

Modeling Human-Level Intelligence by Integrated Cognition in a Hybrid Architecture

Kai-Uwe Kühnberger, Tonio Wandmacher, Angela Schwering,
Ekaterina Ovchinnikova, Ulf Krumnack, Helmar Gust, Peter Geibel

Institute of Cognitive Science, University of Osnabrück
D-49076 Osnabrück, Germany

Abstract. Various forms of reasoning, the profusion of knowledge, the gap between neuro-inspired approaches and conceptual representations, the problem of inconsistent data input, and the manifold of computational paradigms for solutions of these problems challenge AI models for higher cognitive abilities. We propose the I-Cog architecture as a step towards a solution for these problems. I-Cog is a modular system that is composed of a reasoning device based on analogical reasoning, a rewriting mechanism for the ontological knowledge base, and a neuro-symbolic interface for robust learning from noisy and inconsistent data.

1 Introduction

Since the origins of Artificial Intelligence – based on the fundamental work of Alan Turing [44], the first architecture for neural networks by McCulloch & Pitts [35], the development of higher programming languages like LISP [34], and finally the creation of AI as a discipline at the Dartmouth conference – artificial intelligence has (more or less) strongly been committed to interdisciplinary research and the modeling of higher cognitive abilities.¹ Several important achievements can be identified during the last 50 years with respect to modeling (or supporting) cognitive challenging tasks of humans: state-of-the-art computer programs beat world-class chess champions and intelligent programs support our daily life in various respects, for example, when driving a car, flying a plane, creating an engineer’s CAD constructions, or searching the web for information.

Despite these apparent examples for the success of AI, there are severe problems of AI which can provocatively be described as follows: there is not even an idea of how human-level intelligence² (HLI) in the large can be achieved, taking into account the various forms of capabilities of human beings, for example,

¹ The term *higher cognitive abilities* can be identified with all forms of cognition which essentially include a deliberative aspect like reasoning, planning, game playing, learning, problem solving etc. In particular, purely reactive behaviors or behaviors which can be reduced to mere conditioning are not higher cognitive abilities.

² The term *human-level intelligence* is used in the sense of [7], namely as the problem to integrate many different types of representation formats, reasoning devices, computational paradigms etc., in order to approximate a breadth of intelligence usually ascribed to humans.

concerning reasoning, problem solving, learning, adapting, acting, using natural language etc. In the following we list three classes of such problems.

- The profusion of knowledge [7] and its constant updates.
- The variety of types of reasoning and computational paradigms for modeling human reasoning abilities (compare textbooks in AI).
- The gap between neuro-inspired learning approaches to cognition and symbolic representational approaches [4].

We think that these challenges are at the heart of achieving human-level intelligence, because of the following fundamental problem: The more fine-grained the methods are in developing tools for particular (and isolated) AI applications, the more we depart from the goal of achieving HLI and a unified model of higher cognition.³ This paper aims to propose an architecture that provides a possible solution to model higher cognitive abilities by integrated cognition. We think that an integrated architecture can be considered as a device for achieving HLI.

This paper has the following structure. Section 2 discusses obvious problems in modeling a variety of higher cognitive abilities. Section 3 presents the I-Cog architecture consisting of a reasoning module, a background knowledge rewriting module, and a neuro-symbolic integration module. These modules interact in a non-trivial way described in Section 4. Finally Section 5 summarizes related work and Section 6 concludes the paper.

2 Problems for Modeling Higher Cognition in AI Systems

2.1 Knowledge

Knowledge representation is classically connected with coding entities in the environment by symbolic frameworks. Although such a straightforward logical representation is universal for most knowledge representation formalisms, and appropriate logical calculi ensure that many types of inferences for applications can be performed, there are non-trivial challenges for such a logical approach:

- *Problem of expressive strength:* For many applications first-order logic is simply too expressive. Examples are terminological hierarchies for ontological knowledge [43] or representations of STRIPS-like plans [11]. For other applications first-order logic is simply not expressive enough. Examples are forms of modeling common ground [6], or the usage of standard arithmetic.
- *Dynamic updates of background knowledge:* Whereas background knowledge is commonly considered to be static, human agents constantly update, modify, and learn new knowledge. Furthermore, they can overwrite existing knowledge easily without being threatened by inconsistencies.

³ This claim clearly does not mean that other difficulties for modeling cognition in the large are simple or in some way straightforward to solve. Obviously challenges in computer vision, the modeling of autonomous agents and motor control, or natural language processing are also hard problems. But except for natural language processing, they concern lower cognitive abilities and are not considered here.

Due to the first challenge a profusion of representation formalisms emerged. Currently there is no idea how to reduce these paradigms significantly. The second challenge seems to be recently detected as a problem [24].

2.2 Reasoning

Reasoning abilities of humans can be classified into many types: Just to mention some of them, humans can perform deductions, inductions, and abductions. Furthermore, they are able to perform analogical reasoning steps, non-monotonic inferences, and frequency-based inferences (at least to a certain extent). Additionally, human agents are able to reason with vague and uncertain knowledge and they have the ability to associate certain situations with other similar situations. As a natural consequence of this variety of reasoning types, AI developed a tremendous number of frameworks for the computation of inferences. Unfortunately, these computational paradigms are not fully compatible with each other.

2.3 Neuro-Symbolic Integration

The gap between robust neural learning and symbolic representation formalisms is obvious: whereas symbolic theories are based on recursion and compositionality allowing the computation of (potentially) infinitely many meanings from a finite basis, such principles are not available for connectionist networks. On the other hand, neural networks have been proven to be a robust tool for learning from noisy data, pattern recognition, and handling vague knowledge – classical domains with which symbolic theories usually encounter problems. A potential solution for achieving HLI would require an integration of both approaches.

3 The Modules of the I-Cog Architecture

3.1 Analogical Reasoning

It is a crucial hypothesis of this paper that the establishment of analogical relations between a source and a target domain can be used for many forms of classical and non-classical reasoning tasks [14]. Examples for application domains of analogies are string domains [28], geometric figures [41], problem solving [1], naive physics [10], or metaphoric expressions [21]. Furthermore, analogies are a source of creativity [29] and a possibility to learn from sparse data [20]. Deductions and abductions are implicitly modeled in several systems (e.g. [13]).

In this paper, heuristic-driven theory projection (HDTP) will be used for sketching the expressive power of analogy making [21]. HDTP represents the source and target domains by sets of first-order formulas. The corresponding source theory Th_S and target theory Th_T are then generalized using an extension of anti-unification [40]. Here are the key elements of HDTP:

- Two formulas $p_1(a, b)$ and $p_2(a, c)$ can be anti-unified by $P(a, X)$, with substitutions $\Theta_1 = \{P \rightarrow p_1, X \rightarrow b\}$ and $\Theta_2 = \{P \rightarrow p_2, X \rightarrow c\}$.

Table 1. A simplified description of the algorithm HDTP-A omitting formal details. A precise specification of this algorithm can be found in [21].

| | |
|------------------|--|
| Input: | A theory Th_S of the source domain and a theory Th_T of the target domain represented in a many-sorted predicate logic language \mathcal{L} . |
| Output: | A generalized theory Th_G such that the input theories Th_S and Th_T can be re-established by substitutions. |
| Selection | and generalization of fact and rules. Select an axiom from the target domain (according to a heuristic h). Select an axiom from the source domain and construct a generalization (together with corresponding substitutions). |
| Optimize | the generalization w.r.t. a given heuristic h' . Update the generalized theory w.r.t. the result of this process. |
| Transfer | (project) facts of the source domain to the target domain provided they are not generalized yet. Test (using an oracle) whether the transfer is consistent with the target domain. |

- A theorem prover allows the re-representation of formulas.
- Whole theories can be generalized, not only single terms or formulas.

The underlying algorithm HDTP-A is computing candidates of generalizations relative to Th_S and Th_T (Table 1): first, axioms are chosen from the target according to a heuristic ordering. For these axioms generalizations are computed relative to chosen axioms from the source (also governed by a heuristic). If every axiom from the target is generalized, the algorithm allows a creative transfer of knowledge from the source to the target (governed by the computed generalizations already obtained). We consider the analogy between a water-pipe system and an electric circuit in order to clarify the framework:

(M1) *Current is the water in the electric circuit.*

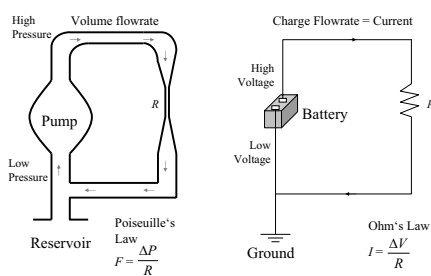


Fig. 1. The analogy between a water pipe system and an electric circuit in a diagrammatic representation. The Figure contains more information than is necessary for an interpretation of the metaphorical description (1).

water system and a closed electric system generalize to an abstract concept $closed(A)$, where A is a variable. The terms *water* and *current* are associated

Figure 1 depicts the situation represented by this analogy.⁴ The analogy associates water-flow in a water pipe system with the flow of current in an electric circuit. An important new conceptualization about electricity can be learned by students using this analogy, namely that current is flowing in a circuit and that a battery has a function similar to a pump in the water pipe system.

We would like to achieve a modeling of metaphor (M1) using HDTP. Table 2 specifies the corresponding concepts in the target and the source domains that are associated with each other. The concepts of a closed

⁴ The figure is based on <http://hyperphysics.phy-astr.gsu.edu/hphys.html>.

Table 2. Examples of corresponding concepts in the source and the target of the analogy between water-flow and the flow of current in an electric circuit. *ws1* denotes an instance of a water pipe system and *es1* an instance of an electric circuit.

| Source | Target | Generalization |
|------------------------------------|---|-----------------------------------|
| <i>water_circuit(ws1,water,p1)</i> | <i>electric_circuit(es1,current,b1)</i> | <i>Circuit(A,C,S1)</i> |
| <i>closed(ws1)</i> | <i>closed(es1)</i> | <i>closed(A)</i> |
| <i>pump(p1)</i> | <i>battery(b1)</i> | <i>Source(S1)</i> |
| <i>pres(p1)>0→flow_in_c(w)</i> | <i>pres(b1)>0→flow_in_c(c)</i> | <i>pres(S1)>0→flow_in_c(C)</i> |
| <i>flow_in_circuit(water)</i> | <i>flow_in_circuit(current)</i> | <i>flow_in_circuit(C)</i> |

explicitly in the metaphoric expression (M1). From background knowledge a rule is available stating that if the pressure caused by the pump *p1* in a water pipe system is different from 0, then water is flowing in the circuit (from high pressure to low pressure). This can be projected to the target side, inferring that due to the “pressure” of the battery *b1* (realized by a positive voltage), current is flowing in the electric circuit. Hence, we end up with the conclusion (5 in Table 2) that current is flowing in the electric circuit (provided there is a “pressure” source). The substitutions Θ_1 and Θ_2 can be summarized as follows:

$$\begin{aligned}
 \Theta_1/\Theta_2: \quad & A \longrightarrow ws1 / es1 \\
 & C \longrightarrow water / current \\
 & Source \longrightarrow pump / battery \\
 & S1 \longrightarrow p1 / b1 \\
 & Circuit \longrightarrow water_circuit / electric_circuit
 \end{aligned}$$

The following list sketches some reasons for the major claim of this subsection, namely that a large variety of human reasoning mechanisms can be modeled by analogies.

- Systems like HDTP allow the computation of analogical relations.
- Establishing analogical relations often requires the re-representation of a domain. HDTP achieves this by a theorem prover that is included in the system and allows the application of rules (cf. Row 4 in Table 2).
- Learning generalizations is a first step towards an induction on given input data [20]. In the example, a new conceptualization of the target domain is learned.
- The fact that analogies are at most psychologically preferred, but never true or false, allows the extension of the system to model uncertainty.
- Non-monotonicity can be considered as a special case of a re-conceptualization of a given a domain very similar to a new conceptualization of a domain by an analogical inference.

3.2 Rewriting Ontological Background Knowledge

In Section 2, two major problems that are connected to knowledge representation and HLI were mentioned: first, the profusion of knowledge and second, the fact that human beings are able to dynamically adapt background knowledge on-the-fly. We sketch some ideas in this subsection, primarily addressing the second problem: we propose a rewriting system that is constantly adapting the ontological knowledge base (memory) focusing on the resolution of inconsistencies. Although the framework was developed primarily for text technological applications, the underlying logical basis is rather weak, and obviously not all types of inconsistencies can be automatically resolved, we think that proposals in this direction are crucial for achieving HLI.

Ontological knowledge is usually formalized within a logical framework, most importantly in the framework of Description Logics (DL) [2]. In the past years, a variety of successful systems have been developed that make use of markup standards based on DL with varying degrees of expressiveness.⁵ However, the storage of ontological information within a logical framework has an undesirable side-effect: inconsistency problems can occur, because items of information may contradict each other, making the given ontology unsatisfiable and useless for reasoning purposes. Because HLI requires permanent updates of ontological knowledge, the problem of occurring inconsistencies becomes even more important. In this section, we sketch some ideas of how to address dynamic updates of ontologies leaving the problem of the profusion of knowledge aside.

Ontologies usually contain a terminological component and an assertion component. A description logic terminology consists of a set of terminological axioms defining concepts by formulas of the form $\forall x : C(x) \rightarrow D(x)$ or $\forall x : C(x) \leftrightarrow D(x)$, where C is a concept name and D is a concept description.⁶ The assertion component mentioned above contains information about the assignment of the particular individuals to concepts and relations from the terminology. Axioms are interpreted model theoretically by an interpretation function mapping concept descriptions to subsets of the domain. A model of an ontology is an interpretation satisfying all axioms. An ontology is inconsistent if it does not have a model.

There are several possibilities why inconsistencies can occur in ontologies. In [24], structural inconsistencies, usage-defined inconsistencies, and logical inconsistencies are distinguished. The last type of inconsistency – potentially caused by dynamic updates of the knowledge base – is of particular interest in our context and is addressed by an automatic rewriting device allowing constant learning and updates of the ontological knowledge base. One aspect of logical inconsistency problems concerns polysemy: If an ontology is updated automatically, then it is hardly possible to distinguish between word senses. Suppose, the concept *tree* is declared to be a subconcept both of *plant* and of *data structure* (where *plant* and *data structure* are disjoint concepts). Both of these two interpretations of *tree* are correct, but it is still necessary to describe two different

⁵ [2] provides an overview of different versions of description logics

⁶ Compare [2] for an exhaustive definition of description logics.

concepts in the ontology with different identifiers (e.g. *TreePlant*, *TreeStructure*). Otherwise, the terminology remains unsatisfiable.

Another important aspect of logical inconsistency problems concerns generalization mistakes. Consider the following classical example:

Example 1 *Assume the following axioms are given:*

$$\begin{aligned} \forall x : Bird(x) \rightarrow CanFly(x) & \quad \forall x : CanFly(x) \rightarrow CanMove(x) \\ \forall x : Canary(x) \rightarrow Bird(x) & \quad \forall x : Penguin(x) \rightarrow Bird(x) \wedge \neg CanFly(x) \end{aligned}$$

In Example 1, the statement “birds can fly” is too general. If an exception occurs (*penguin*), the ontology becomes unsatisfiable, since penguin is declared to be a bird, but it cannot fly. This type of inconsistency is the well-known problem of non-monotonicity, extensively discussed in the relevant AI literature.

The proposed approach – formally developed in [36], [37], and [38] – treats ontologies that are extended with additional axioms conflicting with the original knowledge base. Given a consistent ontology O (possibly empty) the procedure adds a new axiom A to O . If $O^+ = O \cup \{A\}$ is inconsistent, then the procedure tries to find a polysemy or an overgeneralization and repairs O^+ .

We will illustrate the regeneration of the overgeneralized concepts on the ontology in Example 2. Since the definition of the concept *Bird* is overgeneralized, it needs to be rewritten. We wish to retain as much information as possible in the ontology. The following solution is proposed:

Example 2 *Adapted ontology from Example 1:*

$$\begin{aligned} \forall x : Bird(x) \rightarrow CanMove(x) \\ \forall x : CanFly(x) \rightarrow CanMove(x) \\ \forall x : Canary(x) \rightarrow FlyingBird(x) \\ \forall x : Penguin(x) \rightarrow Bird(x) \wedge \neg CanFly(x) \\ \forall x : FlyingBird(x) \rightarrow Bird(x) \wedge CanFly(x) \end{aligned}$$

We want to keep in the definition of the concept *Bird* (subsuming the unsatisfiable concept *Penguin*) a maximum of information that does not conflict with the definition of *Penguin*. The conflicting information is moved to the definition of the new concept *Flying bird*, which is declared to subsume all former subconcepts of *Bird* (such as *Canary* for example) except *Penguin*.

Our algorithm (cf. [36], [37], and [38] for a detailed description) detects problematic axioms that cause a contradiction, defines the type of contradiction (polysemy or overgeneralization) and automatically repairs the terminology by rewriting parts of the axioms that are responsible for the contradiction. Detected polysemous concepts are renamed and overgeneralized concepts are split into more general and more specific ones. This approach is knowledge preserving in the sense that it keeps as many entailments implied by the original terminology as possible.

The sketched solution for a constant adaptation process of background knowledge is a first step towards a general theory of dynamification and adaptation of background knowledge. The framework has been developed primarily for text

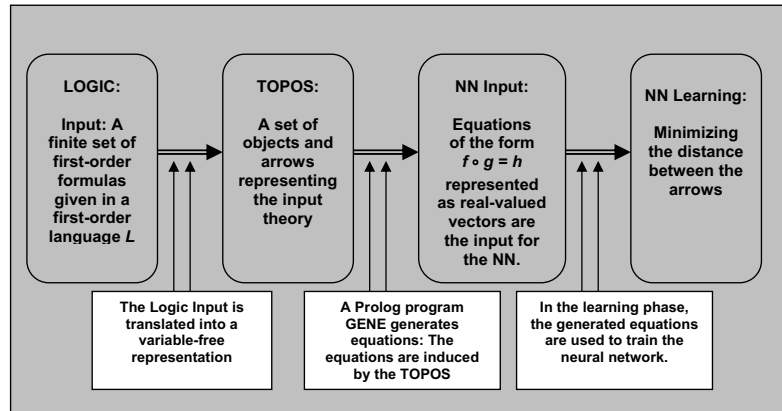


Fig. 2. The architecture for learning a first-order logical theory with neural networks.

technological applications. But the approach can straightforwardly be extended to a wider range of applications.⁷

3.3 Neuro-Symbolic Integration

In order to bridge the gap between symbolic and sub-symbolic approaches we sketch the theory presented in [19] and [22] based on the idea to translate first-order logical formulas into a variable-free representation in a topos [17]. A topos is a category theoretic structure consisting of objects Obj and arrows Ar having their domain and codomain in Obj . Certain construction principles allow to generate new arrows from old arrows. A fundamental theorem connects first-order logic and topos theory: a topos can be interpreted as a model of predicate logic [17]. The overall idea of learning symbolic theories with neural networks can be summarized as follows (compare also Figure 2):

- First, input data is given by a set of logical formulas (axioms and queries) relative to a given first-order logical language \mathcal{L} .
- Second, this set of formulas is translated into objects and arrows of a topos. The representation is variable-free and homogeneous, i.e. only objects and arrows are represented combined by the operation concatenation of arrows.
- Third, a PROLOG program generates equations in normal form $f \circ g = h$ identifying new arrows in the topos. This is possible because a topos allows several construction mechanisms.
- Last but not least, these equations are used as input for the training of a neural network. The network has a standard feedforward topology and learns by backpropagation: the network adapts the representations of arrows in such a way that the arrows representing “true” are approximating the arrow *true*.

⁷ The crucial algorithms for resolving overgeneralization, undergeneralization, and polymy problems, are implemented and prototypically tested in example domains [38].

The arrows *true* and *false* are the only hard-coded arrows, represented as $(1.0, 0.0, 0.0, \dots 0.0)$ and $(0.0, 1.0, 0.0, \dots 0.0)$ respectively.

Learning is possible, because the topos induces constructions that can be used for training the network. Clearly, infinitely many constructions are induced by the topos, but as it turns out a finite number is completely sufficient.

The details of the approach are rather complicated. We do not go into details here. The interested reader is referred to [19] and [22] for more information. The framework was tested with simple and also complex first-order logical theories.

4 The Integration of the Modules

4.1 A Hybrid Architecture for Higher Cognition

The three modules proposed in Section 3 – the neuro-symbolic integration module, the symbolic rewriting module, and the analogy reasoning module – attempt to learn a robust model of ontological background knowledge using a connectionist learning device, to dynamically rewrite ontologies on the symbolic level, and to perform various forms of reasoning, respectively. The task in this section is to integrate these modules into one single architecture called I-Cog (integrated cognition).

The integration of symbolic and sub-symbolic processes in a hybrid framework can be achieved, because the neuro-symbolic learning module is trained on symbolic data (i.e. on first-order logical expressions) and the fact that it learns a model of a logical theory. Although it is currently not possible to directly extract symbolic information from the neuro-symbolic learning device, an interaction and competition of the two modules can be implemented by querying both modules and evaluating their answers. Furthermore, both frameworks can interact with each other: queries of the rewriting module can be answered by the neuro-symbolic integration module. A similar remark holds for the integration of the analogy engine and the neuro-symbolic integration device.

Figure 3 depicts the overall architecture of the system. The architecture consists of the following modules (in the following paragraphs we will use the shortcuts *ORD* for the symbolic *Ontology Rewriting Device*, *NSLD* for the neural network-based *Neuro-Symbolic Learning Device*, *AE* for the *Analogy Engine*, and *CD* for the *Control Device*):

- The input may originate from various sources: input may be collected from resources based on structured data, unstructured data, or semi-structured data. The input needs to be available in an appropriate (subset) of a first-order language \mathcal{L} , in order to be in an appropriate format for the other modules. Therefore *ORD* generates appropriate logical formula from hypotheses.
- The input is used for feeding, updating, and training *ORD*.
- An important aspect is the interaction of *ORD* and *NSLD*: on the one hand, *ORD* trains *NSLD*, on the other hand *ORD* queries *NSLD*. Although *NSLD*

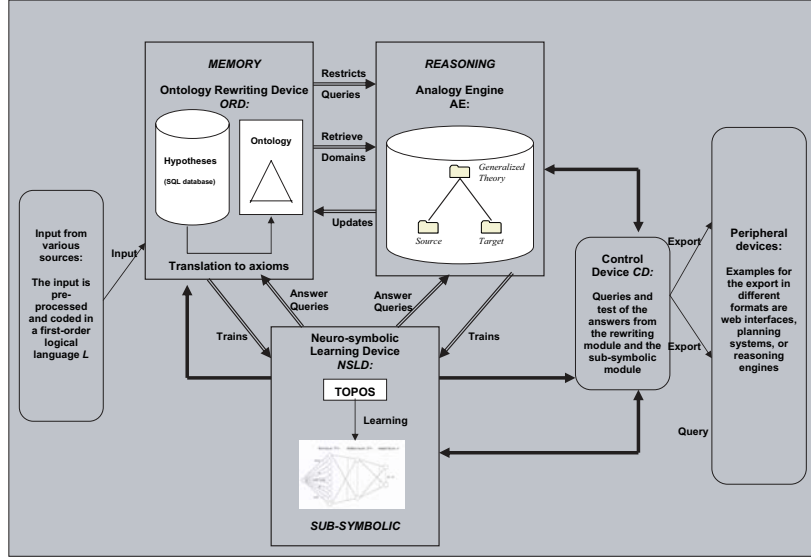


Fig. 3. The I-Cog architecture for an integration of the different modules. Whereas the modules *ORD* and *NSLD* are adapting new ontological axioms to an existing ontology, the analogy engine *AE* computes analogical relations based on background knowledge provided by the other two modules. The control device *CD* is intended to choose answers from all three modules.

can only give a tentative (or better approximate) answer in terms of a classification, this can improve the performance of the dynamic rewriting module in time-critical situations.

- With respect to the interaction of *AE* and *ORD*, ontological knowledge can naturally be used to constrain the computation of possible analogies [18]. Furthermore, newly generated analogies can be used to update and therefore rewrite background knowledge [23].
- Similarly to the relation between *ORD* and *NSLD*, *AE* is used to train *NSLD*, whereas query answering can be performed in the other direction.
- The control device *CD* of the two learning modules is intended to implement a competition of the feedback of the three modules with respect to queries. Feedback may be in accordance to each other or not. In the second case, the ranking of the corresponding hypotheses is decided by *CD* (see below).

We exemplify the interaction between *AE* and *ORD* in more detail (cf. [18], [23]): the establishment of an analogical relation of *AE*, if successful, provides a new conceptualization of the target domain. The example sketched in Subsection 3.1 results in a new conceptualization, where current is flowing in an electric circuit (triggered by a source). With respect to the ontological background knowledge *ORD* this means an update has to be performed, resulting in the introduction of a new (perhaps polysemous) concept, the update of a known

concept using new relational constraints (flowing in an electric circuit), or even the generation of a conflict in the knowledge base (which has to be resolved). Additionally, the generalized theory of the anti-unification process introduces a new concept specifying an abstract circuit, where an entity is flowing caused by a source. On the other hand, *ORD* can be used to restrict possible analogical relations computed by *AE*: Due to the fact that *AE* can generalize arbitrary concepts, fact, rules etc., ontological knowledge may be used to restrict certain undesirable generalizations. For example, for a physics domain containing concepts like *time-point*, *real number*, *force*, *electric charge*, *pressure* etc., it is undesirable to generalize *force* with *real number* or *pressure* with *time-point*. But it is desirable to generalize different types of *force*, or different types of *pressure*. Such restrictions can be implemented by specifying an upper-level ontology in *ORD* which blocks certain (logically possible) generalizations.

A crucial problem of the presented approach concerns the control device *CD*. This module needs to assess possible answers of the three main modules and needs to implement a competition process. The natural way to realize such a control mechanism is to learn the behavior of the systems, based on certain heuristics. We exemplify possible situations with respect to *ORD* and *NSLD*: with respect to underdetermined situations, *ORD* is not able to answer queries, simply because the reasoning engine will not be able to prove anything without sufficient knowledge. In contrast to *ORD*, *NSLD* will be able to give an answer in any case. In such cases the usage of *NSLD* is clearly preferred by the heuristic. On the other hand, if *ORD* is able to prove a particular fact, for example, that a certain subsumption relation holds between two concepts *A* and *B*, then this result should be tentatively preferred by *CD* in comparison to the output of *NSLD*. In cases where time-critical reactions are necessary and *ORD* is not able to compute an answer in time, the natural heuristic would be to use *NSLD* instead. Finally, it could happen that the answers of *ORD* and *NSLD* are contradicting each other. In this case, *CD* cannot base the decision on *a priori* heuristics. A natural solution to this problem is to implement a reinforcement learning mechanism on *CD* itself, namely the learning of preferred choices (dependent on the particular domain) of the knowledge modules involved.

4.2 The Added-Value of a Hybrid Approach

The added-value of the overall architecture (as depicted in Figure 3) can be summarized as follows:

- The architecture is robust due to the fact that the trained neural network can give answers to queries even though noise might be contained in the training data.
- Even in time-critical situations the proposed framework is able to react and to provide relevant information, because the neural network can answer a query immediately without any processing time, although the symbolic rewriting module may be busy with computation tasks. This can be achieved by a heuristic governing the behavior of *CD* in cases of time-critical situations.

- The architecture gives a first idea how an interaction between a symbolic level and a sub-symbolic level of computation can be achieved. The crucial issue is the fact that *NSLD* is able to learn from highly structured training data on a neural level.
- The architecture is cognitively more plausible than pure symbolic or sub-symbolic approaches. Although the hard problem of cognitive science, namely how a one-to-one translation from the symbolic level to the corresponding neural correlate and vice versa can be defined is not resolved, at least a particular direction of communication between such devices can be achieved.

Besides the mentioned advantages of such an architecture for automatically learning and adapting ontologies, covering many aspects of different reasoning paradigms, and providing a hybrid architecture, there is the chance to integrate various forms of cognitive capacities into one framework that are often considered to be incompatible. Perhaps this provides an idea of how to bridge the obvious gap between symbolic and subsymbolic processes, as well as the corresponding differences in computing paradigms and capacities. Models of conceptual theories (in our case of logical theories) can be coded on the neural level in a trained neural network. Additionally, this is complemented by a symbolic representation of the semantic knowledge of the environment, allowing classical (and non-classical) deductions and reasoning processes. In total, we think that the proposed hybrid architecture seems to be cognitively more plausible than isolated approaches that are purely based on one computational reasoning mechanism and representation paradigm.

5 Related Work

Some application domains for analogical reasoning were already mentioned in Section 3. Concerning underlying methods for modeling analogies algebraic [29], graph-based [10], and similarity-based approaches [15] can be found.

A collection of approaches that aims at resolving inconsistencies in knowledge representation is related to non-monotonicity. Some examples are extensions by default sets [25] or by belief-revision processes [12]. In [9], inductive logic programming techniques are proposed to resolve ontological inconsistencies. A family of approaches is based on tracing techniques for detecting a set of axioms that are responsible for particular contradictions in an ontology [3], [30].

With respect to the problem of representing symbolic data structures with neural means, we mention as examples sign propagation [32], dynamic localist representations [5], tensor product representations [42], or holographic reduced representations [39]. Furthermore, researchers tried to solve the so-called inference problem: whereas symbolic approaches allow one to draw inferences from given representations, there is no neural correlate to this capacity. An example to solve this problem is described in [27] in which a logical deduction operator is approximated by a neural network. Another approach is [26], where category

theoretic methods are used for neural constructions. In [8], tractable fragments of predicate logic are learned by connectionist networks.

Recently, some endeavor has been invested to approximate a solution to human-level intelligence. [7] proposes a so-called cognitive substrate in order to reduce higher cognition and the profusion of knowledge to a basis of low computational complexity. [13] propose to explain cognitive diversity of reasoning methods as a reduction to the well-known structure mapping theory [16]. Due to the combination of large knowledge bases, efficient retrieval, an analogy engine and learning modules, [13] is quite similar in spirit to the proposed architecture in this paper. Further approaches that resemble the integration idea presented here follow the tradition of cognitive architectures. Examples are the hybrid AMBR/DUAL model [31], which is modeling neuro-symbolic processing and analogical reasoning, the ICARUS architecture [33], which is focusing primarily on learning, or the NARS architecture [45], which is intended for integrating many different types of reasoning and representation formats.

6 Conclusions and Future Research

The paper proposes a hybrid architecture, based on analogical reasoning, an ontology rewriting device, and a module for neuro-symbolic integration, in order to model HLI. Although each module has been proven to be successfully applicable in theory and practice to the respective domains, many challenges remain open. Besides the fact that the overall architecture needs to be implemented and carefully evaluated, there are several theoretical questions that need to be addressed. One aspect concerns the control architecture, in particular, the question on which basis competing answers from the different modules are evaluated. Another issue concerns the interaction of the particular modules: for example, whereas the training of the *NSLD* module by *ORD* is more or less well-understood, the other direction, i.e. the input from *NSLD* to *ORD* is (at present) rather unclear. Consequently, it is currently only possible to query the neural network, because a direct extraction of symbolic knowledge from the trained network is an unsolved problem. Additionally, the problem of the profusion of knowledge and representation formalisms needs to be addressed. It may be a possibility to restrict ontological knowledge practically to hierarchical sortal restrictions that can be coded by relatively weak description logics, but in the long run, this is probably not sufficient. Last but not least, it would be desirable to add further devices to the system, e.g. planning systems and action formalisms.

The ultimate test scenario for the I-Cog architecture, as well as for HLI in general, would be a modified version of the Turing test: assume a robot operates an avatar in a virtual environment like “Second Life”, where real humans operate their avatars, too. If a human cannot decide whether an avatar is controlled by a robot or a human, the robot shows HLI and higher cognition in the sense of this paper. It is essential that such systems are built for humans to interact with and not for robots. It is obvious that no isolated AI tool like a theorem prover, a knowledge base, a neural network, a planning system etc. is able to control the

behavior of an avatar in a reasonable way in such a scenario. Although we do not claim that the presented architecture is sufficient to pass this type of “grand challenge”, we believe that only integrated cognitive architectures like I-Cog will have a chance at all.

Acknowledgment

This work has been partially supported by the German Research Foundation (DFG) through the grants KU 1949/2-1 and MO 386/3-4.

References

1. Anderson, J., Thompson, R.: Use of analogy in a production system architecture, in: Similarity and analogical reasoning, editors: Vosniadou, Ortony, Cambridge (1989) 267–297.
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D. Patel-Schneider (eds.), P.: *Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press (2003)
3. Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning*, **14**(1) (1995) 149–180.
4. Bader, S., Hitzler, P.: Dimensions of neural-symbolic integration - a structured survey. In S. Artemov, H. Barringer, A. S. d’Avila Garcez, L. C. Lamb, J. Woods (eds.): *We Will Show Them: Essays in Honour of Dov Gabbay* vol. 1, 167–194. International Federation for Computational Logic, College Publications, 2005
5. Barnden, J. A.: Neural net implementation of complex symbol processing in a mental model approach to syllogistic reasoning. In *Proceedings of the International Joint Conference on Artificial Intelligence (1989)* 568–573.
6. Barwise, J.: *On the Model Theory of Common Knowledge*, in J. Barwise: *The Situation in Logic*, Cambridge 1990.
7. Cassimatis, N.: A Cognitive Substrate for Achieving Human-Level Intelligence, *AI Magazine* **27**(2) (2006) 45–56.
8. D’Avila Garcez, A., Broda, K., Gabbay, D.: *Neural-Symbolic Learning Systems: Foundations and Applications*. Springer-Verlag (2002).
9. Fanizzi, N., Ferilli, S., Iannone, L., Palmisano, I., Semeraro, G.: Downward Refinement in the ALN Description Logic. *Proc. of the Fourth International Conference on Hybrid Intelligent Systems (HIS’04)* (2004) 68–73.
10. Falkenhainer, B., Forbus, K., Gentner, D.: The structure-mapping engine: Algorithm and example, *Artificial Intelligence* **41** (1989) 1–63.
11. Fikes, R., Nilson, N.: STRIPS: A new Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence* **2** (1971) 189–208.
12. Flouris, G., Plexousakis, D., Antoniou, G.: Updating DLs Using the AGM Theory: A Preliminary Study, in: *Description Logics, 2006*.
13. Forbus, K., Hinrichs, T.: Companion Cognitive Systems: A step towards human-level AI. *AI Magazine*, **27**(2) (2006) 83–95.
14. Gentner, D.: Why We’re So Smart, in: D. Gentner & S. Goldin-Meadow: *Language in mind: Advances in the study of language and thought*, Cambridge MA: MIT Press (2003) 195–235.
15. Gentner, D.: The mechanisms of analogical learning, in: S. Vosniadou & A. Ortony (editors): *Similarity and Analogical Reasoning*, New York, Cambridge University Press.
16. Gentner, D.: Structure Mapping: A theoretical framework for analogy, *Cognitive Science* **7**(2) (1983) 155–170.
17. Goldblatt, R.: *Topoi : The Categorical Analysis of Logic. Studies in Logic and the Foundations of Mathematics*, **98**, North-Holland, Amsterdam (1979).
18. Gust, H., Kühnberger, K.-U., Schmid, U.: *Ontological Aspects of Computing Analogies. Proceedings of the Sixth International Conference on Cognitive Modeling*, Mahwah, NJ: Lawrence Erlbaum, pp. 350–351.
19. Gust, H., Kühnberger, K.-U.: Learning Symbolic Inferences with Neural Networks. In: Bara, B., Barsalou, L., Bucciarelli, M. (eds): *CogSci 2005, XXVII Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum (2005) 875–880.
20. Gust, H., Kühnberger, K.-U.: Explaining Effective Learning by Analogical Reasoning, in: R. Sun, N. Miyake (eds.): *CogSci/ICCS 2006: 28th Annual Conference of the Cognitive Science Society in cooperation with the 5th International Conference of the Cognitive Science (in the Asian-Pacific region)*, Lawrence Erlbaum (2006) 1417–1422.
21. Gust, H., Kühnberger, K.-U., Schmid, U.: Metaphors and Heuristic-Driven Theory Projection (HDTP), *Theoretical Computer Science* **354** (2006) 98–117.

22. Gust, H., Kühnberger, K.-U. and Geibel, P. 2007. Learning Models of Predicate Logical Theories with Neural Networks based on Topos Theory. In P. Hitzler and B. Hammer (eds.): Perspectives of Neuro-Symbolic Integration, LNCS Springer (to appear).
23. Gust, H., Kühnberger, K.-U., Schmid, U.: Ontologies as a Cue for the Metaphorical Meaning of Technical Concepts, in A. Schalley, D. Khlentzos (Eds.): Mental States: Evolution, Function, Nature, John Benjamins Publishing Company, Amsterdam, Philadelphia (to appear).
24. Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., Sure, Y.: A framework for handling inconsistency in changing ontologies. *Proc. of the Fourth International Semantic Web Conference*, LNCS, Springer (2005).
25. Heymans, S., Vermeir, D.: A Defeasible Ontology Language, In Meersman, R. et al. (eds): Confederated International Conferences: CoopIS, DOA, and ODBASE 2002 Springer 1033-1046.
26. Healy, M., Caudell, T.: Neural Networks, Knowledge and Cognition: A Mathematical Semantic Model Based upon Category Theory. University of New Mexico, (2004), EECE-TR-04-020.
27. Hitzler, P., Hölldobler, S., Seda, A.: Logic programs and connectionist networks. *Journal of Applied Logic*, **2(3)** (2004) 245-272.
28. Hofstadter, D. and The Fluid Analogies Research Group: Fluid concepts and creative analogies. New York: Basic Books, 1995.
29. Indurkha, B.: Metaphor and Cognition, Dordrecht, the Netherlands, Kluver, 1992.
30. Kalyanpur, A.: Debugging and Repair of OWL Ontologies. Ph.D. Dissertation, 2006.
31. Kokinov, B., Petrov, A.: Integrating Memory and Reasoning in Analogy-Making: The AMBR Model, in D. Gentner, K. Holyoak, B. Kokinov (eds.): The Analogical Mind. Perspectives from Cognitive Science, Cambridge Mass. 2001.
32. Lange, T., Dyer, M. G.: High-level inferencing in a connectionist network. Technical report UCLA-AI-89-12, 1989.
33. Langley, P.: Cognitive Architectures and General Intelligent Systems, *AI Magazine*, **27(2)** (2006) 33-44.
34. McCarthy, J.: Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I. *Communications of the ACM* **3(4)** (1960) 184-195.
35. McCulloch, W., Pitts, W.: A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, **5** (1943) 115-133.
36. Ovchinnikova, E., Kühnberger, K.-U.: Adaptive $\mathcal{AL}\mathcal{E}$ -TBox for Extending Terminological Knowledge. In A. Sattar, B. H. Kang (eds.): *Proceedings of the 19th ACS Australian Joint Conference on Artificial Intelligence*, LNAI 4304, Springer, pp. 1111-1115, 2006.
37. Ovchinnikova, E., Kühnberger, K.-U. 2006b. Aspects of Automatic Ontology Extension: Adapting and Regeneralizing Dynamic Updates, in M. Orgun & T. Meyer (eds.): *Advances in Ontologies 2006, Conferences in Research and Practice in Information Technology*, **72** (2006) 52-60.
38. Ovchinnikova, E., Wandmacher, T. and Kühnberger, K.-U. 2007. Solving Terminological Inconsistency Problems in Ontology Design. *International Journal of Interoperability in Business Information Systems*, **2(1)** (2007) 65-80.
39. Plate, T.: Distributed Representations and Nested Compositional Structure. PhD thesis (1994), University of Toronto.
40. Plotkin, G.: A note of inductive generalization, *Machine Intelligence* **5** (1970) 153-163.
41. Schwering, A., Krumnack, U., Kühnberger, K.-U., Gust, H.: Using Gestalt Principles to Compute Analogies of Geometric Figures, in: Proceedings of the 29th Annual Conference of the Cognitive Science Society (2007) (to appear).
42. Smolenski, P.: Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, **46(1-2)** (1996) 159-216.
43. Staab, S., Studer, R. (eds.): Handbook of Ontologies. Springer 2004.
44. Turing, A.: Computing Machinery and Intelligence. *Mind* **59** (1950) 433-460.
45. Wang, P.: Rigid Flexibility: The Logic of Intelligence Springer 2006.