Behavioral Control of Cognitive Agents Using Database Semantics and Minimalist Grammars

Ronald Römer*, Peter beim Graben*, Markus Huber*, Matthias Wolff*, Günther Wirsching† and Ingo Schmitt*
*Brandenburg University of Technology Cottbus-Senftenberg, Cottbus, Germany
†Catholic University Eichstätt-Ingolstadt, Eichstätt, Germany

***Abstract*—Knowledge representation and processing, learning and adjusting knowledge models, communication and interaction as well as problem solving are important skills of cognitive agents. But, in order to share the learned knowledge with other communication participants, the agent must pay attention to maintaining its own functionality when interacting with the physical environment. To comply with this requirement, the meaning of perceptions and the consequences of actions must be understood. This suggests that even the non-verbal exchange of information is based on the foundations of semiotics. In this work, we show that we can model non-verbal interactions with the same linguistic means as verbal communication. We demonstrate this by solving the bidirectional translation problem of symbolic sequences into semantics through the use of minimalist grammars. To verify this approach, we turn back to the well-known example of a cognitive mouse agent living in a maze world and model the interaction and behavioral control using database semantics in a fully deterministic setting. Finally, we propose a unifying perspective for non-verbal interaction and verbal communication as well as justify its relevance for Cognitive Infocommunications.**

I. Introduction

Several theoretical psychologists adopt an evolutionary perspective and classify cognitive systems within the scope of *final systems* [1]. This perspective comprises four ascending stages: the mechanistic level (cause and effect), the semantic level (adaptation), consciousness and self-reflection. Final systems are characterized by the first two stages. The psychologists have emphasized, that low level organisms as well as machines belong to this class of systems. Both share the following properties: They are able to avoid damages caused by dysfunction and deception and they allow to learn minimizing dysfunction and deception from mistakes. Further, the information exchange with the environment is based on actuators and sensors and is extended to semantics and pragmatics. These properties help to increase the chances of survival by adaptation to changing environments. Hence, from a technical perspective we suggest that a final system corresponds to an autonomous agent, whose behavior is dependent on necessities, motives and permissions supporting well-being.

To build such an agent and in order to transfer human cognitive abilities to technical systems [2], we proceed in two steps. First, we merge the essentials of biological- and technical basic systems. From biology we adopt the principle of a cell. This includes memory and basic processing (nucleus), differentiated metabolic processing (behavior) as well as networking and well-being control by the membrane. From cybernetics we adopt the principle of an embedded turing machine. It contains memory and basic computations (turing machine), differentiated application processing (behavior) as well as networking and safety control by an interface. In both cases

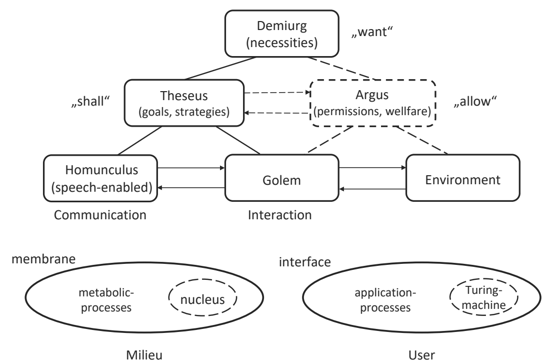

Figure 1. Integration of biological- and cybernetic basic systems by ideas from theoretical psychology. Necessities are given by a system designer, motives and goal directed behavior are related to a figur called "Theseus". The ability to avoid damages is associated with a figure called "Argus". The PAC concerns the information exchange with the environment. It is separated into interaction and communication. Interaction is within the scope of a figure called "Golem", it comprises the control of physical system components and the rating of target objects. The "Golem" itself is controlled by a figure called "Homunculus" which is speech-enabled additionally.

the interaction with the environment can be described by the so called perception-action-cycle (PAC). With the second step we integrate additional psychological aspects: necessities, motives and permissions. Necessities are predefinded by a system designer. They cannot be changed or adapated by the organism. Motives and permissions are associated with the capability to pursue goals and to rate target objects. Both aspects determine the behavior of an agent. Finally, we get a hierarchic organized scheme that helps to clarify structure and capabilities of (rational) cognitive systems (see figure 1). In order to keep apart structure and capabilities we are using figures from ancient mythologies. These figures were not introduced because we presume such figures in a cognitive agent, but because they allow us to more clearly explain the cognitive functions we need.

On the one hand, the integrated model comprises structural properties like control of behavior and well-being by a feedback loop using an embedded knowledge model [3] as well as the use of actuators and perceptors. On the other hand we have four salient cognitive capabilities: knowledge representation, learning and adaptation of the knowledge model, communication and interaction as well as problem solving (includes reification and planning, see [4]).

The key role in cognitive systems comes to the knowledge model. Building and using such a model is based on the ability to exchange information with the environment by communication or interaction. In terms of final systems, both processes

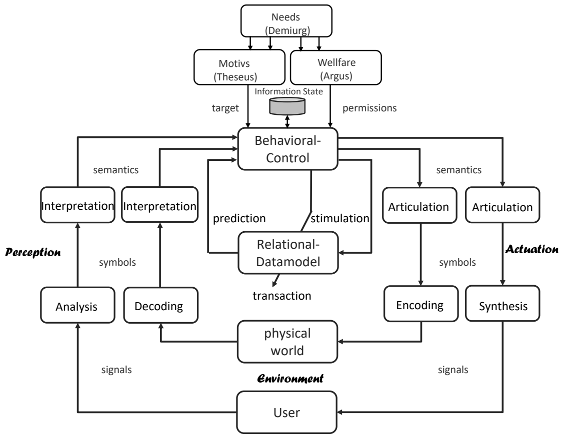

Figure 2. Double cognitive cycle based on an embedded knowledge model. The inner cycle corresponds to the interaction between the "Golem" and the environment (non-verbal information exchange). The outer cycle is used for the communication between the speech-enabled "Homunculus" and any other cognitive agents (verbal information exchange). The transfer of information occurs between two systems with different cognitive abilities (e.g. between humans and cognitive agents, denoted as *Inter-Cognitive Communication* [2]).

must be able to handle semantic representations. This suggests that even the non-verbal exchange of information is based on the foundations of semiotics and should be modeled by linguistic description means as well. As justification, we quote two arguments borrowed from [5]: (1) "Without a carefully built physical grounding any symbolic representation will be mismatched to the sensors and actuators. These groundings provide the constraints on symbols necessary for them to be truly useful". (2) "The analysis of non-verbal cognition is needed in order to be able to plausibly explain the phylogenetic and ontogenetic development of language from earlier stages of evolution without language".

The author concludes that semantic representations and grammatical algorithms are formalisms that can be used as well for describing non-language processes as for describing natural and artificial languages. That is, in applications where the agent is speech-enabled and the communication with a user is relied to a common environment the cognitive structure results in a double cognitive cycle (see figure 2). We denote the inner cycle as interaction and the outer cycle as communication. In addition, we assume that all cognitive activities correspond to manipulations of the knowledge model. This model represents the physical world through a network of semantic objects – described by measurable attributes – that are related to each other. Objects, attributes and relationships are the information of interest to which both non-verbal interaction and verbal communication refer. However, it should be noted that non-verbal interaction refers to object instances, while verbal communication operates on object types. By comparing the information transmitted through interaction and communication, the question can be answered, whether the observed object instance matches to the communicated object type.

The scientific contribution of this work is twofold: (1) Modeling non-verbal information exchange by linguistic description means. (2) Translation from symbol representations to semantic representations (interpretation) and vice versa (articulation) using minimalist grammars [6].

II. Problem Statement

Let us consider the well known mouse-maze problem [4], [3]. The mouse lives in a simple $N \times M$ maze world that is given by a certain configuration of walls. For the mouse to survive, a target object is located on some place in the maze. To understand how the mouse can reach the target object we have to look at the inner cognitive cycle in figure 2. First, the "Golem" measures the current position in a reference system ($x-$, $y-$ coordinates) and determines the presence or absence of cheese $(c, \neg c)$ at this position. Subsequently, the measurement information needs to be encoded as a time linear symbolic sequence and has to be transmitted to a sensory buffer. Next, the "Homunculus" observes the symbol sequence and has to translate it into a semantic representation. Based on this observation and a prediction from the knowledge model, the "Homunculus" can decide for one of its actions (e.g. north, east, south, and west). In the next step the corresponding action representation has to be translated into a symbolic action sequence which needs to be transmitted to the "Golem" again. Finally, after action sequence decoding, the "Golem" can execute the action by the associated actuators. With the environments response the inner PAC starts again as long as the target object has not been reached.

In order to build a knowledge model from non-verbal information two conditions must be met. First, the incoming observations have to be transformed from linearly ordered symbol sequences – delivered from the "Golem" – into semantic representations – received by the "Homunculus". Second, in order to decide "What is the case?" or "What does this observation mean?", the semantic representations need to be rated by truth values. This kind of knowledge can be stored as the result of an exploration phase if no logical contradictions occure. That is, at this level the agent is capable to solve problems by inference and reasoning. This is also true in the orientation phase. To infer the current state, the measurement information has to be compared logically with the model knowledge.

At this point we can summarize the problem statement using a technical language: In order to get a semantically based behavioral control, we have to specify the non-verbal information exchange between the "Homunculus" and the "Golem" by a formal bidirectional linguistic translation. Further, to build a knowledge model we have to store the disambiguated semantic representations of the "Homunculus". Finally, in order to simulate the knowledge model we need a dynamic system that is given by a state space representation.

III. Basics and Definitions

In this section we summarize the most necessary concepts needed for the basic capabilities: state space representation, formal languages, semantic representations and minimalist grammars. We assume that the transduction from signal- to symbol space already exists (e.g. by SST [7]).

A. State Space Representation

To describe the system behavior in terms of cause and effect we distinguish states $z \in \mathcal{Z}$, actions $a \in \mathcal{A}$ and observations $o \in \mathcal{O}$. A behavioral relation is then given by

$$R_V \subseteq \mathcal{Z} \times \mathcal{A} \times \mathcal{Z} \times \mathcal{O},$$

which determines the system dynamics. Further, it corresponds to a network of causal relations. Due to causality, a recursive calculation rule divided in a system- and measurement equation is given, which allows forecasting the system's behavior

$$z' = G(z, a) \text{ with } G: \mathcal{Z} \times \mathcal{A} \to \mathcal{Z},$$
$$o = H(z, a) \text{ with } H: \mathcal{Z} \times \mathcal{A} \to \mathcal{O}.$$

In a fully deterministic setting we assume a totally observable world, hence the behavioral relation simplifies to

$$R_V \subseteq \mathcal{Z} \times \mathcal{A} \times \mathcal{Z}.$$

To decide, whether a tuple belongs to this relation or not, the characteristic function $f_{R_V} : \mathcal{Z} \times \mathcal{A} \times \mathcal{Z} \to \{0, 1\}$ can be applied.

B. Formal Language

In order to specify any technical communication between source and sink, we have to arrange an alphabet $\Sigma_A = \{a, b, \ldots\}$ and to establish a language $\mathcal{L} \subseteq \Sigma_A^*$. Words or sentences are then described by an ordered sequence $s = (s_1, s_2, \ldots, s_k) \in \mathcal{L}$, where $s_i \in \Sigma_A,\ \forall i \leqslant k$. If we want to build a language we can devise a grammar $G = (\mathcal{N}, \mathcal{T}, \mathcal{P}, S)$. It is specified by the sets $\mathcal{N}$-nonterminals, $\mathcal{T}$-terminals and $\mathcal{P}$-production rules as well as a start symbol S. In this work we just use right regular grammars that only need two rules: $S \to wS$ and $S \to w$ with $w \in \mathcal{T}$. Based on these rules permissable sentences can be produced or derived. In opposite to the sentence production we can also analyze a sentence by a method that is called parsing.

C. Semantic Representation

In semantics we distinguish between objects that are described by attributes and relations between such objects [8]. To represent semantics we will focus on feature-value pairs (FVP), which have a flat structure and correspond to tuples of database relations. The notation used here was taken from [9]. We start from a universe $U = \{A_1, A_2, \ldots, A_m\}$ which comprises a finite set of attributes. Further, we have a set of domains $D = \{D_1, D_2, \ldots, D_m\}$ and a mapping $dom : U \to D$. The values of the different attributes are elements of the associated domains: $\omega_i \in D_i = dom(A_i)$. Based on these definitions we can specify types of objects or relations between objects by a set of attributes $X = \{A_1, A_2 \ldots, A_n\} \subset U$. To identify concrete objects we use attribute names which allow the discrimination of instances, these names are called primary keys. Attribute names that allow the discrimination of individual relations between objects are denoted as foreign keys. Distinguishable objects are defined by a set of mappings, which are called tuples:

$$\tau : \{A_1, A_2, \ldots, A_n\} \to \bigcup_{i=1}^{n} dom(A_i),\ n \leqslant m.$$

Please note, in database notation the elements of tuples are not ordered. This corresponds to the idea that the order of attributes is not relevant for describing the type of any object [9]. An object- or relation type is defined by a relation scheme:

$$R(X) = (A_1 : dom(A_1), A_2 : dom(A_2), \ldots, A_n : dom(A_n)).$$

In database representations a scheme is associated to the head of a relational data table. The entries of a table correspond to a set of tuples. This set of tuples corresponds to a relation, which is given by:

$$R_X \subseteq Tup(X) := \{\tau | \tau : X \to dom(X)\},$$

where $Tup(X)$ is the set of all possible tuples of a relation scheme $R(X)$. The set of all relations over this scheme is denoted as $Rel(X)$.

D. Minimalist Grammars

Minimalist grammars (MG) were introduced to describe natural languages [6]. For non-verbal interaction it is crucial, that MG provide a translation mechanism between linearly ordered perceptions/actions and semi- or disordered semantic representations.[1] Based on such a grammar, knowledge can be formally represented and stored consistently in a relational data model.

Following [11], we regard a linguistic sign as an ordered triple

$$q = (e, t, \sigma)$$

with exponent $e \in E$, semantics $\sigma \in \Sigma$ and a syntactic *type* $t \in T$ that we encode by means of MG in its chain representation [12]. The type controls the generation of the syntactic and semantic structure of an utterance. An MG consists of a data base, the mental lexicon, containing signs as arrays of syntactic, phonetic and semantic *features*, and of two structure-generating functions, called "merge" and "move". For our purposes, it is sufficient to point out some syntactic features: Selectors (e.g. "=S") are required to search for syntactic types of the same category (e.g. "S"). Licensors (e.g. "+k") and licensees (e.g. "−k") are required to control the symbol order at the surface. The symbol "::" indicates *simple*, *lexical* categories while ":" denotes *complex*, *derived* categories ("·" is just a placeholder for one of these signs). A sequence of signs "**q**" is called a *minimalist expression*. For further details we refer to [10]. To preserve the readability, we just repeat the two types of rules here. The MG function "merge" is defined through inference schemata

$$\frac{(e_1, \mathbf{::}{=}f\mathbf{t}, \sigma_1) \quad (e_2, \cdot f, \sigma_2)\mathbf{q}}{(e_1 e_2, \mathbf{:}\mathbf{t}, \sigma_1\sigma_2)\mathbf{q}} \text{ merge-1}, \tag{1}$$

$$\frac{(e_1, \mathbf{:}{=}f\mathbf{t}, \sigma_1)\mathbf{q}_1 \quad (e_2, \cdot f, \sigma_2)\mathbf{q}_2}{(e_2 e_1, \mathbf{:}\mathbf{t}, \sigma_1\sigma_2)\mathbf{q}_1\mathbf{q}_2} \text{ merge-2}, \tag{2}$$

$$\frac{(e_1, \cdot{=}f\mathbf{t}_1, \sigma_1)\mathbf{q}_1 \quad (e_2, \cdot f\mathbf{t}_2, \sigma_2)\mathbf{q}_2}{(e_1, \mathbf{:}\mathbf{t}_1, \sigma_1)\mathbf{q}_1(e_2, \mathbf{:}\mathbf{t}_2, \sigma_2)\mathbf{q}_2} \text{ merge-3}, \tag{3}$$

Correspondingly, "move" is given through

$$\frac{(e_1, \mathbf{:}{+}f\mathbf{t}, \sigma_1)\mathbf{q}_1(e_2, \mathbf{:}{-}f, \sigma_2)\mathbf{q}_2}{(e_2 e_1, \mathbf{:}\mathbf{t}, \sigma_1\sigma_2)\mathbf{q}_1\mathbf{q}_2} \text{ move-1}, \tag{4}$$

$$\frac{(e_1, \mathbf{:}{+}f\mathbf{t}_1, \sigma_1)\mathbf{q}_1(e_2, \mathbf{:}{-}f\mathbf{t}_2, \sigma_2)\mathbf{q}_2}{(e_1, \mathbf{:}\mathbf{t}_1, \sigma_1)\mathbf{q}_1(e_2, \mathbf{:}\mathbf{t}_2, \sigma_2)\mathbf{q}_2} \text{ move-2}. \tag{5}$$

IV. METHOD AND MODEL

Our aim is to transfer the symbolic state space representation to a semantic state space representation. To this end, we have to give symbols an internal structure. For example, the state previously designated by the symbol z can be described by

[1] With regard to the linguistic discussion (e.g. relations to the Attribute Value Logic) of minimalist grammars, we refer to [10] in this conference volume.

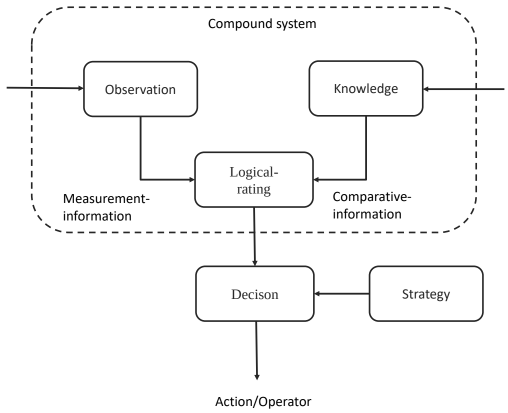

Figure 3. Compound System. Behavioral control relies on a logical compound system that compares measurement information and model knowledge. Subsequently it decides for a strategy based action.

a set of feature-value pairs $\{X \rightarrow x \in dom(X),\ Y \rightarrow y \in dom(Y),\ C \rightarrow c \in dom(C)\}$. Further, we have to store this representations in a relational database (knowledge base). In order to integrate the relational database into the PAC we will translate symbolic sequences into semantic representations and vice versa using the minimalist grammar approach.
Finally, to understand "What is the case?" or "What does the observation mean?", we have to compare the semantic representation of the perception with some of the learned semantic representations in the knowledge base. If we find matches, then the perceptual semantic representation corresponds to the current state of the environment. If not, an adaptation or coping process is triggered [4]. In semiotics, this process is called denotation and is located in a logical compound system (see figure 3). Based on this altered model the cognitive agent should be able to pursue goals, understand situations, select cost optimal actions, predict its consequences and finally, should be able to rate target objects. In order to apply the new model, we have to specify attributes and relation schemes of states, actions and behavior and we have to determine the corresponding relations that are needed for the explorations phase: $States = \{X, Y, C\} \subset U$, $R(States) = (X : dom(X), Y : dom(Y), C : dom(C))$ and $R_{States} \subseteq Tup(States)$. $Action = \{\Delta X, \Delta Y,\} \subset U$, $R(Action) = (\Delta X : dom(\Delta X), \Delta Y : dom(\Delta Y))$ and $R_{Action} \subseteq Tup(Action)$. $Behavior = \{States, Action\} \subset U$, $R(Behavior) = (States : dom(States), Action : dom(Action), States' : dom(States'))$ and $R_{Behavior} \subseteq Tup(Behavior)$. The next two sections handles the bidirectional translation problem and describes the behavioral control based on semantic representations.

V. Linguistic Processing

Based on the formal languages $\mathcal{L}_{obs}$ and $\mathcal{L}_{act}$ we solve the interpretation problem and show that we can describe the articulation problem by the same formalism.

A. Translation Formalism - Interpretation

First, we consider any situation that is measured, encoded and transmitted by the "Golem". In doing so, the "Golem" uses a set of roles $\mathcal{R} = \{r_i | 0 \leqslant i \leqslant n+1\}$ that determine the word positions of the n- measurements within start- and final symbols. In any maze situation, for example, a message may be encoded in the following form $s = (w_0, w_1, w_2, w_3, w_4) = (\langle s \rangle, x, y, c, \langle \backslash s \rangle)$. Indeed, in non-verbal messages the symbols do not contain any implicit semantics so it must be explicitly specified. For this reason, an agreement between the "Golem" and the "Homunculus" is required. This is achieved by the mapping $\beta : \mathcal{R} \rightarrow States$. It is a bijective mapping of roles to the attributes of the relation scheme $R(States)$. Using this mapping, the semantics of w_i for $0 < i < n+1$ is defined by $\sigma(w_i) := \beta(r_i) \rightarrow \omega_i \in dom(\beta(r_i))$ and $\sigma(w_0) = \sigma(w_{n+1}) = \varnothing$. Based on this role dependent semantics any observed situation results in a semantic representation that is given by the tuple

$$\begin{aligned}\tau_{obs} = & \bigcup_{i=0}^{n+1} \sigma(w_i) \\ = & \{A_1 \rightarrow \omega_1 \in dom(A_1), \ldots, A_n \rightarrow \omega_n \in dom(A_n)\}.\end{aligned}$$

Next, we demonstrate how the "Homunculus" applies the MG-formalism to the message received from the "Golem". The formalism needs to be feeded with a linearly ordered sequence of symbols $s \in \mathcal{L}_{obs}$ and has to respond with a tupel $\tau_{obs} \in Tup(States)$.
To better understand the formalism we use the following indexed derivation scheme: $S_0 \rightarrow w_0 S_1$, $S_1 \rightarrow w_1 S_2$, $S_2 \rightarrow w_2 S_3$, $S_3 \rightarrow w_3 S_4$ and $S_4 \rightarrow w_4$, in which the words w_1, w_2, w_3 contain the values $x \in dom(X)$, $y \in dom(Y)$ and $c \in dom(C)$. To build a language structure that contains the derivation tree as well as the corresponding semantics the formalism uses the *key and lock principle*. Transferring this principle to the MG-formalism means, that a selector (e.g. "=S") requires a syntactic type of the same category (e.g. "S"). Licensors and licensees are required to control the symbol order at the surface. The MG-formalism for relation schemes works as follows: First, the incoming words are segmented and mapped to their linguistic signs. These signs have been stored in advance in a minimalist lexicon. To build linguistic signs that are associated with a relational scheme we need only three types of signs:

- Start: $(\langle s \rangle, :: S_0\ {-k}, \varnothing)$,
- End: $(\langle \backslash s \rangle, :: {=}S_n {+k}\ S_{n+1}, \varnothing)$,
- Mapping: $(w_i, :: {=}S_{i-1}\ {+k}\ S_i\ {-k}, \sigma(w_i))$.

Based on a *sequence of linguistic signs* the formalism changes alternately between the rules merge-3 and move-1. But due to the fact, that the semantics of exponents was defined explicitely by a set of mappings, we have to make two small changes. Firstly, we substitute σ through $\sigma(e)$ in both rules and secondly, we define $\sigma_1 \sigma_2 := \sigma(e_1) \cup \sigma(e_2)$. With these minor changes we obtain the formalism for the interpretation. After we have processed the start type, we proceed with the recursive steps (mappings) and stop with the end type:

Start:

$$\cfrac{\cfrac{(w_1, :: {=}S_0{+k}\ S_1{-k}, \sigma(w_1)) \qquad (w_0, :: S_0{-k}, \varnothing)}{(w_1, : {+k}\ S_1{-k}, \sigma(w_1)) \qquad (w_0, : {-k}, \varnothing)}}{(w_0 w_1, : S_1{-k}, \bigcup_{i=0}^{1} \sigma(w_i))}.$$

Recursion:

$$\frac{(w_2, ::=S_1{+}k\ S_2{-}k, \sigma(w_2))\ (w_0w_1, :S_1{-}k, \bigcup_{i=0}^{1}\sigma(w_i))}{(w_2, :{+}k\ S_2{-}k, \sigma(w_2))\ (w_0w_1, :{-}k, \bigcup_{i=0}^{1}\sigma(w_i))}$$

$$\frac{(w_2, :{+}k\ S_2{-}k,\ \sigma(w_2))\ (w_0w_1, :{-}k, \bigcup_{i=0}^{1}\sigma(w_i))}{(w_0w_1w_2, :S_2{-}k, \bigcup_{i=0}^{2}\sigma(w_i))}.$$

$$\frac{(w_3, ::=S_2{+}k\ S_3{-}k, \sigma(w_3))\ (w_0w_1w_2, :S_2{-}k, \bigcup_{i=0}^{2}\sigma(w_i))}{(w_3, :{+}k\ S_3{-}k, \sigma(w_3))\ (w_0w_1w_2, :{-}k, \bigcup_{i=0}^{2}\sigma(w_i))}$$

$$\frac{(w_3, :{+}k\ S_3{-}k,\ \sigma(w_3))\ (w_0w_1w_2, :{-}k, \bigcup_{i=0}^{2}\sigma(w_i))}{(w_0w_1w_2w_3, :S_3{-}k, \bigcup_{i=0}^{3}\sigma(w_i))}.$$

End:

$$\frac{(w_4, ::=S_3{+}k\ S_4, \varnothing))\ (w_0w_1w_2w_3, :S_3{-}k, \bigcup_{i=0}^{3}\sigma(w_i))}{(w_4, :{+}k\ S_4, \varnothing))\ (w_0w_1w_2w_3, :{-}k, \bigcup_{i=0}^{3}\sigma(w_i))}$$

$$\frac{(w_4, :{+}k\ S_4, \varnothing)\ (w_0w_1w_2w_3, :{-}k, \bigcup_{i=0}^{3}\sigma(w_i))}{(w_0w_1w_2w_3w_4, :S_4, \bigcup_{i=0}^{4}\sigma(w_i))}.$$

If we apply this formalism to the relational scheme $R(States)$ we obtain the following semantic representation for any observed situation

$$\tau_{obs} = \bigcup_{i=0}^{n+1}\sigma(w_i)$$
$$= \{X \to x \in dom(X), Y \to y \in dom(Y), C \to c \in dom(C)\},$$

which are disordered attribute-value mappings as required.

B. Translation Formalism - Articulation

After the action decision, the "Homunculus" must apply the MG-formalism to the semantic action representation. The formalism now works in reverse order and needs to be feeded with a tuple $\tau_{act} \in Tup(Action)$ and has to respond with a linearly ordered sequence of symbols $a \in \mathcal{L}_{act}$. Any selected action tuple is given by the form

$$\tau_{act} = \{\Delta X \to \Delta x \in dom(\Delta X), \Delta Y \to \Delta y \in dom(\Delta Y)\}.$$

To produce a message $a = (w_0, w_1, w_2, w_3) = (\langle s\rangle, \Delta x, \Delta y, \langle\backslash s\rangle)$, the values associated to the relational scheme $R(Action)$ must be mapped to their linguistic signs. These signs have been stored as well in advance in the minimalist lexicon. Please note, that for articulation we can use the same three types as in interpretation. As before, the formalism changes alternately between the rules merge-3 and move-1. In this processing direction the values of the attributes are transferred to the symbols of the message. The only difference to the interpretation is that the formalism is now based on a *set of linguistic signs*. In our application the elements of this articulation set are: $(w_0, ::\ S_0\ {-}k, \varnothing)$, $(w_3, ::{=}S_2{+}k\ S_3, \varnothing)$, $(w_1, ::{=}S_0\ {+}k\ S_1\ {-}k, \sigma(w_1))$ and $(w_2, ::{=}S_1\ {+}k\ S_2\ {-}k, \sigma(w_2))$. Applying the production rules the derivation would give the following rule sequence: $S_0 \to w_0S_1$, $S_1 \to w_1S_2$, $S_2 \to w_2S_3$ and $S_3 \to w_3$. This results in $S_0 \to w_0w_1w_2w_3$, in which the root of the derivation tree S_0 comprises the whole action sequence.

The MG-formalism preserves this order through the selectors of the associated exponents: After the formalism is starting with the sign $(w_0, ::\ S_0\ {-}k, \varnothing)$ the word w_1 is required by the selector "$=S_0$" of the sign $(w_1, ::{=}S_0\ {+}k\ S_1\ {-}k, \sigma(w_1))$. Next, the word w_2 is required by the selector "$=S_1$" of the sign $(w_2, ::{=}S_1\ {+}k\ S_2\ {-}k, \sigma(w_2))$. Finally, the word w_2 is required by the selector "$=S_2$" of the last sign in the articulation set $(w_3, ::{=}S_2{+}k\ S_3, \varnothing)$. The complete concatenated string $a = (w_0, w_1, w_2, w_3) = (\langle s\rangle, \Delta x, \Delta y, \langle\backslash s\rangle)$ is inferred then by the last move-1 rule and corresponds to the action sequence which can be transmitted to the "Golem".

VI. Modeling Behavior

A. State Classification

From figure 3, we know that the compound system must either choose cost-optimal actions to find target objects or should rate the found objects before deciding on consummatory actions. To make such decisions, it is necessary to know the current state of the environment or to know the state of the target object. In the following, we focus on finding target objects. The measurement information is given by $\tau_{obs} \in Tup(States)$ and the comparative information is provided by the tuples of the relation R_{States} which were stored during the exploration phase. This amount of tuples can be limited by a prediction, e.g. by applying the system equation. The comparison can be carried out by a binary classifier or by the evaluation of a logical predicate. This can be expressed through its characteristic function

$$f_{R_{States}}(\tau_{obs}) = \begin{cases} True, & \text{if } \tau_{obs} \in R_{States}, \\ False, & \text{if } \tau_{obs} \notin R_{States}. \end{cases}$$

The evaluation of this gives the answer to the questions What is the case?" or "What does this observation mean?". The next two subsections explain, how the agent chooses cost- or utility optimal actions. More details in terms of cost-optimal action decision can be found in [13].

B. Goal Directed Action Decisions

This kind of decisions falls into the sphere of the figure "Theseus". Here, the benefit-based agent optimizes the effectiveness of its decisions $d \in D$ for action $a \in A$ on the expected environmental state $z \in \mathcal{Z}$. The corresponding model is based on a sequential Markov decision process (MDP). In this model, the Bellman equation is used to calculate an expected utility function $U(Z, D)$ [14]. With this equation, the information value $V^{\pi}(z)$ is calculated iteratively for each state $z \in \mathcal{Z}$ for a given strategy π.

$$V^{\pi}(z) = \sum_{d} p(d|z) \sum_{z'} p(z'|d, z)[r(z, d, z') + V^{\pi}(z')].$$

The conditional propability $p(z'|d,z)$ is associated with state transitions and $r(z,d,z')$ correspond to local rewards. The optimal strategy to maximize the expected benefit is obtained with

$$\pi^* = p(d^*|z) = \underset{p(d|z)}{\arg\max}\ [U(Z,D)].$$

The utility function $U(Z,D)$ is also called Q-function.

C. Consummatory Action Decision

Decisions for final actions (e.g. take food) can also be modeled based on the principle of maximum benefit. Here, an additional categorization of the found target objects like "edible" and "inedible" takes place. We follow [1] and consider categories as tools of the perception apparatus. Decisions, whether a target object satisfies a necessity or not belong to semantics as well. But, with reference to figure 1 this kind of decisions falls into the sphere of the figure "Argus". To model consummatory action decisions, we need a stochastic decision process that includes the binary random variables $Z = \{\text{edible}, \text{inedible}\}$, B (any binary features) and $D = \{\text{accept}, \text{reject}\}$. In this case, the *conditional* expected utility function $U(Z|B)$ has to be optimized. This is due to the fact, that object states can only be indirectly observed by the measurement of features. The optimal strategy to maximize $U(Z|B)$ is obtained with

$$P(d^*|b) = \underset{p(d|b)}{\arg\max}\ [U(Z|B)].$$

VII. Conclusion and Outlook

We have shown that modeling of non-verbal information exchange with linguistic means can be achieved by using minimalist grammars. Based on a minimalist lexicon this formalism unifies the translation from symbolic representations to semantic representations and vice versa. To represent semantics we have used database semantics in the form of tuples of feature-value-pairs. These tuples can be summarized into relations so that a knowledge model can be built up. Important cognitive skills such as knowledge processing and problem solving as well as communication and learning are located at this level of information. These capabilities can be used even more efficiently if cognitive agents can also handle natural language. Recent research in computational linguistics suggests that minimalist grammars provide the most promising approach to describe natural languages [12].

Based on the structure in figure 2 we can outline two future research areas: (1) Merging the linguistic processing of the inner circle and of the outer circle. This would allow the communication participants to exchange and synchronize their ideas about the physical world. Obviously, exploration phases can be significantly shortened if the agents use a common language. Further, in a shared environment interaction provides the physical context for immediate communication between cognitive agents. This would help to resolve ambiguities in verbal communication. (2) Currently, there is a large number of descriptive means for cognitive agents. We have concepts from mathematics, logic, linguistics, decision theory, state space representation and many more. From a system theoretic view, we therefore need a unifying perspective. We think that quantum-based formalisms provide such a perspective. There is a number of indications for this: e.g. ontology-learning and technical communication using Fockspace-representation [3], [15], rating of target objects [16], [13] as well as quantum-based knowledge processing [17]. Of particular scientific value are models that are both *interpretable* (there is a clear relationship between representation and reference) and *explainable* (causal or logical relationships). Both criteria are met by vector symbolic architectures [18], that have already been applied to minimalist grammars [19]. An outstanding feature of quantum mechanical formalisms is the easy transition to nondeterministic scenarios. Thus, we conclude that research into quantum-based cognitive agents may lead to significant advances in modeling of Cognitive Infocommunication [2].

References

[1] N. Bischof, *Psychologie, ein Grundkurs für Anspruchsvolle*. Verlag Kohlhammer, 2009, 2. Auflage.

[2] P. Baranyi, A. Csapo, and G. Sallai, *Cognitive Infocommunications (CogInfoCom)*. Springer, 2015, ISBN 9783319196077.

[3] M. Wolff, M. Huber, G. Wirsching, R. Römer, P. beim Graben, and I. Schmitt, "Towards a quantum mechanical model of the inner stage of cognitive agents," in *2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, 2018, pp. 147–152.

[4] M. Wolff, R. Römer, and G. Wirsching, "Towards coping and imagination for cognitive agents," in *2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, 2015, pp. 307–312.

[5] R. Hausser, *Foundations of Computational Linguistics*. Berlin, Heidelberg, New York: Springer Verlag, 2014.

[6] E. P. Stabler, "Derivational minimalism," in *Logical Aspects of Computational Linguistics*. Lecture Notes in Computer Science, Springer New York, 1997, pp. 68 – 96.

[7] M. Wolff, C. Tschöpe, R. Römer, and G. Wirsching, "Subsymbol-Symbol-Transduktoren," in *Elektronische Sprachsignalverarbeitung 2013*, ser. Studientexte zur Sprachkommunikation, P. Wagner, Ed. Dresden: TUDpress, 2013, vol. 65, pp. 197–204.

[8] P. P. Chen, "The entity-relationship model: Toward a unified view of data," in *Proceedings of the International Conference on Very Large Data Bases, September 22-24, 1975, Framingham, Massachusetts, USA.*, 1975, p. 173.

[9] G. Lausen, *Datenbanken, Grundlagen und XML-Technologien*. Elsevier GmbH, Spektrum Akademischer Verlag, 2005, 1. Auflage.

[10] P. beim Graben, R. Römer, W. Meyer, M. Huber, and M. Wolff, "Reinforcement learning of minimalist numeral grammar," in *2019 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, 2019, submitted.

[11] M. Kracht, *The Mathematics of Language, Studies in Generative Grammar.* Mouton de Gruyter, Berlin, 2003.

[12] E. P. Stabler and E. L. Keenan, "Structural similarity within and among languages," in *Theoretical Computer Science*, 2003, pp. 345 – 363.

[13] R. Römer, P. Graben, and M. Wolff, "Entscheidungstheoretische Modellierung der konsummatorischen Endhandlung – Vergleich von klassischen und quantenmechanischen Ansätzen," in *Elektronische Sprachsignalverarbeitung 2019, Tagungsband der 30. Konferenz*, P. Birkholz, Ed., 2019.

[14] D. Polani, "Information: currency of life?" *HFSP Journal*, vol. Vol. 3, No.5, pp. pp. 307 – 316, 2009.

[15] K. J. Boström, "Concepts of a quantum information theory of many letters," https://arxiv.org/abs/quant-ph/0009052, 2000.

[16] J. R. Busemeyer and P. D. Bruza, Eds., *Quantum Models of Cognition and Decision*. Cambridge University Press, 2012.

[17] I. Schmitt, "QQL: A DB&IR Query Language," *VLDB J.*, vol. 17, no. 1, pp. 39–56, 2008.

[18] P. Smolensky and G. Legendre, *The Harmonic Mind. From Neural Computation to Optimality-Theoretic Grammar*. Cambridge (MA): MIT Press, 2006, vol. 1: Cognitive Architecture.

[19] P. beim Graben and S. Gerth, "Geometric Representations for Minimalist Grammars," *Journal of Logic, Language and Information*, vol. 21, no. 4, pp. 393–432, 2012.