Towards an Understanding of Hierarchical Architectures

Christian Goerick, *Member, IEEE*

***Abstract*—Cognitive systems research aims to understand how cognitive abilities can be created in artificial systems. One key issue is the architecture of the system. It organizes the interplay between the different system elements and thus, determines the principle limits for the performance of the system. In this contribution, we focus on important properties of hierarchical cognitive systems. Therefore, we first present a framework for modeling hierarchical systems. Based on this framework, we formulate and discuss some crucial issues that should be treated explicitly in the design of a system. On this basis, we analyze and compare several well-established cognitive architectures with respect to their internal structure.**

***Index Terms*—Behavior space, cognitive architecture, hierarchical architecture, sensor space, system design.**

I. Introduction

COGNITIVE systems are gaining substantial interest in the cognitive and developmental robotics community. The underlying hypothesis is that the organization of the internal processing architecture is the key element in understanding and building intelligent artifacts. Several proposals of such cognitive architectures are overviewed in [1]. We share the view that the internal processing architecture is one of the most crucial research issues. However, most current research activities focus more on the construction of artifacts rather than on the understanding and the condensation of insights gained thereby.

Hierarchical architectures are an important subclass of cognitive architectures, for which we provide a better understanding. We present and elaborate Systematica, a framework for modeling hierarchical architectures. Based on this framework, we discuss some important properties of hierarchical architectures in general and analyze more concretely some established ones.

II. Related Work

The body of related work can be subdivided into three different groups. The first group is concerned with the analysis of architectures, mainly from the formal standpoint of software validation. An example from this group is [2], where a component-based framework for modeling systems is proposed with the aim to establish flexible systems that are global deadlock-free, individual deadlock-free on the component level, and safe in interaction. This type of approach is formal verification oriented, focusing less on architectural aspects necessary for establishing cognitive functions.

The next group is concerned with proposing concrete architectures for contributing to the advancement of robotic system instances. Noteworthy examples of this group are [3]–[6], and the overview from [1]. Most contributions from this group make concrete proposals that address some properties of cognitive systems. This ranges from specific items to comprehensive architecture proposals for complete artifacts.

The final group comprises of research on software architectural issues for large scale intelligent systems, see, for example, [7]–[10]. The emphasis here is on the construction of software environments that facilitate the implementation of several different cognitive architectures. The proposals differ in the constraints the software environments incorporate. They range from very general tools to software packages with a specific architectural bias. In most contributions, the bias is rather implicit with respect to the targeted cognitive abilities. A different approach is presented in [11] and related publications, where the link between the cognitive architecture concepts and the software framework implementation is more explicit.

Our aim is to have a general and explicit discussion of architectural issues closely linked to the properties of cognitive systems for advancing the understanding of architectures and principles. We neither address formal stability analysis nor implementation oriented software environments. Rather, we focus on modeling hierarchical system architectures for a deeper understanding and explicit comparison of their internal structures for the first time.

III. Systematica

We introduce the framework for modeling hierarchical architectures. The pure nomenclature has been presented in [12], here we now present the comprehensive framework including mathematical modeling and an elaboration of cognitive architecture properties. Systems formulated in Systematica are subdivided into units or loops. Each identifiable processing unit or loop i is described by a set of features and spaces (see Fig. 1 for reference).

Definition 1: D_i denotes the internal dynamics or process of unit i. It processes independently and asynchronously of all other units.

Definition 2: The input space X of the full systems is a vector space spanned by the exteroception and proprioception.

Definition 3: The sensory space $S_i(X)$ of unit i is a projection of a subspace of the full sensory space X.

Manuscript received February 25, 2010; revised September 16, 2010; accepted October 20, 2010. Date of publication November 01, 2010; date of current version March 16, 2011.

The author is with the Honda Research Institute Europe GmbH, D-63073 Offenbach, Germany (e-mail: christian.goerick@honda-ri.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAMD.2010.2089982

1943-0604/$26.00 © 2010 IEEE

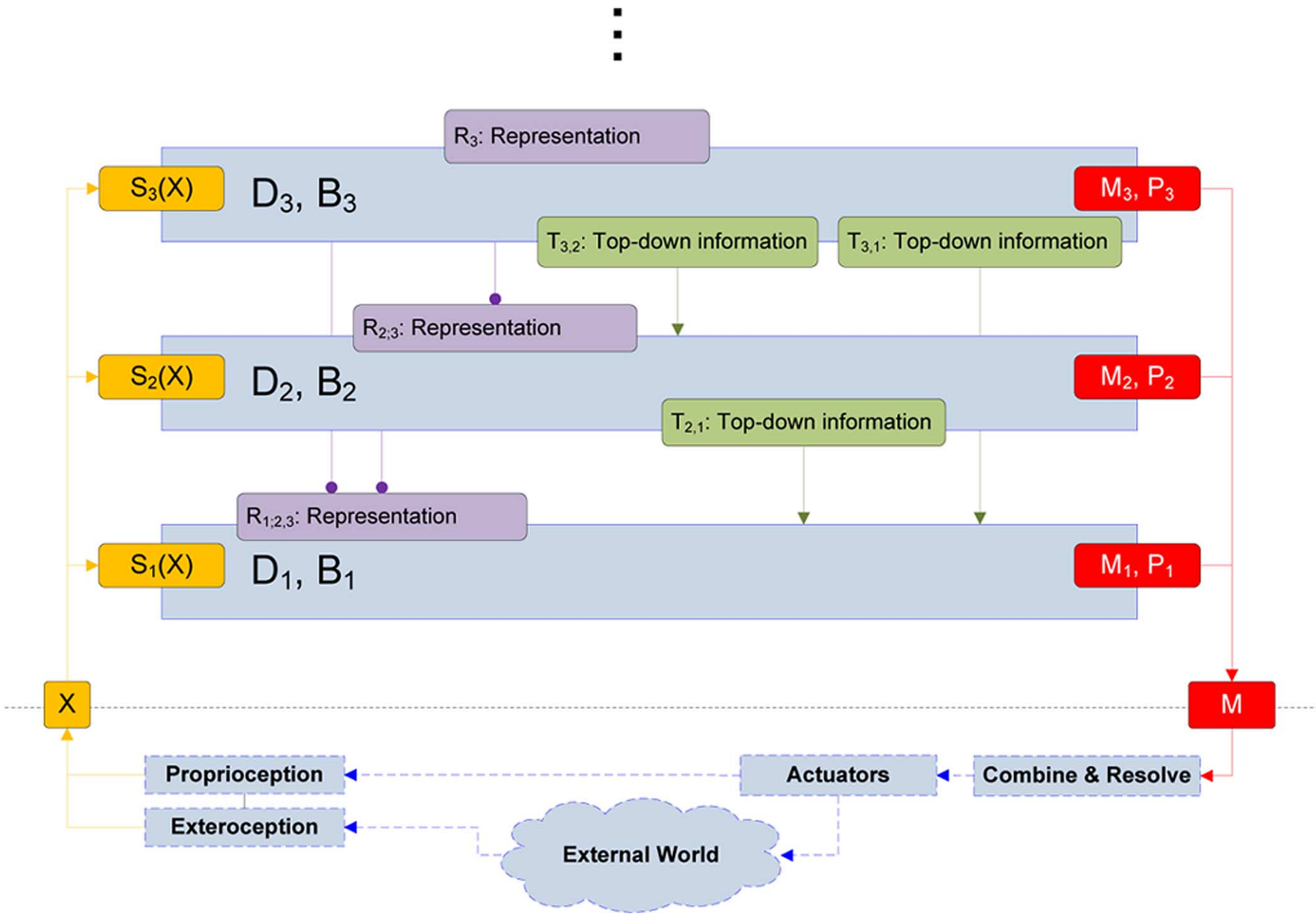

Fig. 1. Generic example of a Systematica schematic showing a system with three layers.

Definition 4: The motor space M_i is a vector space spanned by the motor commands of unit i.

Definition 5: The priority space P_i is a vector space spanned by the priorities of the motor commands of unit i.

Definition 6: The representation space R_i of unit i is a vector space spanned by all publicly observable states of the unit i. R_i is observable for all units n with $n > i$. If unit n observes the representation space of unit i we denote this by appending the subscript n to the representation space $R_{i;n}$.

Definition 7: The top–down information space $T_{i,m}$ between unit i and unit m is a vector space spanned by the information that can be communicated between those two units initiated by unit i. It is only defined for $i > m$.

Definition 8: The behavior space B_i is a vector space spanned by behavior characterizing variables of unit i.

Definition 9: The processing D_i may depend on the respective sensory space $S_i(X)$, all top–down information $T_{n,i}$ for $n > i$ and all representation $R_{m;i}$ for $m < i$.

Definition 10: The conflict resolution decides based on the priorities P_i which motor command to execute in case of conflicting motor commands.

Definition 11: The behavior space B of the entire system is defined as the vector product $B = \bigoplus_i B_i$.

The definitions given above characterize the system and its elements. All defined entities may depend on time. The index i represents the level in the hierarchy. In a developmental sense, it can be associated to an order of creation.

Definition 9 defines the full set of possible dependencies for unit i within the system. The dependency on only a subsets is possible and corresponds to the character of the overall system as discussed in Section VI.

There are two ways for exchanging information between two units within the systems. In Definition 6, it is stated that R_i is observable by the unit i itself and all units n with $n > i$. This means all units on a higher level of the hierarchy can observe the representations of all lower levels. This observation does not require an involvement of the observed module, and therefore does not influence its internal processing. The information is actively collected by unit n. This way of accessing information allows for a loose coupling in the bottom–up direction between communicating units. In contrast to this, Definition 7 states an active sending of information from unit i towards unit m with $i > m$. This may have an immediate effect on the process D_m, and all depending entities like the representations and the behavior. Those definitions together give the hierarchy a clear direction by defining two different mechanisms for upward and downward oriented communication, respectively. Without such a direction the order of units would be arbitrary. See also Section V-D for a continuative discussion of the coupling between units.

The combination and conflict resolution as stated here is not to be understood as the primary instance for such cases but rather as the last resort. Conflicts and combinations must be treated as priority issues between and within of the units of the architecture, e.g., according to the biological principles of inhibition and disinhibition.

The rational for introducing of a motor space and a behavior space for each unit may seem redundant at the first glance. In Section IV-F, we discuss the importance of this subdivision.

IV. Modeling Motivation

Systematica is aiming at a minimal, but meaningful framework for characterizing hierarchical systems and describing the internal dependencies. Therefore, we describe systems by the means given by the Definitions 1 to 11.

A. Processing Hierarchies

Processing hierarchies are common to many models of artificial cognitive architectures as already reviewed in [1]. They are also one common means of describing the internal mechanisms of the brain from a neurobiological point of view [13], [14]. From an early psychologist point of view, hierarchies and especially the transition between different levels in hierarchies for building abstractions, has been of major interest [15].

We assume a hierarchy consists of units, the most general case is that units work independently of each other. Any kind of synchronization can be introduced later by means of communication. This is captured in Definition 1.

B. Sensory Spaces

The next basic question that has to be addressed is "How does the unit perceive the external world and its own embodiment?" This is addressed by the sensory spaces as stated in Definition 2. Sensors transfer physical or chemical properties of the system or the world into a form that is suited for the internal processing of the system. Based on this definition of the sensor, as well as its actual manifestation, internal representations will have certain characteristic properties. Besides selectivity to specific signal modalities sensors are in general direction and location dependent, or exhibit a certain resolution of the signals they measure. The space of all extero- and proprioception is denoted by X, the corresponding subspace a specific unit i is working on as input is denoted by $S_i(X)$. The transition from X to $S_i(X)$ may involve processing like subsampling or projecting for creating derived entities. This clarifies the character of the input to the internal dynamics of the corresponding unit. Different sensory spaces can contribute to the measurement of the same physical entity in a complementary manner, i.e., they can provide different "views" of the same physical or chemical entity. This is for example known from the magno-, parvo-, and konio-cellular visual pathway in the brain, where each pathways focuses on different aspects of a scene like color, motion, or texture with different resolutions [16]. We consider the possibility of a direct access also of higher levels to their own views of the world as important in order to understand abstractions and cognition, see also Section V-A on confinement. A similar type of modeling has also been put forward in [17].

C. Motor Commands and Priorities

Motor commands are common in every modeling of architectures that can emit actions of the embodiment. Every unit may in principle emit motor commands as stated in Definition 4. This is common to the modeling in behavioral robotics like in [17]–[21] and less pronounced in internal processing oriented modeling like [22] and [23]. In biology, there is also evidence for direct access from higher levels of the hierarchy to the motors and actuators, see for example [24]. This may not correspond to the predominant signal flows, but is in some cases necessary for the acquisition of completely new motions. The difference between lower and higher levels is mainly that lower levels act on a coarser level of the sensory signals and do not allow for a fine control of actuators. A very fine analysis of sensory signals and a corresponding fine control of, e.g., finger motions is subject to cortical and not subcortical regions of the brain [14].

D. Representations

In Definition 6, all publicly observable internal states and data of a unit i are formalized. It is assumed that those representations are useful for the internal processing of the respective unit, i.e., that R_i is meaningful in the context of unit i. Within the system, the representations of one unit might additionally be useful for other units, as they can rely on those "intermediate" processing results for their own processing. The involved communication and the gain in performance is a crucial issue in the research of cognitive systems, it may be the key to understanding abstractions in artificial systems. The importance of this issue is reflected in the debate about central representations versus decentral representations as started with the advent of behavioral robotics [18]. In biology, there is evidence for the reuse of representations as for example in the relation of the visual cortex to the superior colliculus. The target for the next gaze direction as determined by the superior colliculus is observed by the cortex for preparing internal representations for the gaze shift [25]. A similar reasoning applies to the area AIP, where the coarse information about graspable objects is maintained, which is observed by the premotor cortex and used for configuring and target setting of the motor cortex [26]. Since we are dealing with hierarchical systems, we do not model the full access of any unit to any representation as already argued in Section III.

E. Top–Down Information

The representations model the use of the intermediate processing results of a unit i as useful information for other units in the system. As stated in Definition 6 the observation of this information by other units has no influence on the unit i itself. The top–down information modeling as stated in Definition 7 addresses the injection of data from unit n into unit i. It models the employment or the control of the processing D_i of unit i by unit n. This is common in the modeling of architectures that control physical actions, like, e.g., [18] and [27]. This concept of top–down information can be characterized by indirectly controlling a process rather than directly controlling an actuator. Another concept of top–down information is the goal-oriented modulation of lower–level stabilization mechanisms. Control processes in the brain perform a basic stabilization and allow higher areas to modulate those stabilizations according to some goal. This is, e.g., the case for the balance and the upright standing of the human body that is maintained by the brain stem (mid brain, hind brain, and medulla oblongata) [28]. The higher areas in the brain rely on those functional loops. A third form of top–down information is the biasing of processing in perception, like priming a visual search. The control of human gaze relies in bottom–up, as well as this kind of top–down information [29].

F. Behavior Space

It is important to clarify the relations between internal and externally observable state changes of the system. There may be internal state changes that are not observable, e.g., prediction and subsequent selection before execution. The state changes of one unit are governed by internal dynamics. The internal dynamics emit signals to the physical actuators. Those signals are called motor commands. The motor commands yield physical actions of the system that can externally be observed. The physical actions have a time course. This can be a trajectory of an end effector or an audible sound signal. Those time series are not arbitrary, but do exhibit a character that depends on the internal dynamics.

The behavior space is spanned by the variables describing this character. If a unit i modulates another unit m by top–down information $T_{i,m}$, then the original behavior space of unit i is expanded by the portion of the behavior space of unit m that is controllable via $T_{i,m}$. See also Section V-A on confinement for another important aspect of the behavior space of units. In other words, the behavior space characterizes the effects a unit can cause in a more abstract way than the local motor commands and in a more comprehensive way by including the effects that top–down information can cause.

A specific behavior is a point or a trajectory in the behavior space. This is an observer independent characterization of the externally observable effect elicited by the internal dynamics of the system. There may be different meanings or semantics that can be attributed to the behaviors. Conversely, the same semantics can be attributed to different internal dynamics and behavior spaces. This is for example the case if a desirable state of the world and the system can be achieved by different units. The description of the behaviors is unit centric and depends solely on the internal dynamics and the corresponding parameters.

In summary, the M_i are the immediate local descriptions of the actions of the unit i; the B_i describe more comprehensive skills or effects from the perspective of the artifact. The unit i may cause effects not just by sending direct motor commands M_i, but also by sending top–down information $T_{i,m}$ to other units. The effects may be applicable in more than one context which implies that one behavior may have different semantics depending on the context. The unit i may not need to model its own semantics, but higher levels may do so. See Section V-B for a discussion about the relation between behaviors and semantics.

This distinction between motor command space, behavior space, and semantics allows for a clear separation of physical effects, system oriented skills, and their meanings. We consider this to be important if we would like to understand how cognitive systems make abstractions and acquire meanings.

V. Architectural Properties

The system modeling capability of Systematica permits addressing some architectural biases explicitly as subsequently described. We consider those properties as crucial for cognitive architectures.

A. Behavioral and Sensory Confinement

Consider a system where only the lowest level $i = 1$ has motor commands M_1 and spans a behavior space B_1. All the higher level units employ only top–down information $T_{n,1}$ for evoking certain behaviors of the system. As a consequence, the behavior space of the full system is confined to B_1. This applies to all kinds of learning and development that might take place at higher levels. As a consequence, B_1 is either sufficiently rich for being able to cope with all future requirements or it has to change over time. A structured approach to overcome this limitation is to have higher levels that can directly emit motor commands bypassing all lower levels. Especially in the case of continuously developing artifacts exploring new behavior spaces this possibility is important.

Similar to behavioral confinement, sensory confinement may occur for all levels i that do not have a unique sensory space $S_i(X)$, but rather rely on representations R_m for $m < i$. This limits the perception of the higher levels to the sensory spaces of the lower levels.

B. Separation of Behaviors and Semantics

In Section IV-F, we have discussed the terms behavior and semantics and have made a clear separation between them. For the behaviors and the behavior space, we have argued that they are unit centered entities. For the semantics, we have argued so far that they can be attributed to the behavior of a unit, but that they can be external to the respective unit. We now elaborate two distinctive points of semantics. First, the semantics are still external to the respective unit but internal to the system, i.e., explicitly modeled within another unit observing the behavior generating unit. And second, the semantics are modeled completely outside the system by the (human) observer. For a reactive system that consists of units mapping sensory perceptions directly to motor commands or corresponding top–down information, there is no necessity to have an internal model of the semantics of the contributing units. Nevertheless, external observers can still label the observed behaviors in semantic terms rather than in terms related to the internal dynamics. Additionally, the observed system can be constructed according to those external models, but there is no explicit representation containing the semantics internal to the system. Most existing hierarchical artificial systems fall in this category.

The first category is of higher importance to us because they address the internal acquisition and treatment of semantics. Systems falling in this category contain units that explicitly model the meanings or semantics of lower level units and the lower level units are to a large extent semantics free or semantically polyvalent. Why should a system exhibit such kind of an internal structure? The decisive reason is because it should be beneficial for its existence. The instances usually associated with this benefit within living beings or artifacts are *needs*, which drive the development towards an improved satisfaction of those needs. In this sense, needs represent a general internal absolute evaluation. Explicit semantic units emerge as stereotypical building blocks (or concepts) for satisfying the needs of the system. Having them represented explicitly makes them accessible for prediction and planning. Having those elements represents the step from a reactive system to a prospective system having a higher probability to satisfy its needs than merely reacting.

C. Hierarchical Internal Representations and Grounding

The principal schematic depicted in Fig. 1 may suggest an intent to implement an incremental hierarchy of units that all access the sensors directly and issue motor commands also quite directly. The possibility to represent this case is certainly necessary, but one major aim here is to understand how higher layers can perform based on representations R_m already created by lower layers, or vice versa, how the behaviorally relevant structuring introduced by lower layers provides the fundament for higher level processing in later layers. Formally, for higher layers this implies $S_i(X) \equiv 0$ and $M_i \equiv 0$. By using an internal behaviorally relevant representation R_i as the basis of internal processing, we ensure that all derived abstractions are grounded.

The representations R_i are initially the internal states of the dynamics D_i. Being embedded in the hierarchy they are observable for all higher levels $n > i$, and they may be modulated by higher levels n via the top–down modulation $T_{n,i}$. What is the benefit of such kind of representations? It is the decomposition of the outside world and the system dynamics in a behaviorally relevant way. In this sense, they provide a meaningful platform for all higher levels $n > i$ that can directly benefit from this decomposition in terms of enriched processing results in addition to the direct sensory spaces. An example is the incremental refinement of perceptions. Lower levels can provide rough perceptions and corresponding representations that can be used by higher levels as initial starting points for a refined analysis. For our understanding this is one of the key elements in hierarchical systems.

D. Temporal Dependencies

Temporal dependencies, or the necessity for synchronization, are a major issue in complex systems. Every processing stage provides its results with a specific latency, i.e., all representations R_i or motor commands M_i have a delay with respect to the corresponding sensory stimulus. Those delays can add up with increasing levels of hierarchies within a system. If the overall system performance is sufficiently high those delays may not matter and the issue is solved in a trivial way. For a better understanding, we want to explicitly deal with the problem of temporal dependencies.

It is a general design goal to avoid temporal dependencies in the system design and consider the different levels as autonomous entities. They might exchange information by observing lower level representations or sending top–down information, but they do not depend on that information in a tightly coupled way. This approach of *loose coupling* maintains the autonomous character of the units, but allows for an exchange of information. It can be explicitly addressed by the modeling introduced by Systematica.

Definition 12: The coupling between unit i and unit n via representation $R_{i;n}$ is loose if, and only if (iff) the process D_n is not blocking on a specific timing of $R_{i;n}$.

This type of coupling is for unit i loose by definition. In other words, the coupling is loose if the observing process D_n is not waiting for a specific update of information in R_i.

Definition 13: The coupling between unit i and unit n via top–down information $T_{n,i}$ is loose iff the process D_i is not blocking on a specific timing of $T_{n,i}$.

This type of coupling is for unit n loose by definition, since there is no feedback information in the top–down information.

Both conditions aim at maintaining the inherent temporal autonomy of the respective processes. This does not contradict a possible prompt reaction to temporal changes in R_i or $T_{n,i}$. One possible means for realizing a loose coupling even if specific information is necessary at a specific time is to employ predictions.

E. Plasticity and Learning

Plasticity and learning are major issues in cognitive systems. Here, we want to clarify the effects of plasticity on the overall system, i.e., the dependencies that have to be considered.

Definition 14: The strict condition for local plasticity of unit i is that no top–down information space $T_{n,i}$ for $n > i$ is changed, no top–down information space $T_{i,m}$ for $i > m$ is changed, and no unit p with $p > i$ is observing the representation R_i.

If the condition as stated in Definition 14 is met, the effects of all changes to $S_i(X)$, D_i, R_i, B_i, and M_i are local to unit i. We call this intraunit plasticity. The overall behavior space B may still change according to Definition 11. Changing the priority P_i will also have a global effect, but the causes can clearly be attributed to one specific unit. If condition 14 is not met, the coupling between the units introduces global effects of local changes that depend on the relations of units. We call this interunit plasticity. A third form of changes is structural plasticity. Units may be deleted or created by some developmental process. This can be modeled by comparing the description of the system before and after the insertion or deletion of a unit. We consider the separation into the categories of plasticity and learning beneficial because they have different structural impacts on the system. They can help understanding the internal dependencies of different architectures as discussed in VI.

F. Hierarchical Versus Incremental

If we describe systems as compositions of layers, we can distinguish between hierarchical and incremental systems. Hierarchical systems are composed of a stack of layers without satisfying further constraints. Incremental systems are hierarchical systems where lower layers can already perform some meaningful behaviors without input from higher layers. This autonomy of lower layers is a major conceptual difference in system architectures that should explicitly be addressed.

VI. Architectures Classes and Instances

We now describe and compare some architecture classes and instances by means of Systematica. They represent some landmarks in the spectrum of architectures. There are variants of these architectures which may deviate from the formulation presented subsequently, but we adhere mainly to the originally introduced concepts.

The architectures have been selected for the following reasons: The subsumption architecture and the three layers architectures represent two classical, but diametrically opposed starting points for researching intelligent systems. One of the original reasons to propose the subsumption architecture was to show an alternative way to planning–based systems as represented by the three layer architectures. The HCogAff and the iCub architecture are currently very actively researched and are sufficiently unique for being analyzed and discussed. The architecture will be reformulated by means of our framework and evaluated according to the aspects from Sections V-B–V-F.

A. Applying Systematica

The discussions in this section also serve as examples for the modeling of systems with Systematica. The general question is how to constructively apply the framework for modeling hierarchical systems. The direct way is to design a system already from the beginning by means of the framework. This has been done based on a preliminary version of the framework in [30]. We will support this approach by a tool supporting this kind of design, but this is beyond the scope of this paper. Analyzing existing systems by means of the framework requires the identification of the elements according to the Definitions 1 to 11. The most crucial part is the determination of the order in the hierarchy. For systems already defined in a hierarchical pattern, this is usually a one-to-one mapping. Systems which are intrinsically hierarchical, but are not explicitly formulated as such require the determination of the order. Since the order is given by the dependencies by the top–down information $T_{n,i}$ and the representations R_i, they have to be determined first, the remaining structure follows subsequently.

B. Subsumption Architecture

The subsumption architecture can shortly be summarized as a composition of hierarchical control loops, where each loop represents a direct link from perception to action [18], see Fig. 2 for an abstracted form of this architecture. The loops are therein called "layer of competence." Higher–level loops have a higher priority for issuing motor commands and can suppress the sensing or the issuing of motor commands of lower levels. More formally, the top–down information $T_{n,i}$ is either input or output suppression of the module i, or the substitution of the input of module i.

Each unit covers a clear subspace of the behaviors and directly produces motor outputs, i.e., $M_i \neq 0 \forall\, i$, and the corresponding priorities are strictly hierarchical: $P_n > P_i$ for $n > i$. The behavior spaces B_i do not have an explicit relation. There may be an implicit relation between the sensory subspaces $S_i(X)$ as discussed in Section IV-B. The subsumption architecture does not suffer from sensory or behavioral confinement as addressed in Section V-A since $S_i(X) \neq 0$ and $M_i \neq 0 \forall\, i$.

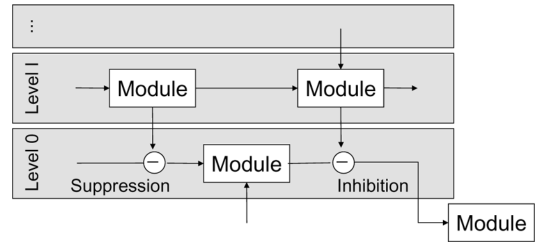

Fig. 2. Schema of subsumption architectures, drawing after [21].

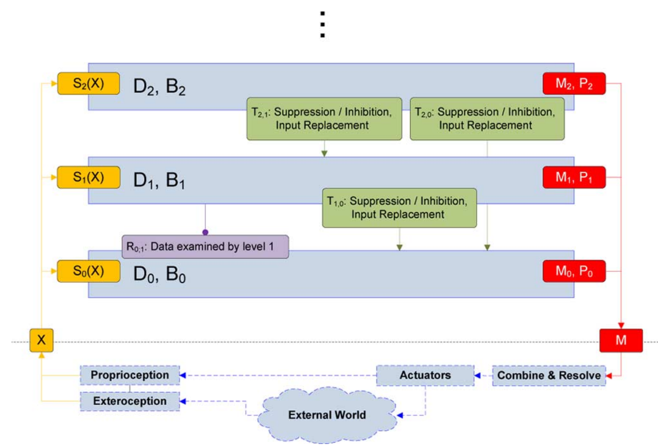

Fig. 3. Subsumption architecture formulated in Systematica.

This allows in principle for an open system wrt. the acquisition of novel abilities, but those possibilities are not exploited as discussed below.

As depicted in Fig. 2, the subsumption architecture incorporates primarily no publicly accessible representations R_i. However, in the related literature the notion "modules inspecting the data pathways of other modules" can be found, indicating the existence of representations. Especially in [18], the phrase "level 1 is able to examine data from the level 0 system" can be found. This can be modeled by introducing the representation $R_{0;1}$. The concept of hierarchical representations as discussed in Section V-C is nevertheless not pronounced within this approach. All internally produced data is grounded. Fig. 3 shows the subsumption architecture formulated in Systematica.

The decomposition of the subsumption architecture is strictly incremental since lower levels are independent of higher levels and form a complete operational system (see Section V-F). This independence is also underlined by the loose coupling according to Definitions 12 and 13 since there is no handshaking between levels of competence intended [18]. This is beneficial for an incremental construction of lower levels independently from future higher levels, but assigning the independence such a high priority that hierarchical representations are excluded as discussed in Section V-C is a major drawback for cognitive or intelligent systems.

The subsumption architecture also does not comprise means for plasticity as discussed in Section V-E except for structural plasticity by adding new layers in a design process. There is also no separation between the motor commands, behaviors and the associated semantics (Section IV-F). Both aspects together preclude the development of cognitive abilities.

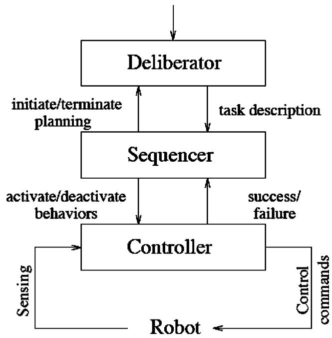

Fig. 4. Schema of three layer architectures taken from [27].

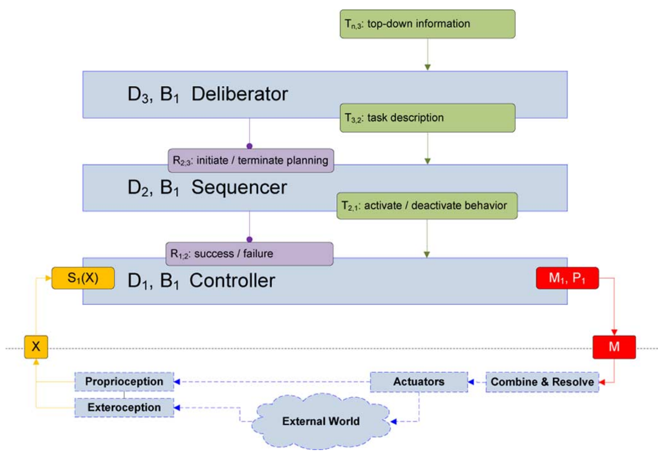

Fig. 5. Three-layer architecture formulated in Systematica.

The general character of the subsumption architecture is clearly oriented towards incremental action generation rather than the generation of hierarchical abstractions or semantics. Originally proposed as an alternative route to intelligence it clearly lacks means for abstractions and learning as required in cognitive architecture.

C. Three Layer Architectures

Three-layered architectures (see Fig. 4) are a very general class of architectures that usually consist of the controller, sequencer and deliberator layers (or similar name sets [27]). This means $i \in \{1,2,3\}$. Only the lowest level (controller) is directly interacting with the outside world, i.e., $S_1 \neq 0$, $S_{i>1}(X) = 0$, $M_1 \neq 0$, $M_{n>1} = 0$, $P_{n>1} = 0$. They are strictly hierarchical in the sense that only adjacent layers communicate, i.e., $T_{n,i<n-1} = 0$, $D_i = f(R_{i-1}, T_{i+1,i})$. According to the reasoning in Section V-A, the behavior of the overall system is confined to the behavior space B_1 spanned by the first layer. The same applies for the sensory space: it is confined to the sensory space $S_1(X)$ of the first layer. See Fig. 5 for an illustration.

There is no general indication for plasticity and learning (Section V-E) and no separation of behaviors and semantics (Section V-B). A system organized according to this architecture paradigm is clearly hierarchical, but not incremental (Section V-F). The hierarchy of internal representations (Section V-C) consists of low volume synchronization events or data concerning success or failure of controller actions. In this sense, the hierarchy is clearly action, but not abstraction oriented. The possible benefits of hierarchies are limited to benefits of top–down information as described in Section IV-E. Depending on the implementation of the coupling it may be loose according to Definitions 12 and 13. This may ease synchronization problems, but does not contribute to the autonomy of the layers. Except for the lowest layer autonomy would also not make sense in this kind of architecture. A decomposition in a classical three layer architecture does not promote abstractions in the sense of cognitive systems.

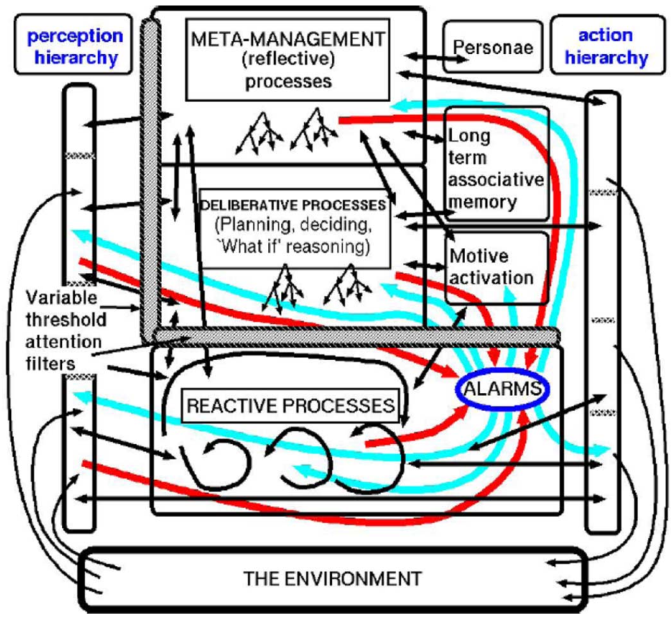

Fig. 6. Schematics of HCogAff taken from [31].

In summary, the major character of this class of architectures is the achievement of a given task by a planned series of actions. Major elements for cognitive architecture are clearly missing. Considering the classical three layers as part of a larger cognitive system would require additional layers on top with different characteristics.

D. The HCogAff Architecture

The CogAff architecture was suggested by Sloman [31]. It is a proposal for a class of architectures or framework rather than a specific instance, with the aim to understand the role of affective phenomena in cognitive architectures. A more concrete proposal developed from CogAff is the HCogAff. We will consider the HCogAff architecture here since the CogAff is too general for a purposeful discussion. For an overview of the architecture, see Fig. 6.

It consists mainly of three layers [reactive mechanisms ($i = 1$), deliberative processes including motive activation ($i = 2$), reflective processes ($i = 3$)], but there are significant differences to the three layer architectures as discussed in the previous section. See Fig. 7 for an illustration of the modeling. All levels have access to sensory modalities $S_i(X) \neq 0 \forall i$ and emit motor commands $M_i \neq 0 \forall i$, hence there is no confinement as discussed in Section V-A, which is beneficial for the development of cognitive abilities. The representations of the reactive layer (R_1) and the deliberative layer (R_2, long term associative

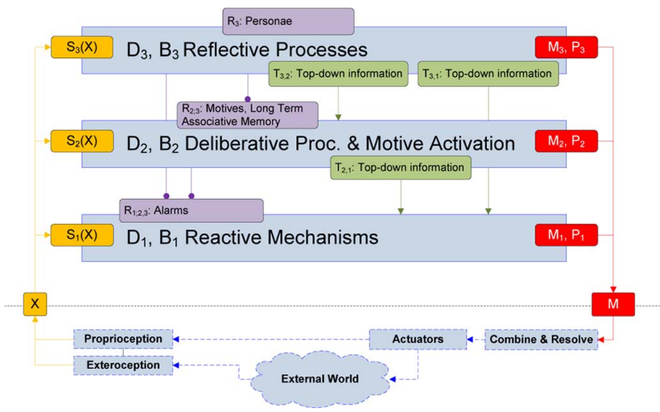

Fig. 7. HCogAff architecture formulated in Systematica.

memory) are accessed by the respective higher layers, but it is not clear if they are intended to be hierarchical in the sense of Section V-C. The same applies to the personae representation R_3. There is general rich top–down information flow $T_{n,i}$ from all higher to all lower layers.

Nevertheless, it is not clear which kind of top–down information (Section IV-E) is intended by the HCogAff framework. A similar reasoning applies to the temporal dependencies (Section V-D). It is not clearly stated what kind of dependencies are considered or what kind of effect they will have on the overall architecture.

Plasticity and learning (Section V-E) are discussed as one major target of the affections and emotions within the system, but not explicitly modeled. This applies also to some other issues discussed in the text like the distinction between majorly processing of external $(S_i(X))$ or internal (R_i) information, which are not explicitly modeled.

With the separation of deliberative processes from the reactive ones, the subdivision between motor actions and their meaning (Section IV-F) is addressed. The proposal is clearly hierarchical (Section V-F), but not intended to be incremental as a major design goal, i.e., the whole system has to be considered from the beginning.

The HCogAff architecture was proposed as a framework for the scientific treatment of cognitive architectures. As discussed above, several elements are included or satisfied by the proposal in order to allow for cognitive abilities. The open points (majorly the character of top–down information and the temporal dependencies) need to be better specified in order to assess the potential of this approach or to guide a practical approach within this framework.

E. The iCub Cognitive Architecture

A first proposal of the iCub cognitive architecture is given in [32]. It consists of three major layers, as can be seen in Fig. 8. Fig. 9 illustrates the modeling in Systematica. The lowest layer $(i = 1)$ is constituted by the phylogenetically self-organized perceptuomotor skills. Those are several parallel processes that can compete or cooperate. In summary, those processes perform all the sensing and all the actions of the body. Hence, the sensory space $S_1(X)$ and the behavior space B_1 spanned by this layer

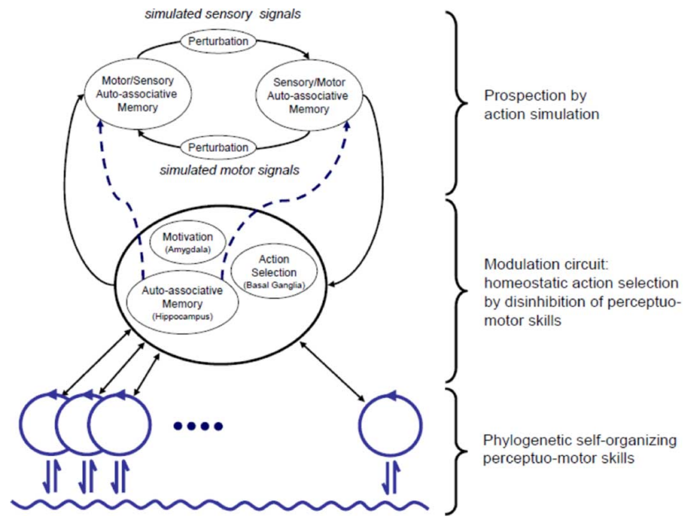

Fig. 8. Schematics of the iCub cognitive architecture taken from [32].

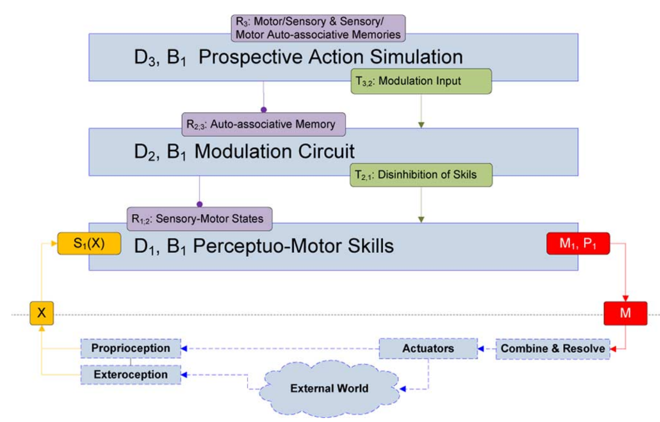

Fig. 9. The iCub architecture formulated in Systematica.

confine all the sensing and all the behaviors of the overall systems as discussed in Section V-A. This layer also provides representations R_1 that can be employed by the next layer. This next layer $(i = 2)$ is called the modulation circuit and performs a homeostatic action selection by disinhibition of perceptuomotor skills by providing top–down information $T_{2,1}$ to layer 1. The autoassociative memory of this layer constitutes the representation R_2 which is used for both processing within this layer, as well as in the next higher layer. The highest layer $(i = 3)$ is called prospective action simulation and performs predictions based on two autoassociative memories forming the representation R_3. The output of this layer is top–down information $T_{3,2}$ influencing the processing of layer 2. There is no direct influence on layer 1, i.e., $T_{3,1} = 0$. For the two higher layers, all sensory spaces $S_2(X)$ and $S_3(X)$, as well as the motor commands M_2 and M_3 are zero.

The proposed architecture is incremental in the sense of Section V-F since the lower layer can perform without the necessity of the higher layers. The architecture is also strictly hierarchical since only neighboring layers communicate, which also underlines the confinement of the sensor and behavior spaces to the ones of layer 1. This also excludes any kind of hierarchical sensory refinement as discussed in Section IV-B.

Temporal dependencies (Section V-D) may be intended to be loose, but they are not explicitly modeled. The representations R_i are truly hierarchical as discussed in Section V-C since they depend on each other incrementally. The separation of motor action, behaviors, and semantics (Section IV-F) is not explicitly modeled, but could be realized by means of the hierarchical representations.

Plasticity and learning (Section V-E) are a central point in this proposal. The authors intend to consider all three kinds of plasticity: intra, inter, and structural, whereas the last two are not modeled explicitly.

The confinement requires the phylogenetically developed skills to be sufficiently rich in order to have a sufficiently rich behavior space B_1 at the lowest level or to rely on the intramodule plasticity of level 1. It is surprising to find confinement in a biologically inspired architecture, because it seems contradictory to what we can learn from biology (see explanation in Section IV-B). One reason could be an implicit focus on the learning in the higher layers, but this should be clearly stated.

VII. Discussion and Summary

The lack of means to explicitly understand hierarchical architectures is a major obstacle in the advancement of cognitive systems. Hence, we presented an explicit modeling of hierarchical systems called Systematica. Based on this modeling, we stated some important properties that should be explicitly addressed by any architecture proposal.

Those elements (Systematica plus the properties) represent a structured approach to modeling and discussing hierarchical architectures as an important class of cognitive architectures. We demonstrated this approach by applying it to several existing architectures. This approach reveals the general characteristics and properties of the respective architectures. These may include possible deficiencies that may require further investigation. We think that future work must focus on the explicit formal aspects of architectures in order to gain scientific insight into artificial cognition. In this sense, Systematica is an initial step towards understanding cognitive systems. This should apply to architecture classes, as well as concrete system instances that are researched and implemented in order to show the current level of understanding as well as to help formulating the next research questions.

Acknowledgment

The author would like to thank all the colleagues at the Honda Research Institute Europe for the inspiring discussions and the continuous fruitful collaboration in pursuing the goal of creating and understanding artificial cognitive systems. Especially, the author would like to acknowledge that the feedback from I. Mikhailova and B. Dittes was invaluable for maturing the presented concepts.

References

[1] D. Vernon, G. Metta, and G. Sandini, "A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents," *IEEE Trans. Evol. Comput.*, vol. 11, pp. 151–180, Nov. 2007.

[2] G. Gossler and J. Sifakis, "Composition for component-based modeling," *Sci. Comput. Program.*, vol. 55, no. 1–3, pp. 161–183, 2005.

[3] K. Kawamura, R. A. Peters, II, R. E. Bodenheimer, N. Sarkar, J. Park, C. A. Clifton, A. W. Spratley, and K. A. Hambuchen, "A parallel distributed cognitive control system for a humanoid robot," *Int. J. Humanoid Robot.*, vol. 1, no. 1, pp. 65–93, 2004.

[4] O. Brock, A. Fagg, R. Grupen, R. Platt, M. Rosenstein, and J. Sweeney, "A framework for learning and control in intelligent humanoid robots," *Int. J. Humanoid Robot.*, vol. 2, no. 3, pp. 301–336, 2005.

[5] S. Chernova and R. C. Arkin, "From deliberative to routine behaviors: A cognitively inspired action-selection mechanism for routine behavior capture," *Adapt. Behav.*, vol. 15, no. 2, pp. 199–216, 2007.

[6] R. Arkin, M. Fujita, T. Takagi, and R. Hasegawa, "Ethological modeling and architecture for an entertainment robot," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Seoul, Korea, 2001.

[7] V. Andronache and M. Scheutz, "APOC—A framework for complex agents," in *Proc. AAAI Spring Symp.*, Palo Alto, CA, 2003, pp. 18–25.

[8] J. Kramer and M. Scheutz, "Radic—Towards a general method for integrating reactive," *Int. Trans. Syst. Sci. Appl.*, vol. 3, no. 3, pp. 183–192, 2007.

[9] P. Fitzpatrick, G. Metta, and L. Natale, "Towards long-lived robot genes," *Robot. Autonom. Syst.*, vol. 56, no. 1, pp. 29–45, 2008.

[10] A. Ceravola, M. Stein, and C. Goerick, "Researching and developing a real-time infrastructure for intelligent systems—Evolution of an integrated approach," *Robot. Autonom. Syst.*, vol. 56, no. 1, pp. 14–28, 2008.

[11] N. Hawes, M. Zillich, and J. Wyatt, "BALT & CAST: Middleware for cognitive robotics," in *Proc. IEEE RO-MAN 2007*, Jeju Island, Korea, Aug. 2007, pp. 998–1003.

[12] C. Goerick, B. Bolder, H. Janssen, M. Gienger, H. Sugiura, M. Dunn, I. Mikhailova, T. Rodemann, H. Wersing, and S. Kirstein, "Towards incremental hierarchical behavior generation for humanoids," in *Proc. IEEE-RAS Int. Conf. Human.*, Pittsburgh, PA, 2007.

[13] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, *Principles of Neural Science*, Fourth ed. New York: McGraw-Hill, 2000.

[14] L. W. Swanson, *Brain Architecture: Understanding the Basic Plan*. London, U.K.: Oxford Univ. Press, 2002.

[15] L. Vygotsky, *Mind in Society*. Boston, MA: Harvard Univ. Press, 1978.

[16] R. Rodieck, *The First Steps in Seeing*. Sunderland, MA: Sinauer Associates, 1998.

[17] T. J. Prescott, P. Redgrave, and K. Gurney, "Layered control architectures in robots and vertebrates," *Adapt. Behav.*, vol. 7, pp. 99–127, 1999.

[18] R. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. 2, no. 1, pp. 14–23, 1986.

[19] V. Braitenberg, *Vehicles: Experiments With Synthetic Psychology*. Cambridge, MA: MIT Press, 1984.

[20] R. C. Arkin, *Behavior-Based Robotics*. Cambridge, MA: MIT Press, 1998.

[21] R. Pfeifer and C. Scheier, *Understanding Intelligence*. Cambridge, MA: MIT Press, 1999.

[22] J. E. Laird, A. Newell, and P. S. Rosenbloom, "SOAR: An architecture for general intelligence," *Artif. Intell.*, vol. 33, no. 1, pp. 1–64, Sep. 1987.

[23] P. Langley, "An adaptive architecture for physical agents," in *Proc. IEEE/WIC/ACM Int. Conf. Intell. Agent Technol.*, Compiegne, France, 2005, pp. 18–25.

[24] P. A. Chouinard and T. Paus, "The primary motor and premotor areas of the human cerebral cortex," *The Neuroscientist*, vol. 12, no. 2, pp. 143–152, 2006.

[25] M. A. Sommer and R. H. Wurtz, "What the brain stem tells the frontal cortex. I. oculomotor signals sent from superior colliculus to frontal eye field via mediodorsal thalamus," *J. Neurophysiol.*, vol. 91, pp. 1381–1402, 2004.

[26] A. Battaglia-Mayer, R. Caminiti, F. Lacquaniti, and M. Zago, "Multiple levels of representation of reaching in the parieto-frontal network," *Cereb. Cortex*, vol. 13, pp. 1009–1022, 2003.

[27] E. Gat, "On three-layer architectures," in *Artificial Intelligence and Mobile Robots*. Cambridge, MA: MIT/AAAI Press, 1997.

[28] D. Purves, G. J. Augustine, D. Fitzpatrick, C. Hall, A.-S. Lamantia, J. O. McNamara, and S. M. Williams, *Neuroscience*. Sunderland, MA: Sinauer Associates, 2004.

[29] J. M. Henderson, "Human gaze control during real-world scene perception," *TRENDS Cogn. Sci.*, vol. 7, no. 11, pp. 498–504, Nov. 2003.

[30] B. Dittes, M. Heracles, T. Michalke, R. Kastner, A. Gepperth, J. Fritsch, and C. Goerick, "A hierarchical system integration approach with application to visual scene exploration for driver assistance," in *Proc. 7th Int. Conf. Comput. Vis. Syst.*, Liege, Belgium, Oct. 13–15, 2009, pp. 255–264.

[31] A. Sloman, "What are emotion theories about?," in *Proc. AAAI Spring Symp., Techn. Rep.*, 2004, pp. 128–134.

[32] D. Vernon, G. Sandini, and G. Metta, "The iCub cognitive architecture: Interactive development in a humanoid robot," in *Proc. IEEE Int. Conf. Develop. Learn.*, London, U.K., Jul. 2007, pp. 998–1003.

Christian Goerick (M'10) received the Dipl.Ing. degree in electrical engineering and the Ph.D. degree in electrical engineering and information processing from the Ruhr-Universität Bochum, Bochum, Germany, in 1993 and 1998, respectively.

From 1993 to 2000, he was with the Institute for Neural Computation, Ruhr-Universität Bochum, where he became a Research Assistant, Doctoral Worker, and Project Leader in 1993, and a Postdoctoral Worker, Project Leader, and Lecturer in 1998. The research was concerned with biologically motivated computer vision for autonomous systems and learning theory of neural networks. Since 2000, he has been at the Honda Research Institute Europe, Offenbach, Germany, now holding the position of a Chief Scientist. His special interests include embodied brain-like intelligence covering research on behavior-based vision, audition, behavior generation, cognitive robotics, car assistant systems, systems architecture, as well as hard- and software environments.