I-Cog: A Computational Framework for Integrated Cognition of Higher Cognitive Abilities

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. There are several challenges for AI models of higher cognitive abilities like the profusion of knowledge, different forms of reasoning, the gap between neuro-inspired approaches and conceptual representations, the problem of inconsistent data, and the manifold of computational paradigms. The I-Cog architecture – proposed as a step towards a solution for these problems – consists of a reasoning device based on analogical reasoning, a rewriting mechanism operating on the knowledge base, and a neuro-symbolic interface for robust learning from noisy data. I-Cog is intended as a framework for human-level intelligence (HLI).

1 Introduction

Historically, artificial intelligence has (more or less) been strongly committed to interdisciplinary research and the modeling of higher cognition. 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 or intelligent programs support our daily life in various respects, for example, when driving a car, flying a plane, or searching the web for information. Despite these apparent examples of success, there are also severe problems: there is not even an idea of how human-level intelligence (HLI) in the large can be achieved, taking into account the various forms of human capabilities, for example, concerning reasoning, problem solving, learning, adapting, acting etc. Three classes of problems are discussed in this paper: First, the problem of constant updates of knowledge, second, the variety of types of reasoning of human cognition, and third, the gap between neuro-inspired learning approaches and symbolic representations.

We think that these challenges are at the heart of achieving HLI, 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.[1]

[1] This claim clearly does not mean that other challenges for modeling cognition are simple or somehow straightforward to solve. Obviously open problems in computer vision or the modeling of autonomous agents and motor control, are also hard problems. But they concern lower cognitive abilities and are not aspects of HLI.

We propose an integration architecture as a model for higher cognitive abilities by the resolution of the three mentioned sub-problems.

This paper has the following structure: Section 2 sketches some problems in modeling a variety of higher cognitive abilities. Section 3 presents the I-Cog architecture consisting of an analogy engine (AE), an ontology rewriting device (ORD), and a neuro-symbolic learning device ($NSLD$). These modules interact in a non-trivial way as 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 understood as encoding entities in the environment using symbols. Although a logical representation is generally accepted as an appropriate framework for this task, there is a non-trivial challenge for the logical approach: Whereas background knowledge is usually considered to be static, human agents constantly update, modify, and learn new conceptualizations, and they can overwrite existing knowledge easily without being threatened by inconsistency problems. Dynamic rewriting of knowledge is hard to model in AI and therefore a challenge, in order to reach a theory of HLI.

2.2 Reasoning

There is a manifold of human reasoning abilities. Humans perform not only deductions, inductions, and abductions, but are also able to perform analogical reasoning steps, non-monotonic inferences, and frequency-based inferences. Additionally, human agents can reason with vague and uncertain knowledge. As a natural consequence of this variety of reasoning types AI research 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 proven to be a robust tool for learning from noisy data, pattern recognition, and handling vague knowledge – classical domains with which symbolic theories usually have their problems. A potential solution for achieving HLI would require an integration of both approaches in one architecture.

3 Modules of the I-Cog Architecture

3.1 Analogy Engine (AE)

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 [7]. Examples for application domains, where analogical reasoning was successfully applied, are string domains [18], geometric figures [27], problem solving [1], naive physics [5], or metaphoric expressions [13]. Furthermore analogies are a source of creativity [19] and a possibility to learn (inductively) from sparse data [12]. Vagueness plays a crucial role in every account of analogical reasoning. Deductions and abductions are implicitly modeled in several systems [6].

A rather expressive theory for computing analogies in a variety of domains is provided by heuristic-driven theory projection (HDTP) [13]. 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 [26]. Here are some key elements of HDTP:

- Two formulas $p(a)$ and $p(b)$ can be anti-unified by $p(X)$, where the variable X is substituted by $\Theta_1(X) = a$ and $\Theta_2(X) = b$ (first-order anti-unification).
- Two formulas $p_1(a)$ and $p_2(a)$ can be anti-unified by $P(a)$, where the predicate variable P is substituted by $\Theta_1(P) = p_1$ and $\Theta_2(P) = p_2$ (second-order anti-unification).
- A theorem prover allows the re-representation of formulas.
- Whole theories can be generalized, not only single terms or formulas.

We exemplify analogy making in HDTP using a simple example.[2] Consider the following analogy induced by a metaphoric expression:

Current is the water in the electric circuit.

The metaphor establishes a new conceptualization of an electric circuit (target domain) by the association of *electric circuit* and *water-pipe system*, *current* and *water*, *pump* and *battery* etc. dependent on the available background knowledge. HDTP computes these associations together with a generalized theory and substitutions governed by heuristics (e.g. the complexity of relevant substitutions). Notice that by establishing a new conceptualization of the target domain, the systems learns new concepts, needs to modify the definitions and relational restrictions of hitherto concepts, or end up in an inconsistency.

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.

[2] A formalization of the following analogy, as well as the specification of the underlying algorithm can be found in [12].

– Establishing analogical relations requires often the re-reprensentation of a domain. HDTP achieves this by using a theorem prover included in the system, i.e. the systems allows logical deductions.
– Learning generalizations is a first step towards an induction on given input data [12]. In the example, a new conceptualization of the target 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 re-conceptualizing a given domain very similar to a new conceptualization induced by an analogy.

3.2 Ontology Rewriting Device (ORD)

The fact that human beings are able to dynamically adapt background knowledge on-the-fly was mentioned as a major challenge for HLI (cf. Section 2). We sketch some ideas (based on [23] and [24]) of a rewriting system that is constantly adapting the ontological knowledge base (memory) focusing on the resolution of inconsistencies. Although the framework was originally developed for text technological applications, the underlying logical basis is rather weak, and 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 often in the framework of Description Logics (DL) [2]. However, the storage of ontological information within a logical framework has an undesirable side-effect: updates can cause inconsistency problems, because items of information may be contradicting, making the given ontology unsatisfiable and useless for reasoning purposes.

Ontologies usually contain a terminological and an assertion component. A DL terminology consists of a set of terminological axioms defining concepts by formulas of the form $\forall x : C(x) \rightarrow D(x)$, where C is a concept name and D is a concept description, i.e. a logical formula.[3] The assertion component mentioned above contains information about the assignment of the particular individuals to concepts and relations from the terminology. Axioms are interpreted 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 how inconsistencies can occur in ontologies [15]. In particular, logical inconsistencies – potentially caused by dynamic updates of the knowledge base – are of interest in our context and are 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

[3] Compare [2] for a exhaustive definition of description logics.

is still necessary to describe in the ontology two different concepts with different identifiers (e.g. *TreePlant*, *TreeStructure*). Another important aspect of logical inconsistency concerns generalization mistakes.

Example 1 *Assume the following axioms are given:*
$\forall x : Bird(x) \rightarrow CanFly(x)$ $\quad$ $\forall x : CanFly(x) \rightarrow CanMove(x)$
$\forall x : Canary(x) \rightarrow Bird(x)$ $\quad$ $\forall x : Penguin(x) \rightarrow Bird(x) \wedge \neg CanFly(x)$

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. We will illustrate the regeneralization of the overgeneralized concept *Bird* in Example 2.

Example 2 *Adapted ontology from Example 1:*
$\forall x : Bird(x) \rightarrow CanMove(x)$ $\quad$ $\forall x : CanFly(x) \rightarrow CanMove(x)$
$\forall x : Canary(x) \rightarrow FlyingBird(x)$ $\quad$ $\forall x : Penguin(x) \rightarrow Bird(x) \wedge \neg CanFly(x)$
$\forall x : FlyingBird(x) \rightarrow Bird(x) \wedge CanFly(x)$

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*.

The algorithms developed in [23] and [24] are intended to detect problematic axioms causing a contradiction, to define the type of the contradiction (polysemy or overgeneralization) and to automatically repair 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. Such solutions for a constant adaptation process of background knowledge are a first step towards a general theory of dynamification and adaptation of background knowledge. Although the framework has been tested in the area of text technology, it can be straightforwardly extended to a wider range of applications.

3.3 Neuro-Symbolic Learning Device

In order to bridge the gap between symbolic and sub-symbolic approaches, we sketch the theory presented in [11] based on the idea of translating first-order logical formulas into a variable-free representation in a topos [9]. 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 [9]. The overall idea of learning symbolic theories with neural networks can be summarized as follows (compare also Figure 1):

- First, input data is given by a set of logical formulas in a language $\mathcal{L}$.

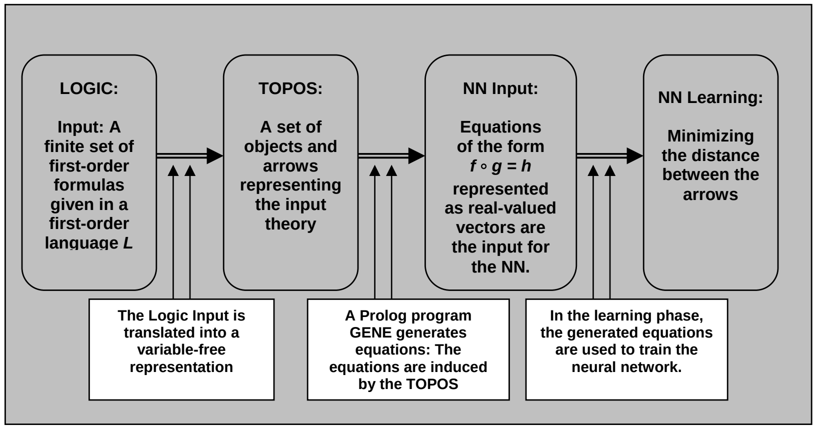

Fig. 1. The architecture for learning a first-order logical theory with neural networks.

- 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.
- 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 that correspond to "true logical expressions" are approximating the arrow *true*. The arrows *true* and *false* are the only hard-coded arrows.

Learning is possible, because the topos induces constructions that can be used for training. Although infinitely many constructions are induced by the topos, it turns out that a finite number is completely sufficient. We cannot go into details this approach. The interested reader is referred to [11] for more information. The framework has been tested with simple and also complex FOL theories.

4 The Integration of the Modules

4.1 A Hybrid Architecture for Higher Cognition

The three modules proposed in Section 3 – *NSLD*, *ORD*, and *AE* – 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 integration of these modules can be achieved as follows: symbolic and sub-symbolic processes can be integrated, because *NSLD* is trained on symbolic data (i.e. on first-order logical expressions) and it learns a model of a logical theory. Although it is currently

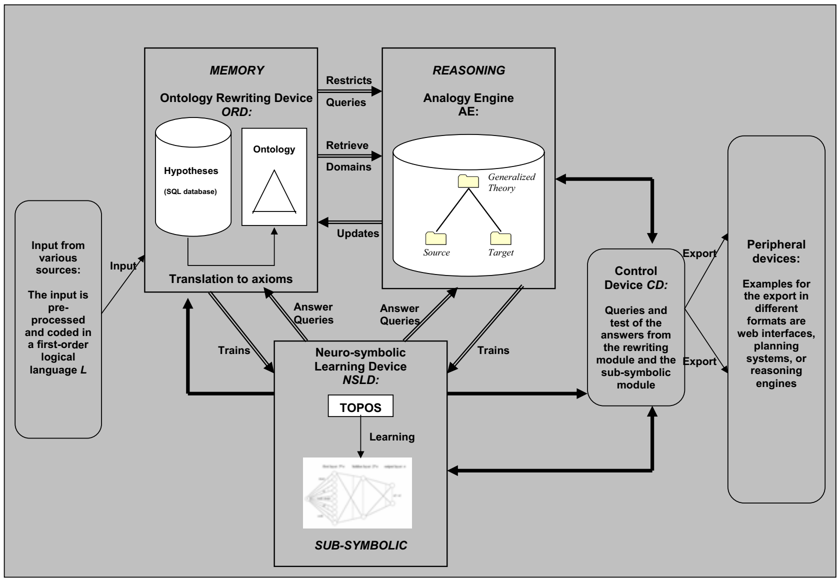

Fig. 2. The overall architecture for an integration of the different modules. Whereas the modules *ORD* and *NSLD* are adapting new ontological axioms to an existing ontology, the analogy engine *AE* computes analogical relations based on background knowledge provided by the other two modules. The control device *CD* is intended to choose answers from all three modules.

not possible to extract directly symbolic information from *NSLD*, a competition of *ORD* and *NSLD* can be implemented by querying both and evaluating their answers. Furthermore both modules can directly interact due to the fact that input from *ORD* can be used for training and querying. A similar remark holds for the integration of *AE* and *NSLD*. Figure 2 depicts the overall architecture of the system.

- The input may originate from various sources, e.g. 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.
- 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* can only give an approximate answer in terms of a classification, this can improve the performance of *ORD* 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 [10]. Furthermore newly generated analogies can be used to update and therefore rewrite background knowledge [14].
- 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: the establishment of an analogical relation, if successful, provides a new conceptualization of the target domain. The metaphor 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 ontology rewriting device *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, ontological knowledge may be used to restrict certain undesirable generalizations. For example, for a physics domain containing concepts like *time-point, real number, force, 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* blocking certain (logically possible) generalizations.

We continue with some remarks concerning *CD*. This module evaluates 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*: in underdetermined situations, *ORD* is not able to answer queries, simply because *ORD* 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, in which 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 involved knowledge modules.

4.2 The Added-Value of a Hybrid Approach

The added-value of the overall architecture (as depicted in Figure 2) can be summarized as follows:

- The architecture is robust, because 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.
- 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 that *NSLD* is able to learn from highly structured training data.
- 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 reached yet, at least one direction of such a translation can be modeled.

Besides the mentioned advantages of such an architecture for automatically learning and adapting lexical ontologies, there is an important cognitive aspect that should be mentioned: phenomenologically the dualism between a conceptual level of cognition (*mind perspective*) and a neural level of cognition (*brain perspective*) is hardly questionable. From the *mind perspective*, cognition rests fundamentally on conceptual knowledge, i.e. on complex data structures, whereas from the *brain perspective* knowledge is not visible, but somehow hidden in the weights of the network or in specialized single neurons (or families of neurons) etc. To put it differently, although humans tend to use language as a prototypical tool for coding conceptual knowledge, it is not clear what the corresponding neural correlate should be, although this correlate is the only objectively measurable and directly accessible activity. Perhaps one can circumvent this hard problem of cognitive science by rooting higher cognitive abilities in models that humans use in order to think, to reason, or to communicate, instead of manipulating symbolic systems. If this is true, then this model of a conceptual theory (in our case of a logical first-order theory) can be coded on the neural level in a trained neural network. Additionally, this is complemented by a symbolic representation of semantic knowledge about the environment, allowing 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. With respect to the underlying methods for analogy making, algebraic accounts [19], graph-based approaches [5], and similarity-based approaches [8] are to be mentioned.

A collection of approaches attempting to resolve inconsistencies in knowledge representation is related to classical methods of non-monotonic reasoning in logical systems. An example is the extension by default sets [16]. In [4], 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 [20].

With respect to neuro-symbolic integration, we mention as examples sign propagation [22], tensor product representations [28], or holographic reduced representations [25]. Furthermore, researchers tried to model inferences with neural networks. An example to solve this problem is described in [17] in which a logical deduction operator is approximated by a neural network.

Recently, some endeavor has been invested to address the problem of achieving HLI. [3] proposes a so-called cognitive substrate in order to reduce higher cognition to a basis of low computational complexity. [6] propose to explain cognitive diversity as a reduction to the well-known structure-mapping theory. Another research tradition for higher cognition is the development of cognitive architectures like the AMBR/DUAL model [21] or the NARS architecture [30].

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.

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. Cassimatis, N.: A Cognitive Substrate for Achieving Human-Level Intelligence, *AI Magazine* **27(2)** (2006) 45–56
4. 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
5. Falkenhainer, B., Forbus, K., Gentner, D.: The structure-mapping engine: Algorithm and example, Artificial Intelligence **41** (1989) 1–63
6. Forbus, K., Hinrichs, T.: Companion Cognitive Systems: A step towards human-level AI. *AI Magazine*, **27(2)** (2006) 83-95
7. 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
8. Gentner, D.: The mechanisms of analogical learning, in: S. Vosniadou & A. Ortony (editors): *Similarity and Analogical Reasoning*, New York, Cambridge University Press (1989) 199–241
9. Goldblatt, R.: Topoi : The Categorial Analysis of Logic. Studies in Logic and the Foundations of Mathematics, **98**, North-Holland, Amsterdam (1979)
10. Gust, H., Kühnberger, K.-U., Schmid, U.: Ontological Aspects of Computing Analogies. Proceedings of the 6th International Conference on Cognitive Modeling, Mahwah, NJ: Lawrence Erlbaum (2004) 350–351
11. 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
12. 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, Lawrence Erlbaum (2006) 1417–1422
13. Gust, H., Kühnberger, K.-U., Schmid, U.: Metaphors and Heuristic-Driven Theory Projection (HDTP), Theoretical Computer Science **354** (2006) 98–117
14. 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 191–212 (to appear)
15. Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., Sure, Y.: A framework for handling inconsistency in changing ontologies. *Proc.of the Fourth Internation Semantic Web Conference*, LNCS, Springer (2005)
16. 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
17. Hitzler, P., Hölldobler, S., Seda, A.: Logic programs and connectionist networks. Journal of Applied Logic, **2(3)** (2004) 245-272
18. Hofstadter, D. and The Fluid Analogies Research Group: Fluid concepts and creative analogies. New York: Basic Books (1995)
19. Indurkhya, B.: Metaphor and Cognition, Dordrecht, the Netherlands, Kluver (1992)
20. Kalyanpur, A.: Debugging and Repair of OWL Ontologies. Ph.D. Dissertation, (2006)
21. 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)
22. Lange, T., Dyer, M. G.: High-level inferencing in a connectionist network. Technical report UCLA-AI-89-12 (1989)
23. Ovchinnikova, E., Kühnberger, K.-U.: Adaptive $\mathcal{ALE}$-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 (2006) 1111–1115.
24. 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.
25. Plate, T.: Distributed Representations and Nested Compositional Structure. PhD thesis, University of Toronto (1994)
26. Plotkin, G.: A note of inductive generalization, *Machine Intelligence* **5** (1970) 153-163
27. 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)
28. Smolenski, P.: Tensor product variable binding and the representation of symbolic structures in connectionist systems. Artificial Intelligence,**46(1–2)** (1996) 159–216
29. Staab, S., Studer, R. (eds.): Handbook of Ontologies. Springer (2004)
30. Wang, P.: Rigid Flexibility: The Logic of Intelligence Springer (2006)