

EMBODIED CONVERSATIONAL AGENT BASED ON THE DUAL COGNITIVE ARCHITECTURE

Stefan Kostadinov, Georgi Petkov and Maurice Grinberg
*Central and Eastern European Center for Cognitive Science, New Bulgarian University
Montevideo str. 21, Sofia 1618, Bulgaria*

Keywords: ECA, cognitive agent, software assistant, context-sensitivity, priming effects.

Abstract: A working model of an ECA with cognitive capabilities based on the DUAL cognitive architecture is described. The cognitive model used inherits the advantages of a high context-sensitivity, general and episodic memory, and reasoning by analogy of the DUAL/AMBR model. However, several crucial new mechanisms are proposed which allow for the continuous functioning of the agent and the completion of several question-answer cycles with meaningful priming and context effects. This paper presents these mechanisms and discusses the results of simulations of a user-agent interaction session.

1 INTRODUCTION

RASCALLI (Responsive Artificial Situated Cognitive Agents Living and Learning on the Internet) is a FP6 EC project (see <http://www.ofai.at/rascalli> for more information) aimed at the development of a platform whose purpose is to help users to search for information in Internet and in large data-bases and ontologies by communicating with an Embodied Conversational Agent (ECA). This ECA should be able to understand and answer questions, look for and find information in Internet, but also memorize the interactions with the user and the environment, and learn from its experience. Thus it will start to know its user and his/her preferences and adapt its activities in order to achieve better completion of the given tasks.

In order to naturally conceptualize and model Rascalli's virtual life in a virtual environment a 'human' metaphor has been adopted.

The **mind** – specialized to Rascalli' specific knowledge structure and tasks – communication with its owner, type of knowledge (e.g. music), events etc. The mind operates only on represented knowledge and has only a mediated connection to the body and the environment. Thus it contains a partial, selected representation of the environment at abstract conceptual level and experiential memories related to specific episodes: interaction of Rascalli

with user(s), other Rascalli and the environment. The interaction with the environment and the body is mediated by the sensory-motor layer.

The **Sensory-Motor Layer** consist of two main parts – the **Perception Layer** that selects the information provided by the **Sensors** (e.g. the translation of specific question from the user) and translates this information in the symbolic form required by the mind and the **Action Layer** that translates action commands from the symbolic form used by the mind into specific command to the body.

The **body** (e.g. specific tools for translating a question or send a query to DB) consists of various sensors and effectors which allow Rascalli to acquire information from the environment and to perform actions in it.

Rascalli acts in an **environment**, which is defined as everything outside the Rascalli like the user(s), other Rascalli, knowledge bases (KB), external tools that would be able to function without the Rascalli, etc.

The work, presented in this paper mainly focuses on the mind. Where necessary for the explanation of the integration of the mind in the general Rascalli platform some communication tools with the Sensory-Motor layer will be mentioned and their function explained.

The core of the Rascalli, their mind, is based on the cognitive architecture DUAL and the analogy-making cognitive model AMBR (Kokinov, 1994, Kokinov & Petrov, 2001). The mind includes a Long

Term Memory (LTM) where general and episodic knowledge is stored and a Working Memory (WM) which is the active part of the LTM, perceptual input and goals. LTM contains concepts (including relations) as well as instances of concepts, organized in coalitions to represent tools, episodes of interaction with the owner, already acquired knowledge, etc.

2 DUAL ARCHITECTURE AND THE AMBR MODEL

The DUAL architecture consists of a large number of relatively simple interconnected hybrid (connectionist and symbolic) micro agents. The main advantages of the architecture are its context-sensitivity, based on spreading of activation, dynamic and emergent symbolic computations.

The main purpose of DUAL/AMBR development has been the modeling of human analogy making (Kokinov & Petrov, 2001). Various simulations have been performed with the AMBR model and compared successfully to empirical data of analogy-making related to structural constraints, context effects, and blending of memory episodes.

The following section discusses the added mechanisms that allow the mind to perform cycles of perception-action-communication.

3 RASCALLI'S MIND

As stated in the first section, the mind is part of Rascalli platform together with the body. This section is about the mechanisms that allow the mind of Rascalli to function inside the general platform by:

- dealing with a question (perception);
- extracting information from its own memories or from an source in the Environment (action and formation of a solution space);
- selecting the right solution (judgment and decision making);
- presenting it to the user (communication);
- store the interaction as an episode in LTM (evaluation and learning);
- being ready for the next question without losing the context of the previous one (continuous functioning in a given context).

3.1 DUAL/AMBR Mechanisms

As mentioned above, DUAL/AMBR is built of a relatively large number of interconnected DUAL micro agents. An utterance is represented in a structured form (as a coalition of micro agents) and in order to be 'perceived' by Rascalli it must be attached to the INPUT and GOAL nodes. The micro agents representing the question become target micro agents (which comes from the terminology used in analogy research). The INPUT and GOAL nodes are the only source of activation of the architecture, so they activate the question coalition and via them the concept level micro agents to which they are linked. The concept micro agents activate their instances through the inverse links. Thus, activation spreads throughout LTM and the micro agents which become active enough enter WM and start participating in the analogy mapping and transfer mechanisms. These mappings range from direct correspondence to distant analogical objects which allow the transfer of knowledge from episode in domains different from the utterance domain.

A mechanism based on anticipation, first introduced in a robot implementation has been adapted for Rascalli (Petkov et al., 2006).

3.2 Specific Knowledge Transfer Mechanism

The DUAL/AMBR mapping mechanisms along with the added anticipatory mechanism (Petkov et al., 2006), are too unspecific and cannot lead to knowledge transfer. Thus new mechanisms had to be developed, on the first place – knowledge extraction mechanism.

The utterances must be represented in a form which contains information about the provided details and, if present - specific answer expected (e.g. a name of a music album or a child, see the examples in Section 4). This form is provided through NLP analysis by the input processing tool that handles the utterances from the user.

Thus the utterances presented to the mind can have two tags - 'of-interest' for the elements of information given and 'question' to define what is specifically asked for if the latter can be extracted from the question.

The specific knowledge transfer mechanism comes into play after one of the arguments of a certain relation is mapped. Then, the other arguments are directly transferred after verifying whether the first argument has tag 'of-interest'. At

the same time, the extracted information can replace some empty placeholders that have the tag ‘:question’. This new mechanism works locally and in parallel with all other mechanisms. The relevance requirement, however, still holds because knowledge retrieval is constrained in two ways: first, transferred micro agents should be sufficiently active (i.e. relevant); and second, the tag ‘:of-interest’ should be present in the utterance elements for a transfer of specific information.

3.3 Action Transfer Mechanism

The final mechanism needed to close the perception-action-communication cycle is the selection and sending of an action command. It is triggered by the anticipated cause-relations that are linked to the GOAL node(s) (Petkov et al., 2006). The cause-agents, as indicated by their name, represent causal relations. If a cause-agent is linked to a goal agent (e.g. ‘find-album’), it receives the ‘close-to-goal’ message. If a ‘close-to-goal’ cause-agent participates in a winner-hypothesis, it checks its antecedents for action micro agents (micro agent describing an action). If all the above conditions are met, the action mechanism executes the action.

To put it simple, when a whole structure from INPUT to GOAL, supported by enough winner-hypotheses establishes, the respective actions would be triggered for execution. The action is sent to the Sensory-Motor layer, that further processes it and sends it to the appropriate tool.

3.4 WM Cleanup and Learning

The capability of Rascalli to give reasonable, context-sensitive, and flexible answers to simple questions relies on previous knowledge in LTM. Without the possibility to acquire new knowledge and to modify the existing one the system would be rigid and limited.

Thus, various mechanisms for working memory cleanup and episode storage have been developed. They can be summarized with the following algorithm: (1) Define the moment when the goal is achieved. After that: (2) erase all current correspondence hypotheses. (3) Delete all markers in all concepts. (4) Terminate all suspended symbolic operations. (5) Create a new episode with all the elements from the current one including the answer and the user evaluation. (6) Adjust/create new inverse links from concepts to instances.

Equipped with these routines for WM cleanup and episode storage, the system is able to work

continuously, without interruption between the cycles; it enriches its memory with new information after each session, and it is able to support and use the context of a continuous conversation.

All these abilities of Rascalli are demonstrated with the simulation, presented in the next section.

3.5 Mind and Body

As described above the body of the Rascalli platform provides an interface to various tools for communication, exploration and information acquisition. The tools and the mind communicate via a sensory-motor layer that translates the agents from the mind into RDF (see <http://www.w3.org/RDF/> for details) messages to the tools and vice-versa. The tools themselves carry out various tasks – translating natural language into RDF graphs, translate RDF graphs into natural language and voiced by Rascalli, search in DB, consult Google, etc.

The Sensory-Motor Layer essentially translates RDF graphs into DUAL micro agent structures and vice-versa. The Action Layer additionally decides which tool to use based on the RDF command. This process is completely automated, as the mind’s internal representation format and the RDF ontology have a similar structure (e.g. semantic graph).

The current implementation of the mind deals with three basic tools – for input processing, database search and output of messages to the user. This is the minimal set of tools required for Rascalli to understand a request from the user, undertake some action(s) to satisfy this request and finally report the answer back.

4 PUTTING EVERYTHING TOGETHER: SIMULATIONS

The scenario demonstrating the system capabilities consists of a dialog of five utterances in the music domain – artists and details about their personal lives like religion, children, etc.

The **first utterance** is: “Tell me something about Britney Spears”. The input processing tool processes the words and sends the message representation to the input of the mind. Britney Spears is of interest to the mind, so it tries to transfer information and link it to the Britney Spears. The mind has in its LTM information about Britney Spears so it is activated by the question and is transferred by the anticipation transfer mechanism described in Section 3 and the parts in this information compete among them. Eventually, the information about the album

Blackout wins the competition, as it is considered most relevant and it is sent as answer to the user.

The **second utterance** is a question: “Who are the children of Madonna?” It can be noticed that this time the utterance is specific about what is needed – the names of Madonna’s children – so the node representing it has the tag ‘:question’. The rest of the message has the tag ‘:of-interest’.

The system tries to replace ‘child’ (with tag ‘:question’, subsection 3.3) with information from LTM. We assume that this information is available to the mind so it is represented and attached to the corresponding concepts of LTM.

The **third utterance** from the simulation is the same as the first one: “Tell me something about Britney Spears.” One option for the mind is to answer as in the first question by giving the name of an album. But its internal state is determined by the second question related to the children of Madonna. There is no information about the children of Britney Spears in LTM so the mind primed by the second question decides to search for it in DB where such type of information is available. The command sent to the data source search tool contains the Britney Spears, the ‘has-child’ predicate and the ‘child’ as something to be filled in. The former two are marked with ‘:of-interest’ tag and the latter with the ‘:question’ tag. This information allows the data source search tool to transform this message into a search in the musical DB with key words ‘child’ and ‘Britney Spears’, the answer is completed with the new information and sent to the user via the output tool.

The **fourth utterance** is again a specific question: “What is the religion of Madonna?” The mind has such information, so it transfers it to the target and thus provides the answer to the user.

The **fifth utterance** completes the priming demonstration of the scenario. It is again the same as the first and the third question: “Tell me something about Britney Spears”. Again the mind has this information in the LTM and directly provides the answer – Britney is Christian.

5 CONCLUSIONS

In this paper we presented a full working model of the mind of a future ECA based on the cognitive architecture DUAL augmented with a number of new mechanisms. The agent is able to carry on a simple conversation consisting of a series of questions and displays context sensitivity in its answers – an essential trait for a more natural and flexible conversation with a user.

The performance observed is a combination of DUAL/AMBR mechanisms and a set of newly developed ones based on the main principles of this cognitive architecture.

The simulation demonstrates that the major mechanisms needed for realistic situations are available in Rascalli’s mind. Rascalli can encode the incoming information, can reason using cognitive mechanisms, can act according to the tasks, and can learn and adapt itself.

The newly developed agent will be integrated in the general Rascalli platform developed in the Rascalli project and efforts are currently in progress to refine the automatic question encoding for at least a limited set of simple questions.

ACKNOWLEDGEMENTS

This work was supported by the Rascalli FP6 project with EC.

REFERENCES

- Kokinov, B. & Petrov, A. (2001) Integration of Memory and Reasoning in Analogy-Making: The AMBR Model. In: *Gentner, D., Holyoak, K., Kokinov, B. (eds.) The Analogical Mind: Perspectives from Cognitive Science, Cambridge, MA: MIT Press.*
- Petkov, G., Naydenov, Ch., Grinberg, M., Kokinov, B. (2006). Building Robots with Analogy-Based Anticipation. In: *C. Freksa, M. Kohlhase, and K. Schill (eds.) Proceedings of the 29th German Conference on Artificial Intelligence (KI-2006), LNAI 4314, Bremen, Germany, 2007. Springer, pp. 72-86.*
- Krenn, B. et al. (2007) A Smart Music Companion. In: *AAMAS 2008 Special Track on Virtual Agents*, submitted