognitive Processor queries the human operator for the next action. The user can then either give the next action or change the environmental state. This means that the user changes the environment physically without telling the system explicitly about this. The system then recognizes the new environmental state via the measured variables y, reasons about the new world state y^* and derives the next best action u^* based on y^*.

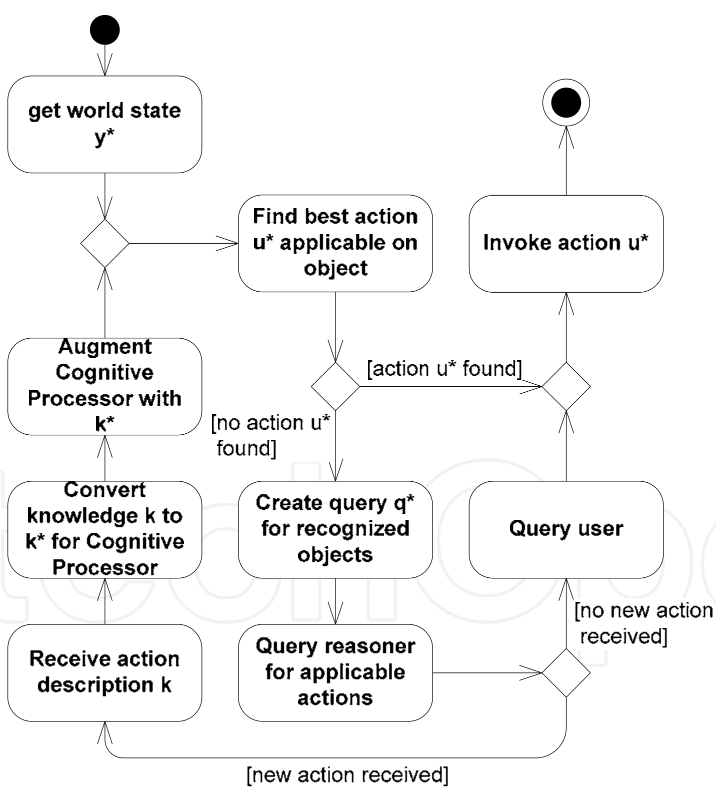

Fig. 5. Activity diagram of possible actions of the Cognitive Processor

Several architectures have been developed for the understanding of the human control behavior. The EPIC (Executive-Process Interactive Control) architecture combines cognitive and perceptual operations with procedural task analysis (Keiras 2004). The different

interconnected modules, called processors, operate in parallel. The Soar architecture is a cognitive architecture based on the "unified theory of cognition" (Newell 1994), which aims to model general intelligence (Laird 1996). It models behavior as selection and application of operators to a state. A state represents the current situation of knowledge and problem-solving, and operators transfer knowledge from one state to another. At runtime, Soar tries to apply a series of operators in order to reach a goal (Laird 1996). Control in Soar refers to conflict solution and is implemented as a deliberate and knowledge-based process. ACT-R control is regarded as an automatic process by using an automatic conflict resolution strategy (Johnson 1998). Of these architectures the Soar architecture was chosen as the Cognitive Processor (Hauck 2008).
Soar is a rule based production system. Rules are fired if they match elements of the inner representation of the current y^* and modify this representation. Via input- and output-links Soar is capable of communication with its environment, e.g. to retrieve a new world state or invoke actions. In addition, a combination of the Soar architecture with a classical planning algorithm like Fast Forward (Hoffmann 2001) is currently investigated. This provides the ability to exploit the capabilities of Soar but also enables the generation of a quick plan to solve a task.

5.4 Coordination Layer

The Coordination Layer is the executable layer of the Cognitive Technical System. It provides executable services to the Planning Layer. These services correspond to the actions the Cognitive Processor can invoke. The Coordinator in the Coordination Layer also processes the measured variables y received from the Reactive Controller via the Reactive Layer and aggregates this information to the current world state y^*.
Also, the Coordinator component receives the next action u^* to be executed. The abstract service invoked by u^* is a sequence of actor commands u. A simple example is the stapling process of two blocks. Provided the positions of the two are known, the service `move(blockA,blockB)` then invokes the sequence of moving the actor, e. g. a robot, to the position of blockA, grasping it and transferring it to the position of blockB and releasing it. u is stored in the Coordinator component and will be executed with parameters given by u^*. u is then executed in the Technological Application System via the Reactive Layer. That way, the Planning Layer is exculpated from the details of the robot movements, e. g. the exact coordinates of the block-locations, etc., which leads, due to a reduced problem space, to faster decisions.

5.5 Reactive Layer

The Reactive Layer and in it the component Reactive Controller is responsible for the low level control of the system. The vector of the measured variables y is observed for values which indicate a possible emergency situation. The Reactive Controller responds then with the according actor commands u.
This ensures low response times in case of an emergency. The Reactive Controller cannot ensure a safe behavior for the system as a whole. This means if a wrong actor command sequence is sent to the actors in the Technological Application System the Reactive Controller does not check this sequence for potential consequences for the Technological

Application System according to the current state. This has to be done by the Cognitive Processor.

5.6 Knowledge Module

The Knowledge Module contains the Knowledge Base which contains the necessary domain knowledge for the Cognitive Technical System to perform the desired task. The domain knowledge k in the Knowledge Base has to be translated in a form which is interpretable by the Cognitive Processor. This is done by a Knowledge Compiler, which consists of two components: The Reasoner and the Mediator. The Reasoner queries the Knowledge Base and receives additional knowledge k. This knowledge is then translated into an intermediate format k' and transferred to the Mediator. The Mediator then compiles the knowledge k' into the syntax k^* which is then processed by the Cognitive Processor. Fig. 6 shows the signal flows and the involved components. In case of an additional information request q^* by the Cognitive Processor the Mediator first translates q^* in q' and the Reasoner accesses the Knowledge Base to infer the requested information.
For assembly tasks, the domain knowledge has to contain the involved actors controlled by the Cognitive Technical System. The formalism used for the domain knowledge is the Web Ontology Language (OWL) (Smith 2004). To store the procedural knowledge, which is used by the cognitive processor in form of production rules the original form is not sufficient. Therefore, an extension to the OWL, the Semantic Web Rule Language (SWRL) (Horrocks et al. 2004) in combination with a description formalism for the direct representation of procedural knowledge in the ontology is used.

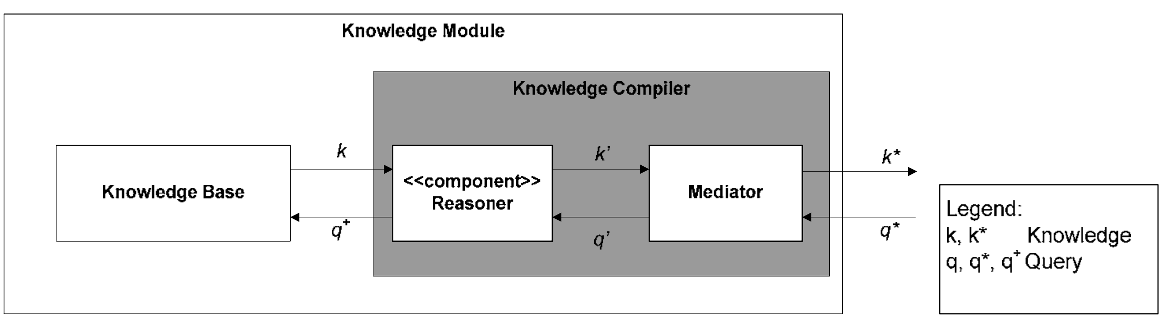

Fig. 6. Component diagram of the Knowledge Module

5.7 Conclusion

The multilayer approach ensures the encapsulation of different information abstractions in different layers. The components in the Planning Layer operate with the highest abstraction of information. The Cognitive Processor invokes the corresponding service according to the next best action. The different services manipulate the environment without dealing with the low level constraints given by the used actors. The Coordination Layer contains the service description in form of sequences of actor commands, which the Reactive Layer than executes and controls.
Due to this approach, the system can deal with a continuously changing environment and adapt itself to it. The system is hybrid in a double fold sense of the word. It connects a continuously stream of input signals with their discrete representation in states and includes reactive and deliberative components.

6. Example: Control of an Assembly Cell

The schematic layout of a robot cell, which is controlled by the Cognitive Technical System is shown in Fig. 7. It consists of two robots and a transport system, which transfers the parts via a conveyor belt. The first robot grasps the incoming parts and puts it on the conveyor. The parts colors and contours are identified by an object recognition software via a CCD camera. If the part is needed for the assembly at hand, the second robot grasps the part and transfers it either to the assembly area in case the part is needed immediately, or to the buffer area. In case that an object is not needed the conveyor transports the object to the leaving part container and it is being discharged. The second robot is equipped with a three finger robot hand to conduct complex gripping operations.
The first evaluations of the Cognitive Technical System will only involve parts with a simple contour, like blocks, spheres etc. This is necessary due to the fact that the object recognition as well as the color recognition would take much longer for complex objects. The system has to adapt to different states without the possibility to preplan the whole assembly process. Therefore the feeding of the parts is stochastic. In addition the actual world state will be repeatedly checked to evaluate if the internal representation in the Cognitive Technical System corresponds to the environmental state.

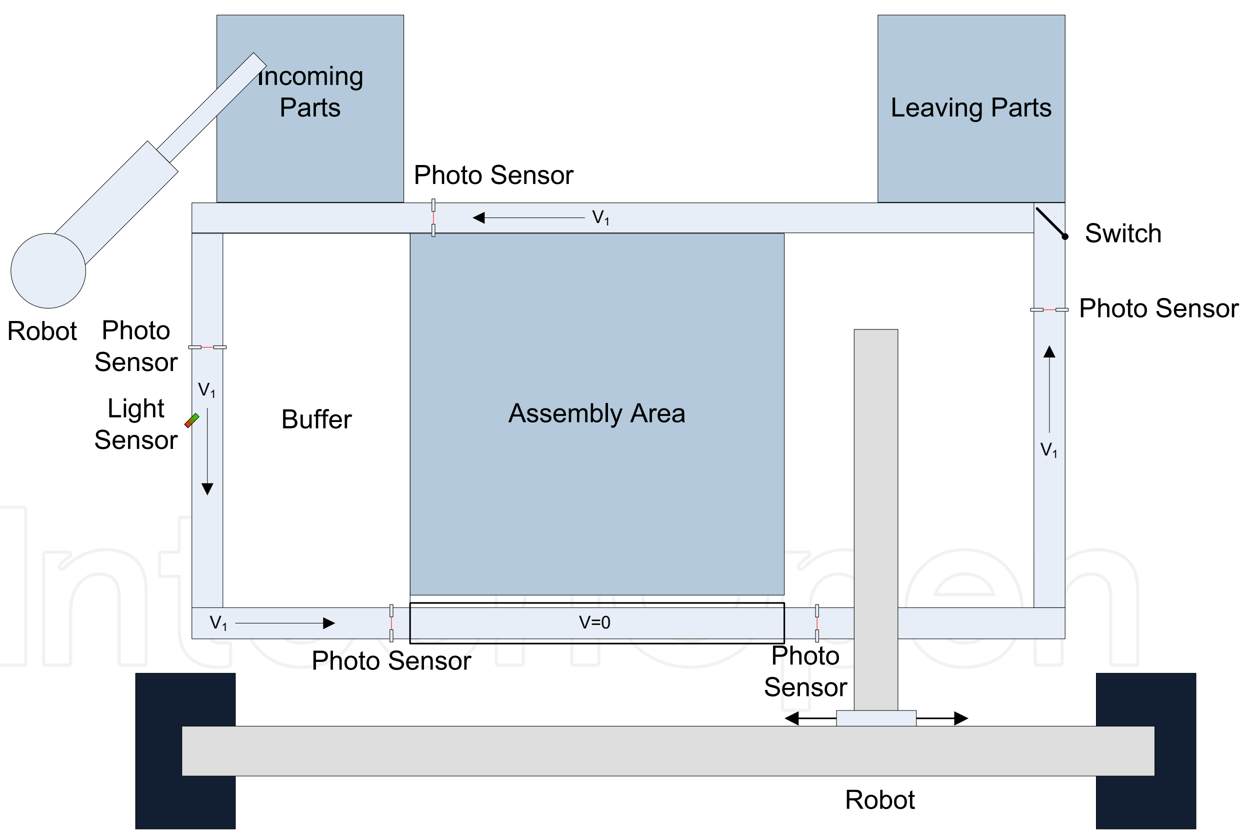

Fig. 7. Schematic of the assembly cell used for the application of the Cognitive Technical System

Possible reasons for unexpected changes in the environmental state can be:

- Erroneous identification of a part
- Dropping or misplacement of a part by the robot
- Changes in the current assembly

Erroneous identification of a part can lead to a false building order for the whole assembly and affect the outcome of an assembly operation significantly. A drop of a part can happen if the three finger robot hand grasps an object wrong or the object falls during the transfer operation. The last possible change in an environmental state is the change of the assembly. This is a scenario where the machine works in cooperation with a human. The change will then be noticed by the system via the measured variables $\boldsymbol{y}$. This is not focus of the current research, but has to be considered for future applications.
Therefore, the Cognitive Technical System has to check the actual world state periodically to prevent the consequences arising out of these changes in the environmental state. To evaluate the system, a simple assembly task will be conducted by the system. The most simplistic geometry is a tower of blocks but this will be extended to the realize of more complex geometries.

7. Conclusion and Future Work

The multilayer approach for a Cognitive Technical System suitable of conducting assembly tasks in a production environment is a feasible one. The software architecture meets the different functional as well as non-functional requirements a production environment has towards such a system. The current work focuses on the implementation of the software architecture and simulation of the environmental states. Future work will include the connection to the assembly cell and the application of the system to more complex object contours.
For interested readers the following links are recommended:
http://www.zlw-ima.rwth-aachen.de/forschung/projekte/exzellenzcluster/index.html
http://www.production-research.de

8. Acknowledgements

The authors would like to thank the German Research Foundation DFG for the support of the depicted research within the Cluster of Excellence "Integrative Production Technology for High-Wage Countries".

9. References

Brecher, C. et al. (2007). *Excellence in Production,* Apprimus Verlag, ISBN: 3940565008, Aachen

Cockburn, A. (2003). Writing effective use cases, Addison Wesley, ISBN: 9780201702255, London

Ding, H. et al. (2008). A Control Architecture for Safe Cognitive Systems, *10. Fachtagung Entwurf komplexer Automatisierungsysteme,* Magdeburg, April 2008, Gat, E. (1998). On Three-Layer Architectures in *Artificial Intelligence and Mobile Robots*, Kortenkamp D., Bonnasso R., Murphy R., (Ed.), pp. 195-211, AAAI Press, ISBN: 9780262611374, Cambridge

Gausemeier, J. (2008). *Towards a Design Methodology for Self-optimizing Systems*, Springer Verlag, ISBN: 978-1-84628-004-7, London

Hauck, E.; Gramatke, A. & Henning,K. (2008). Cognitive technical systems in a production environment, *Proceeding of the 5th international Conference on Informatics in Control, Automation and Robotics*, pp. 108-113, ISBN: 9789898111326, Madeira, May 2008

Hägele, M. (2008). Industrial robotics, In: *Handbook of Robotics*, Siciliano, B., Khatib, O., (Eds.), pp. 963-986, Springer Verlag, ISBN: 9783540239574, London

Heide, A. & Henning, K.(2006). The cognitive car - A roadmap for research issues in the automotive sector, *Proceedings of the 9th IFAC Symposium on Automated Systems Based on Human Skill And Knowledge*, ISBN: 9783902661050, Nancy, May 2006,

Hoffmann J. & Nebel B. (2001). *The FF Planning System: Fast Plan Generation Through Heuristic Search*. Journal of Artificial Intelligence Research, Vol. 14, (2001), pp.253-302, ISSN:11076 – 9757

Horrocks, I. et al. (2004). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/

Johnson, T.R. (1998). A comparison of ACT-R and SOAR. In: Schmid, U., Krems& J.,Wysotzki, F. (Eds.) *Mind modeling*, pp. 17–38, Papst, ISBN: 3933151252, Lengerich

Karim, S. et al. (2006). A Hybrid Architecture Combining Reactive, Plan Execution and Reactive Learning, *Proceedings of the 9th Biennial Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, China, August 2006

Konolige, K. & Myers, K. (1998). The saphira architecture for autonomous mobile robots. In: Kortenkamp et al. (Eds.) *Artificial intelligence and mobile robots: case studies of successful robot systems*, pp.211-242, MIT Press, ISBN: 0262611376, Cambridge

Kieras, D. & Meyer, D. (2004). *EPIC Architecture – Principle of Operation*, Univ. of Michigan, Ann Arbor

Laird, J.E.; Lehman, J.F. & Rosenbloom P. (1996). A gentle introduction to Soar, an architecture for human cognition, In: *Invitation to Cognitive Science*, MIT Press, Boston

Matlin, M. W. (2005). *Cognition*, Wiley& Sons, ISBN: 0471427780, New York

Montemerlo, M. et al. (2006). Winning the DARPA Grand Challenge with an AI robot, *Proceedings of the AAAI National Conference on Artificial Intelligence*, pp. 982-986, ISBN 9781577352815, Boston, July 2006, AAAI, Boston

Newell, A. (1994). *Unified theories of cognition*, Harvard University Press, ISBN: 9780674921016, Cambridge

Putzer, H. (2004). *Ein uniformer Architekturansatz für kognitive Systeme und seine Umsetzung in ein operatives Framework*, Dissertation, München

Smith, M. et al. (2004). *OWL Web Ontology Language Guide*, http://www.w3.org/TR/2004-/REC-owl-guide-20040210/*OWL Web Ontology Language Guide*, http://www.w3.org/TR/2004/REC-owl-guide-20040210

Sommerville, I. (2007). *Software Engineering*, Addison Wesley, ISBN: 9780201398151, London

Strasser, A. (2004). *Kognition künstlicher Systeme*, Ontos Verlag, ISBN:393720296X, Frankfurt

Strube, G. (1998). Modelling motivation and action control in cognitive systems. In *Mind modeling: a cognitive science approach to reasoning, learning and discovery*, Schmid, U., Krems, J., & Wysocki, F. (Eds.), pp 89-108, Pabst, ISBN 159326044X, Berlin

Urmson, C. et al. (2007). *Tartan Racing: A Multi-Modal Approach to the DARPA Urban Challenge*, Pittsburgh

Zimbardo, P. & Gerrig, R. (2005). *Psychology and Life*, Pearson, ISBN: 0205428673, Boston

Automation Control - Theory and Practice
Edited by A D Rodi

ISBN 978-953-307-039-1
Hard cover, 350 pages
Publisher InTech
Published online 01, December, 2009
Published in print edition December, 2009

The present edited book is a collection of 18 chapters written by internationally recognized experts and well-known professionals of the field. Chapters contribute to diverse facets of automation and control. The volume is organized in four parts according to the main subjects, regarding the recent advances in this field of engineering. The first thematic part of the book is devoted to automation. This includes solving of assembly line balancing problem and design of software architecture for cognitive assembling in production systems. The second part of the book concerns different aspects of modelling and control. This includes a study on modelling pollutant emission of diesel engine, development of a PLC program obtained from DEVS model, control networks for digital home, automatic control of temperature and flow in heat exchanger, and non-linear analysis and design of phase locked loops. The third part addresses issues of parameter estimation and filter design, including methods for parameters estimation, control and design of the wave digital filters. The fourth part presents new results in the intelligent control. This includes building a neural PDF strategy for hydroelectric satation simulator, intelligent network system for process control, neural generalized predictive control for industrial processes, intelligent system for forecasting, diagnosis and decision making based on neural networks and self-organizing maps, development of a smart semantic middleware for the Internet , development of appropriate AI methods in fault-tollerant control, building expert system in rotary railcar dumpers, expert system for plant asset management, and building of a image retrieval system in heterogeneous database. The content of this thematic book admirably reflects the complementary aspects of theory and practice which have taken place in the last years. Certainly, the content of this book will serve as a valuable overview of theoretical and practical methods in control and automation to those who deal with engineering and research in this field of activities.

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821