FLEXIBILITY VERSUS EFFICIENCY: THE DUAL ANSWER

BOICHO KOKINOV

Institute of Mathematics, Bulgarian Academy of Sciences
Bl.8, Acad. G. Bonchev Street, Sofia 1113, BULGARIA
E-mail: kokinov@bgearn.bitnet

ABSTRACT

A fundamental question that always arises in cognitive modeling is how to combine two contradictory constraints: the model should be as flexible as possible (to reflect human flexibility) and at the same time it should be maximally efficient. The decision proposed in this paper is the following. The space to be searched has to be restricted for computational reasons, but this restriction should not be done in advance. It should be dynamic and should reflect the particular situation encountered, i.e. it should reflect the dynamically evolving context. In this way making the computations context-sensitive they will be both flexible and efficient.

A multi-agent cognitive architecture is proposed consisting of hybrid (symbolic/connectionist) micro-agents. The behavior of the system at the macro level emerges from the collective behavior of the micro-agents. The symbolic computation performed by the system emerges from the symbolic micro computations performed by the agents, while their “power” or “rate” depends on their connectionist activation levels. The activation distribution over the agents reflects the particular context. This architecture allows for a greater flexibility of the cognitive system while at the same time decreasing the complexity of computation.

1. Introduction

A fundamental question that always arises in cognitive modeling is how to combine two contradictory constraints: the model should be as flexible as possible (to reflect human flexibility) and at the same time it should be maximally efficient. Each model is a trade-off between these two requirements.

The problem is that the greater flexibility makes the space to be explored larger therefore reduces the efficiency. Moreover, in most cases it makes the computation intractable. This problem is usually solved by restricting the problem space in advance in a way facilitating the search for the appropriate knowledge for completing the particular task or set of tasks the system is designed to accomplish. However, this restricts the system's flexibility: it makes the use of the model in other tasks (where the solution is outside the restricted problem space) impossible. In technical applications this is not a problem, because the most important feature there is efficiency, moreover, different systems could be designed for different cases. However, in cognitive modeling a single model of the cognitive process should be able to account for people's behavior under various circumstances.

Let's consider the following example. In an analogical reasoning task, relations from two subject domains should be put into correspondence. The set of possible pairings of relations is enormous large and it is impossible to search it in real time. That is why every model (proposed so far) restricts the search in a particular way. Thus Gentner[5,6] pairs only identical relations (this is a quite extreme restriction, but even in that case a large space of possible pairings remains, especially when the same relations are used several times in the problem description). This restriction, however, makes even the simple analogy between the following cases impossible: *on(vase, desk)* and *over(tablecloth, table)* - here, *on* and *over* are not identical, but are semantically similar. Holyoak and Thagard[10, 15] solve the same problem allowing two relations which have an immediate common super-class to be paired - so, in the above example an immediate common super-class of *on* and *over* might be *above*. However, even in this case a severe restriction is made: only relations with *immediate* common super-class are paired, e.g. *on(vase, desk)* and *supports(desk, computer)* cannot be put into correspondence using this mechanism, although humans will do it because both relations have a common super-class at a more abstract level. The restriction to *immediate* super-classes is made only for computational reasons, otherwise an exhaustive search is needed.

The decision proposed in this paper is the following. The space to be searched has to be restricted for computational reasons, but this restriction should not be done in advance. It should be dynamic and should reflect the particular situation encountered, i.e. it should reflect the dynamically evolving context. In this way making the computations context-sensitive they will be both flexible and efficient.

2. DUAL – A Context-Sensitive Multi-Agent Cognitive Architecture

DUAL is a multi-agent architecture, i.e. it consists of a large number of micro-agents, each of which represents some specific declarative and / or procedural knowledge. The agents are relatively simple - they do not have their own goals or reasoning mechanisms. They are some kind of specialized computational devices.

Each micro-agent is hybrid – it has a symbolic and a connectionist part called L-Brain and R-Brain, respectively. The symbolic part represents a piece of knowledge, while the connectionist part - its relevance to the current context. If, for example, the L-Brain of an agent represents the fact that “the Bulgarian state was established in 681” then the activation level established by its R-Brain represents its relevance to the current context and determines its accessibility at that moment. If, on the other hand, the L-Brain of an agent represents the procedural knowledge about a particular symbolic operation, such as marker passing or structure mapping, then the activation level established by its R-Brain determines whether this operation is allowed at that moment and what is its priority or rate. The former case allows only a small fraction of the knowledge base (KB) represented by the L-Brains of the agents to be searched at every particular moment and the latter case allows particular operations to be supported or suppressed depending on the context. This allows for a greater flexibility of the cognitive system while at the same time decreasing the complexity of computation.

All the micro-agents may act in parallel, but at each particular moment only the active agents are working together and competing or cooperating with each other. Moreover, every agent acts at its own rate (in an asynchronous manner) depending on its activation level. In this way the behavior of the system at the macro level emerges from the collective behavior of the micro-agents, i.e. the symbolic computation performed by the system emerges from the symbolic micro-computations performed by the micro-agents, where the particular set of agents taking part in the computation process as well as their competitive power depends on their activations which reflect the particular context.

Compared to other hybrid systems[1, 2, 4, 7, 8, 16], DUAL is hybrid at the micro level rather than at the macro level. That is, instead of having separate modules implemented according to the symbolic and connectionist paradigms each modeling a particular cognitive process (or a particular stage of it), it consists of a large set of small hybrid agents contributing to all cognitive processes. In this way both symbolic and connectionist aspects are considered important for every aspect of human cognition.

Compared to multi-agent systems such as the Blackboard architectures there are a number of differences:

- The agents themselves are part of the blackboard, i.e. there is no separation between data structures being placed on the blackboard and agents acting

on them and representing particular procedures. Instead, agents might by treated as data structures from other agents, while at the same time actively processing some other agents.
- The working memory is not a global base accessible for all agents, but instead every agent “sees” only a small fraction of the blackboard - the specific part connected with its functioning, i.e. there is only *local processing*: the agents are connected with each other (some of the links are permanent, others are dynamically created and removed) and every agent exchanges information only with its neighbors.
- All agents act in parallel, each of them at its own rate proportional to its activation level. This makes the computation context-dependent and allows competition between agents.

Another cognitive architecture close to DUAL is the one proposed by Hofstadter and his group[3, 9]. It is a multi-agent architecture with Codelets acting in parallel as agents over the blackboard (the Workspace consisting of data structures) and influenced by the activation levels of the nodes in a semantic network (called Slipnet). In this architecture, however, procedures and data structures are separated (in Coderack and Slipnet and Workspace, respectively) and different mechanisms (“urgencies”, activation levels, and strengths) are used for their control. Agents are able to observe the whole Workspace and in this way they act as global processors. The parallel work of the Codelets is simulated by a stochastically constructed sequence of its running with probabilities corresponding to their “urgencies”.

3. Internal Structure of DUAL Micro-Agents

Each DUAL agent consists of two highly interrelated processors – the L-Brain and the R-Brain. The L-Brain is designed according to the symbolic paradigm, whereas the R-Brain – according to the connectionist paradigm. The R-Brain of an agent acts as an power supply for the corresponding L-Brain. Thus, although all the L-Brains can potentially work in parallel, in each particular moment only a small fraction of them has the necessary energy supplied by the corresponding R-Brains for actual working. On the other hand, all the R-Brains are continuously working in parallel calculating the activation levels of the agents, i.e. their power supply.

3.1. L-Brains: The Symbolic Part of the Micro-Agents

The L-Brain of an agent represents a combination of declarative and procedural knowledge about a particular object / situation, a generic concept, or a given action. For this reason a frame-like representation scheme is used.

An example of a frame in the DUAL architecture is presented in Figure 1. The slot fillers are simply pointers to other frames or their slots and no special language is used for their description. The links between the agents correspond to these pointers and represent various semantic links. This leads to a highly distributed representation of the knowledge and keeps the symbolic processors quite simple.

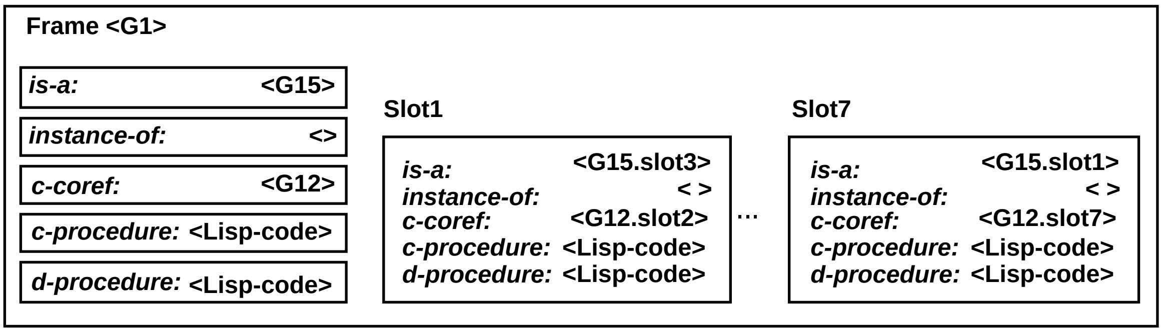


Figure 1.
An example of a frame in the DUAL architecture.

The *is-a* and *instance-of* links define the concept as a specialization of or as a particular instance of a class. Every two agents linked to each other by a *c-coref* link (short for “conceptual coreference”) represent one and the same entity in the world possibly from two different points of view. This allows for multiple descriptions of one and the same object, concept, situation, etc. The *c-procedure* and *d-procedure* represent the procedural knowledge and correspond to “procedures to be called” and “demons”, respectively.

The L-Brains are specialized symbolic processors (Figure 2). They have permanent memory for all outgoing links (pointers to other frames and labels of the semantic links) as well as temporary memory for markers (structures containing pointers to other, possibly non-neighboring nodes) and other local data. All L-Brains have the ability to receive and send markers and to differentiate links with different labels (e.g. to pass the markers only along links with specific label). In addition, the L-Brains of some agents are able to perform specific hard-wired programs (c- and d-procedures) corresponding to some possible actions of the cognitive system. Some examples of such specialized agents are the agents able to initiate a marker-passing process, the agents able to construct new agents (node constructors), the agents able to initiate a mapping between two descriptions, the agents able to establish local correspondence between two structures, etc.

Specialized symbolic processes: (c-procedures and d-procedures)	Common symbolic processes: (local marker-passing ability)
Temporary memory for its local data	Temporary memory for markers
Permanent memory for representing semantic links to other frames	

Figure 2.
The L-Brains of the agents.

3.2. R-Brains: The Connectionist Part of the Micro-Agents

The R-Brains of the micro-agents in DUAL represent context and relevance. Context is represented by the relevance factors of each agent to the current situation. The degree of connectivity of each element with all other elements of that situation is chosen as a particular measure of relevance and is represented by the activation level of the corresponding agent. Thus the activation level of the agent computed by its R-Brain within the connectionist aspect represents the relevance of the description corresponding to the agent within the symbolic aspect.

The links between the agents within the connectionist aspect have no labels and reflect only the strength of the associative relations between them, i.e. how often the two agents appear in the same context. All the links which have some semantic interpretation within the symbolic aspect are used also by the connectionist aspect ignoring their specific semantic interpretation. In addition the *a-link*s (short for "associative links") represent arbitrary associations which are ignored by the symbolic aspect. They are not recognized by the symbolic processors and are used only by the connectionist aspect of the architecture.

Perception and system’s goals are sources of activation. That is the R-Brains of agents corresponding to entities being perceived at the moment as well as of agents corresponding to the current goals of the cognitive system continuously emit activity.

The R-Brains are connectionist processors (Figure 3) calculating the activation values and outputs of the nodes on the basis of their input values and current activity running. They work in parallel in a discrete synchronous manner in order to simulate the continuous process of spreading activation. They have permanent memory for all outgoing links (pointers and weights) and temporary memory for their net input, activation value, previous activation level, and output. They have hard-wired programs calculating the activation and output functions as well as programs for weights learning.

<table>
<tr><th colspan="5">Connectionist processes</th></tr>
<tr><td colspan="2">activation</td><td colspan="2">output</td><td>learning</td></tr>
<tr><td>current
net input</td><td>current
activation
level</td><td>previous
activation
level</td><td>current
output</td><td>connection
weights</td></tr>
<tr><td colspan="4">Temporary memory</td><td>Permanent
memory</td></tr>
</table>

Figure 3.
The R-Brains of the agents.

4. An Example of a Context-Sensitive Computation Performed by DUAL

Let us consider an example of performing context-sensitive computations which demonstrates the use of DUAL in a fragment of an analogy-making task. Let's have two simple propositions: *The pot is on the plate*, and *The immersion heater is immersed in the water*. A correspondence between *on* and *is immersed in* is being searched. This problem cannot be solved using the techniques of Gentner[5, 6] or Holyoak and Thagard[10, 15] as this two relations are neither identical nor have an immediate common superclass. So, to solve the problem we have to allow searching for common superclasses at any level. However, this leads to enormously enlarging the search space which will made the computations untractable. In DUAL this search is performed by a marker-passing mechanism, i.e. by a highly parallel process, but this is still not enough to reduce the search process to reasonable complexity. However, the marker-passing process is performed by the micro-agents and that is why only active agents can take part in the process. This reduces enormously the search space restricting it to the active part of the memory – the Working Memory (WM). The search within the WM is already a quite effective task. In this way DUAL combines the flexibility of being able to find common superclasses at any level of abstraction with the efficiency of searching only the currently active paths.

Moreover, if several paths are concurrently active and therefore several solutions are concurrently possible a natural criterion for preference exists in DUAL. Instead of using some fixed predefined criterion as "the shorter path", or "the path crossing nodes with lower fun outs", or any combination of these[7] a more flexible and more natural criterion is used – the more active path is selected as the path which is more relevant to the current context. It may happen that this is the shorter path (and probably this will occur more frequently), but it could also happen that this is the longer path (which may be a rare situation but allows interesting and deep analogies).

There is an additional flexibility in the system: its behavior becomes context-sensitive, i.e. in different contexts - different concepts will be active and therefore different paths will be followed by the markers (Figure 4) and as result different correspondences will be found. This is especially important in cognitive modeling as it is well-known that human cognition is context-sensitive[11, 12].

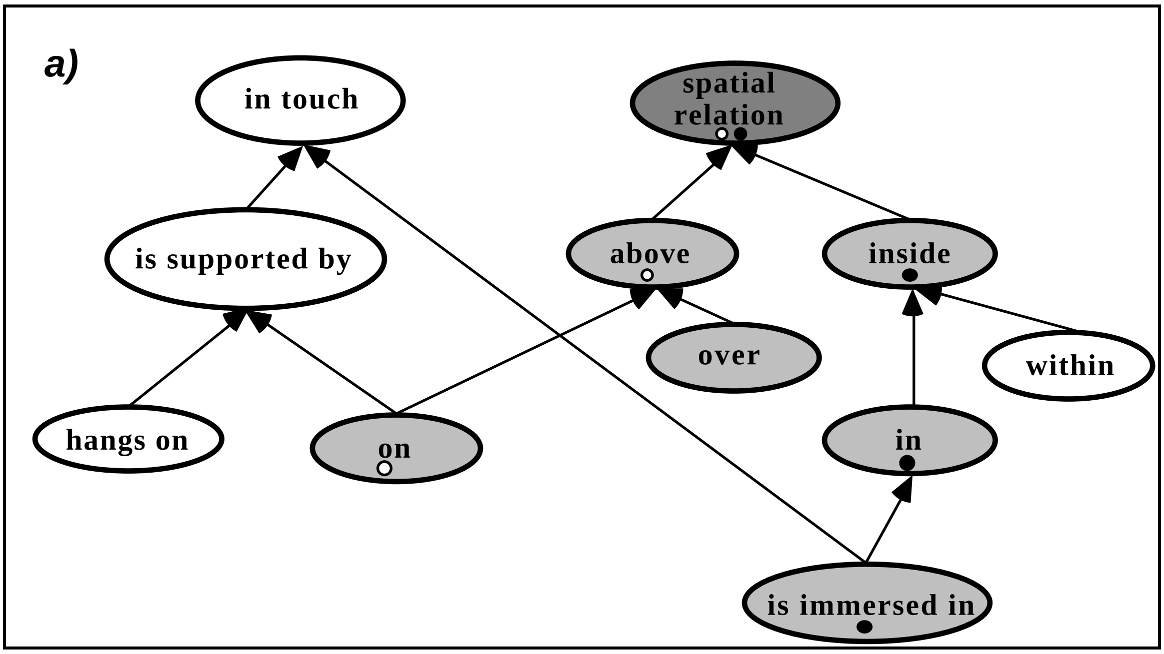
5. Conclusions

Models based on the DUAL architecture demonstrate high flexibility and variability in their behavior thereby reflecting the dynamic context-sensitive nature of human cognition. On the other hand they demonstrate high efficiency restricting all searches to small parts of the knowledge base.

The DUAL architecture has been used in modeling similarity judgements[13] and analogical reasoning[14].

6. Acknowledgments.

This research has been partially supported by the Bulgarian National Science Fund.



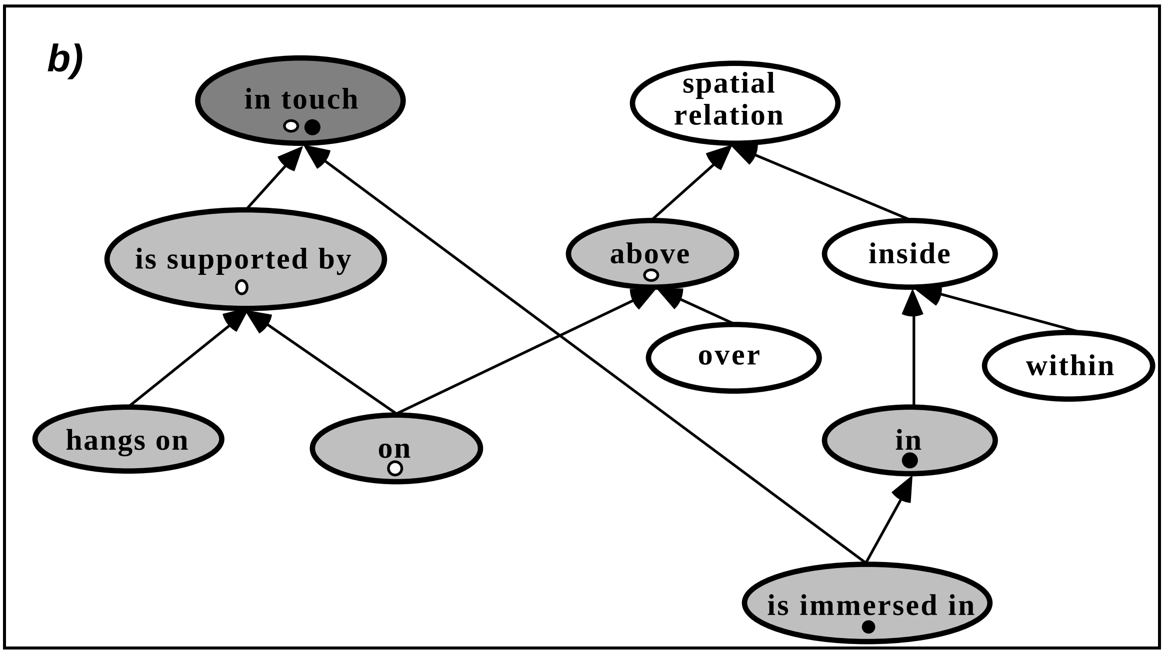


Figure 4.
Context-Sensitive Marker-Passing.
Depending on the particular memory state – the distribution of activity over the network (presented by the filling patterns of the nodes) – different ways will be followed by the markers (the white and black small circles) and consequently different correspondences will be established.

7. References

1. J. Barnden & J. Pollack, *High-Level Connectionist Models, Advances in Connectionist and Neural Computation Theory*, **1,** (Ablex Publ. Corp., Norwood, NJ, 1991).
2. L. Bookman & R. Sun Special Issue: Integrating Neural and Symbolic Processes. *Connection Science*, **5(3&4)** (1993).
3. B. French, *Tabletop*. (PhD Thesis. Univ. of Michigan, 1991).
4. J. Dinsmore, *The Symbolic and Connectionist Paradigms: Closing the Gap*. (Lawrence Erlbaum Associates, Hillsdale, NJ, 1992).
5. D. Gentner, Structure-Mapping: A Theoretical Framework for Analogy, *Cognitive Science* **7(2)**, (1983), pp. 155-170.
6. D. Gentner, The Mechanisms of Analogical Learning. In: S. Vosniadou & A. Ortony (eds.) *Similarity and Analogical Reasoning*. (Cambridge Univ. Press., New York, NY, 1989).
7. J. Hendler, Marker Passing over Microfeatures: Towards a Hybrid Symbolic/Connectionist Model. *Cognitive Science*, **13**, (1989). pp. 79-106.

8. J. Hendler, Hybrid Systems (Symbolic/Connectionist). Special Issue of *Connection Science*, **1(3)**, (1989).
9. D. Hofstadter & M. Mitchell, The Copycat Project: A Model of Mental Fluidity and Analogy-Making. In: Holyoak, K. & Barnden, J. (eds.) *Analogical Connections, Advances in Connectionist and Neural Computation Theory*, **2**, (Ablex Publ. Corp. Norwood, NJ, in press).
10. K. Holyoak & P. Thagard Analogical Mapping by Constraint Satisfaction. *Cognitive Science*, **13,** (1989), pp. 295-355.
11. B. Kokinov, About Modeling Some Aspects of Human Memory. In: Klix, Streitz, Waern, Wandke (eds.) *MACINTER II*, (North-Holland, Amsterdam, 1989).
12. B. Kokinov, Associative Memory-Based Reasoning: Some Experimental Results. In: *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, (Lawrence Erlbaum Associates, Hillsdale, NJ, 1990).
13. B. Kokinov, Similarity in Analogical Reasoning. In: B. du Boulay & V. Sgurev (eds.) *Artificial Intelligence* **V**. (Elsevier, Amsterdam, 1992).
14. B. Kokinov, A Hybrid Model of Reasoning by Analogy. In: Holyoak, K. & Barnden, J. (eds.) *Analogical Connections, Advances in Connectionist and Neural Computation Theory*, **2**, (Ablex Publ. Corp., Norwood, NJ, in press).
15. P. Thagard, K. Holyoak, G. Nelson, D. Gochfeld, Analog Retrieval by Constraint Satisfaction. *Artificial Intelligence*, **46**, (1990).pp. 259-310.
16. C. Thornton, Special Issue on Hybrid Models. *AISB Newsletter*, **78**, (1991).