Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 31 – 37

ISBN 978-83-60810-14-9
ISSN 1896-7094

The Triple Model: Combining Cutting-Edge Web Technologies with a Cognitive Model in an ECA

Maurice Grinberg and Stefan Kostadinov
Central and Eastern European Center for Cognitive Science, New Bulgarian University,
Montevideo 21, 1618 Sofia Bulgaria
Email: mgrinberg@nbu.bg, stefan@yobul.com

***Abstract*—This paper introduces a new model which is intended to combine the power of a connectionist engine based on fast matrix calculation, RDF based memory and inference mechanisms, and affective computing in a hybrid cognitive model. The model is called Triple and has the following three parts: a reasoning module making advantage of RDF based long-term memory by performing fast inferences, a fast connectionist mapping engine, which can establish relevance and similarities, including analogies, and an emotional module which modulates the functioning of the model. The reasoning engine synchronizes the connectionist and the emotional modules which run in parallel, and controls the communication with the user, retrieval from memory, transfer of knowledge, and action execution. The most important cognitive aspects of the model are context sensitivity, specific experiential episodic knowledge and learning. At the same time, the model provides mechanisms of selective attention and action based on anticipation by analogy. The inference and the connectionist modules can be optimized for high performance and thus ensure the real-time usage of the model in agent platforms supporting embodied conversational agents.**

I Introduction

INTERNET has become an extremely rich environment, which starts to become comparable with a real life environment – the available information is too abundant and the existing options are too numerous to be considered and taken into account completely. Even in the case, in which we can make all possible inferences based on a large ontology in extremely short time (of the order of milliseconds), the increase of information seems to be too large to be useful. It seems that although there is some control in the way the information is presented, namely sometimes in a structured form, pure AI approaches are not sufficient to design and implement artificial cognitive systems in Internet that can achieve human level of performance. Analogously to robotics, accomplishing tasks in complex environments seems to require novel (with respect to GOFAI) approaches starting from the connectionist modeling and going to more and more biologically inspired architectures. Similarly, inference based on ontologies can be done extremely efficiently but this is a problem in itself because of the risks of information explosion.

In cognitive systems this problem is addressed first of all by introducing the concept of working memory (WM) which broadly speaking can represent the most relevant part of the long term memory (LTM) which contains all the knowledge an agent knows. The relevance of the information retrieved from LTM can be insured by activation spreading mechanisms, with the goal and the perceived input (e.g. task, scene etc.) as a source of activation (e.g. see [1]). Such activation spreading can use the connections in LTM or some other types of connection – associative, based on semantical similarity, etc. Other possible mechanisms of selectivity of the information used can be based on anticipatory mechanisms in perception and action (e. g. see [2]—[4]) or other selective attention mechanisms. The role of emotions in cognitive modeling and especially in embodied conversational agents (ECA) has been given a lot of attention recently (e.g. see [5]) and seems to be an important set of mechanisms which can add a lot in communication with users but also influence the reasoning processes. In the case of an ECA which is supposed to interact with users in real time, the problems of scalability and processing time become even more important and efficiency issues are of primary concern.

The model Triple, introduced here for the first time, is aimed at being a cognitive model for cognitive systems and particularly for ECA platforms. It includes, on one hand, several of the necessary mechanisms mentioned above and

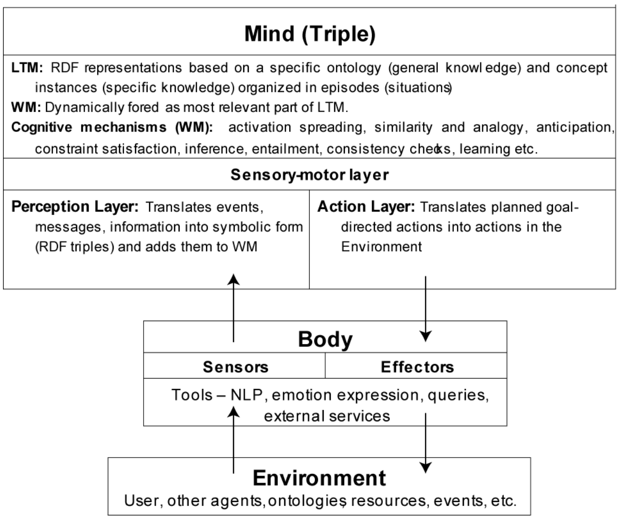
Mind (Triple)
LTM: RDF representations based on a specific ontology (general knowl edge) and concept instances (specific knowledge) organized in episodes (situations)
WM: Dynamically fored as most relevant part of LTM.
Cognitive mechanisms (WM): activation spreading, similarity and analogy, anticipation, constraint satisfaction, inference, entailment, consistency checks, learning etc.
Sensory-motor layer
Perception Layer: Translates events, messages, information into symbolic form (RDF triples) and adds them to WM
Action Layer: Translates planned goal-directed actions into actions in the Environment
Body
Sensors
Effectors
Tools – NLP, emotion expression, queries, external services
Environment
User, other agents, ontologies, resources, events, etc.

Fig 1. A Mind-Body-Environment conceptualization of an ECA, using the Triple model as Mind.

on the other it tries to achieve maximal computational efficiency in order to allow real time functioning of the ECA. These two constraints lie at the basis of this model: adding all the useful cognitive modeling techniques which allow flexibility, context sensitivity and selectivity of the agent and in the same time – maximal computational optimization of the code and use of very efficient inference methods (e.g. see [6]).

Following this strategy, the model has been designed in three parts that function in parallel. The so called Reasoning Engine (RE) is coordinating and synchronizing the activities of the model and relates the agent with the environment (e. g. user, other agents, etc.) and with the tools the agent can use (e. g. tools to communicate with the user, make actions like access ontologies and data bases, search the Internet (see [3] and [4]), extract LSA information from documents, etc.). RE is also responsible for instance learning – storing of useful episodes in LTM after evaluation. The Inference Engine (IE) is used by RE and can operate on demand. Its main role is to augment parts of WM with inferred knowledge and do consistency checks.

The second part of Triple is the so-called Similarity Assessment Engine (SAE). It is designed to be a connectionist engine, based on fast matrix operations and is supposed to run all the time as an independent parallel process. The main mechanism is activation spreading in combination with additional mechanisms which allow to retrieve knowledge relevant to the task at hand. Communication of SAE with RE is elicited by events related to the level of activation. The SAE mechanisms are supposed to lead to the retrieval of information which is related to the task at hand at different level of abstraction.

The third part is the Emotion Engine (EE) which is based on the FAtiMA emotional agent architecture [7, 8]. FatiMA generates emotions from a subjective appraisal of events and is based on the OCC cognitive theory of emotions [9]. EE, similarly to SAE, is supposed to run in parallel and influence various parameters of the model like the volume of WM, the speed of processing, etc. (see [10] for a possible role of emotions in analogy making). In the same time it will allow achieving higher believability and usability based on the emotional expressions corresponding to the current emotional state of the agent.

The Triple model is connected to the DUAL/AMBR model [1] by inheriting some important mechanisms. Triple like DUAL/AMBR makes use of spreading of activation as a method for retrieval from memory of the most relevant episodic or general knowledge. The mapping of the knowledge retrieved to the task at hand and to the current input to the system is based on similarity and analogy in both models. However the underlying mechanisms are essentially different. In DUAL/AMBR knowledge representation is based on a large number of micro-agents which perform local, decentralized operations and are dualistic in the sense that they spread activation and perform symbolic operations at the same time. The messages exchanged between the micro-agents trigger the establishment of mappings, structural correspondence assessment (which stand for concepts and relations) and the speed of the symbolic processing of the messages depends on their so-called 'energy' (basically an integral over the recent activation). In Triple, an attempt has been made to achieve the same functionality on the basis of clearly separated symbolic mechanisms (reasoning, inference, consistency checks, anticipation, etc.) and connectionist mechanisms (spreading of activation over different types of connections, similarity assessment, distributed representations, etc.). A third component is the emotional module (EE) which is missing in DUAL/AMBR [10]. In DUAL/AMBR the 'duality' is achieved at the level of each micro-agent while in Triple it is achieved by two systems which run in parallel and communicate on event-driven basis. An important additional difference is that Triple is using a full fledged reasoning part in the standard AI sense, which is not available in DUAL/AMBR. The inference and entailment capabilities are integrated with the spreading of activation and evaluation of retrieval and action planning. Only the most active part of WM, corresponding to the focus of attention of the system is subject to augmentation based on inference and to other symbolic processing like evaluation, transfer, and action. The Amine platform [11] has similar augmentation mechanisms which are based on purely symbolic manipulation and are not conditioned by the attention of the system (see [11] and [12] for similar mechanisms like 'elicitation' and 'elaboration').

At this stage Triple inherits from DUAL/AMBR all the mechanisms of transfer of knowledge and anticipation from LTM based on analogy-like reasoning (e.g. see [3] and [4]) but additionally uses consistency checks and inference. The latter is expected to improve efficiency considerably because it will allow timely canceling of impossible plans and will not rely only on external feed back.

One of the main roles of an ECA is to be a sophisticated intelligent interface between a human and a virtual or real (artificial or natural) environments (e.g. see [13] and [14]). An example of such an environments is the Internet, as discussed above. In order to think of the agent as embodied and situated in any environment, the structure shown in Fig 1 has been adopted. The advantage of this representation is the possibility to consider physically and virtually embodied agents on equal footing. It supports also the conceptualization of Internet (or any virtual environment) made above which implies that sufficiently rich virtual environments should be considered as 'real' ones with respect to their richness and complexity. In Fig 1, the Mind is shown with its specific knowledge structures and tasks. The Sensory-Motor Layer makes a mediated connection between the Mind, and the Tools of the agent, which are the sensors and effectors that work with the Sensory-Motor Layer and perceive or act on the Environment. The Sensory-Motor Layer provides symbolically represented knowledge to the Mind and thus makes mediated connection between the Mind and the Environment.

The user occupies the central part in a ECA environment and has a complex set of goals, needs, interests, expectations and previous knowledge. On the other hand, as stressed for instance in [5], any interaction with the user has its affective and emotional background which is highly user and situation specific. In order for such an interface to be maximally efficient under these conditions it has to be highly personalized and context sensitive. It seems that this cannot be done only on the basis of statistical information gathering (e.g. like in a recommender system). The agent has to be a real personalized partner which remembers important (or all) past episodes of interaction with the user and thus is able to establish a deeper relationship with the her based on personalized common experience. For instance, in these memories, the satisfaction of the user should be encoded as, generally speaking, the agent is supposed to satisfy the user. This personalization is achieved in Triple by the rich episodic memory in which all kinds of episodes and information are stored: general knowledge, typical and specific situations, events and user evaluations of task completion outcomes. Examples of how such episodic knowledge can be used can be found in [3] and [4]. Episodes are a rich source of personalized experience and can be used for analogical reasoning or case-based reasoning. They represent a rich source for user-centred learning and generalization. They can encode, among other things, the preferences of the latter in various contexts, his/her definitions of key concepts, etc. and thus allow for information selection and form of presentation.

As our previous experience has shown [4], the success of such an approach is based on the richness of episodic knowledge and in the way the LTM part relevant to the task is accessed. The encoding of episodes is unavoidably specific and unique (a specific set of concept and relation instances is used). The specificity is guaranteed by the learning mechanism consisting in the storage (retention) of task completion episodes and knowledge provided by the user in LTM. It may turn out that the questions or goals are formulated differently than the episodes stored in LTM although using the same concepts, e.g. by using different relations between the same instances of concepts. In order to make use of previously encoded knowledge one needs mechanisms generating various equivalent representations to overcome the limitations of a specific encoding (e.g. the possibility to transform “John is the father of Mary” into “Mary is the daughter of John”). It is obvious that flexibility and reliability in this direction can be achieved only if the similarity and equivalence of knowledge could be assessed, as well as its consistency with previous knowledge. In order to be able to do so, the ECA should dispose of powerful inference and reasoning capabilities in order to achieve maximal flexibility. Thus in comparison to DUAL/AMBR, additionally to analogy making, the Triple model extensively uses rule-based inference and entailment based on previous knowledge based on its relevance.

In the following sections, the main principles of Triple will be presented in more detail and the implementation of the main modules discussed.

II The Model Implementation

The model presented here is a continuation of an effort to design an ECA architecture with cognitively based mind for a real application [15]. The attempt to use the DUAL/AMBR architecture [1] made evident the existence of scalability problems and the need for faster and more flexible model mechanisms. This lead to the creation of the model Triple.

The embedding of Triple in a real agent platform is currently in progress but some simple evaluations have been already performed with partial functionality and showed promising results in terms of efficiency. The platform implementation (see [15] for details) uses the Nebula engine [16] for the multi-modal generation and NLP processing and speech synthesis [17, 18]. In order to provide the required speed of processing for a real-time application, fast OWLIM RDF inferencing and handling was provided [6] and adapted to the model needs. RDF triples representations and inference on them play a crucial role in the implementation of LTM and IE. All this, combined with the fast matrix-manipulation-based similarity assessment engine provides response time of a few seconds for simple tasks as estimated with the initial implementation of the model.

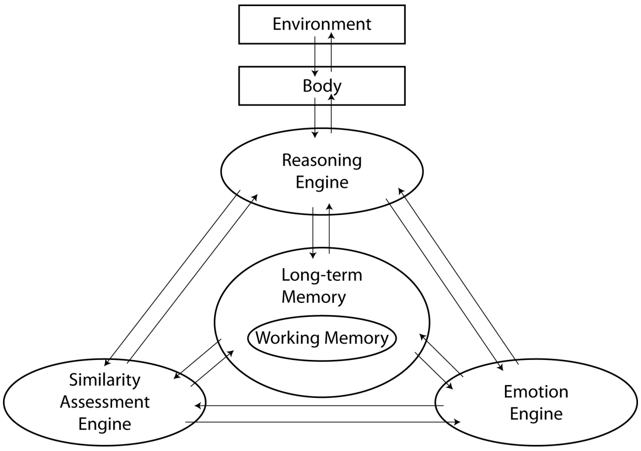

Fig 2. The Triple architecture and interactions among the IE, SAE and RE. (See the text for explanations.)

It should be stressed that the model is intended to be general enough and is not related in an essential way to the use of the specific tools quoted above.

A typical interaction episode with an user would be similar to the interaction episodes described in [4], where DUAL/AMBR has been used as a cognitive model. The user asks a question which is processed by NLP tool [17] and the information about the type of utterance (greeting, question, task, etc.), what is asked for (if the utterance is a question), and what is the knowledge contained in the utterance is expressed in RDF triples and attached to WM as instances of existing in LTM shared concepts. The set of nodes represents the goal or the task given by the user to the agent and is labeled as 'target'. The target could include any additional input accessible to the agent via its 'sensors'. The basic process of task completion is to augment this target set of nodes with knowledge from LTM or from external ontologies or sources and formulate an answer or reaction to the user utterance. The following subsections explain in detail how this is achieved and what is the interplay between the three processing modules of Triple presented schematically in Fig 2.

A. Similarity Assessment Engine (SAE)

As discussed in previous sections, the agent's knowledge is represented as concepts (including relations) and their instances by nodes related through weighted links forming a semantic network which could include other types of connection (e.g. associative ones) especially in the episodic part (see [1]). The main problem to be solved by SAE is to assess the level of similarity or correspondence between two nodes typically belonging to the target (the task) of the agent and to LTM, respectively. The problem with similarity assessment is central for such approaches and has been extensively explored in the analogy and case-based reasoning literature (e.g. see [19, 20, 21] and the references there in). The Similarity Assessment Engine (SAE) is exploring some established and some new approaches in finding similarities aiming at the highest possible efficiency.

Typically, part of the nodes come from the target (e.g. a task by the user or an internal goal for the agent) and the others come from the active part of LTM and both form the WM content. The normalized similarity (lying between 0 and 1) between target and LTM nodes is taken to be the probability of correspondence between the two nodes. The basis for similarity evaluation can be quite various and generally speaking takes into account the taxonomic links and the proximity of the nodes in the semantic tree by using different types of distributed representations, some of which will be explained bellow. Additional mechanisms as the one proposed in [19] using Latent Semantic Analysis (LSA) [22, 23] and are based on semantic similarity.

The connectionist processing is based on the representation of the WM of the agent as a set of matrices, that reflect different dependencies among the concepts, relations and their instances. It should be stressed that there are no named connections in this representation and all relations including 'instance-of', 'sub-class', 'super-class' and others are represented as nodes, linked by weighted connections. Thus all the contents of the WM are represented as a vector of nodes (concepts, relations and their instances) and the connections are a matrix of weights. All the nodes related to the input and the goal for the agent are part of a 'target' set of nodes and the achievement of the goal (e. g. finding the answer of a question) is formed by augmenting this target set so that it contains the results of the processing and include completion of the goal or a failure. In order to do so the agent must retrieve knowledge similar (or analogical) to the input and transfer the required knowledge to the target or trigger a plan of actions in order to find it, e. g. search in an ontology or data source. So, to each node in WM a vector is attached, which contains distributed information about its definition, relations and actions in which it participates, its instances or super-classes, etc. which can be used to measure its similarity to other nodes. It should be noted that this distributed representation is spanned only over the nodes in WM, i. e. the nodes with sufficient activation. Moreover, each term in the distributed representation is multiplied by the corresponding node activation. This procedure will change the similarity between nodes depending on the activities of the nodes present in the WM. Thus similarity becomes dynamic and changes depending on the content of WM at a given time and thus is highly context dependent. For example if the neighboring nodes of a node have a higher activation than the higher concepts in the taxonomy more specific similarity will be found and if the higher concepts are with larger activation more abstract and high level similarity (analogy) will be established. The similarity depends on a similarity measure between any two such characteristic vectors (e.g. a normalized scalar product). This normalized similarity is interpreted in the model as the probability for useful correspondence between the two nodes which eventually will lead to the retrieval and mapping of the target task to a relevant part in LTM, which may contain the required knowledge for task completion.

This principle of similarity assessment seems to be quite general and we are planing to test and combine different implementations. All are based on the relations between the nodes in Triple LTM. Other methods we are considering are the ones based on Latency Semantic Analysis [15, 16] for the respective domains. Additional mechanisms, presently explored are related to spreading of different type of activation which require much more space and will be reported in a separate future publication as applied to a music domain. The SAE has been implemented using compiled Matlab code and making use of the sparse matrices manipulation routines.

B. Emotional Engine(EE)

The Emotional Engine works continuously and in parallel with the SAE and provides continuous fine-tuning of the rest of the Mind modules and, from time to time – takes control over the reasoning process (see end of this section). The so-called fine-tuning is done as the EE is constantly decaying to a “neutral” state. The size of the Working Memory, the severity of the reasoning constraints, etc. all depend on the emotional state. As the emotional state changes over time, when major changes occur, events are sent to the Reasoning Engine. An example of such event is the case where an action of information retrieval from an Internet web-service is performed. If the service does not respond at all, the emotional state would decay over time and thus the action that triggered the emotional state would be suspended. This solution seems better than to have a fixed amount of “real” time (in seconds), as the EE depends on the time elapsed, the importance and desirability of the action performed, the initial emotional state and any other events that might occur in the meantime. When an action is interrupted if proven unsuccessful, or because the EE halted it, the next action in the line is going to be executed (see Reasoning Engine section for details). The details about this engine will be published in a forthcoming paper [8].

C. Reasoning Engine (RE)

The reasoning engine integrates and controls the operation of the SAE, IE, and EE on one hand and the communication flow with the Body (and thus with the Environment) on the other (see Fig 2). It works with the sensory-motor part of the Mind by receiving the information from the sensors (user utterances, information about results from actions, etc.) and sending action commands to the Environment via the Body's tools (Multi-Modal Generation, Data Source Lookup, etc.). The main interactions with the Body and Environment are the same as the ones reported in [4] and are briefly explained in the next section.

I Information flow

Information (in RDF triple form) received by the Mind is either a new information from the Environment (question, task, definition, etc.), or a response from an action (result from a tool). The human input is considered as information from the Environment and is processed by the NLP tool into a set of RDF triples. Each coalition, coming from the Body, is assigned a context, and if there is no current task, a new context is created.

When there is an action coalition of nodes transferred (see next section), the RE identifies the statements related to this coalition and sends them to the Body as an "action command". In order to be meaningful, this action command must adhere to the requirements of the specific tool it is addressed to.

II Main mechanisms

When a new message is received from the Body, the RE adds it to WM (and thus – to LTM) and marks all the statements from the message as "target". All the parts of the message, that are not internal (e.g. which Tool generated the message) are marked as being "goal" for the system (see [4]). The target set (called "input" and "goal") is the source of activation for the SAE module. This module is started by RE and gives continuously information about similarities found between the target set and knowledge in WM and initially determines the focus of attention of the agent – the most active part in WM. SAE estimates the level of similarity and based on that RE establishes candidate correspondences between the target set and the LTM contents in WM in the so-called Similarity Assessment and Correspondence Processors.

It should be stressed that in all tasks the ultimate goal is to satisfy the user by providing the needed information or solution of the task.

More precisely, when the task is to answer a question this general goal would be to provide the user (or another agent) with the answer. Initially, the goal of the system is quite general but with the processing of the question it becomes more specific. For instance, for the currently implemented music domain [15], the goal could be to give the name of an album of a singer, his/hers religious status, birth date, etc. In the current, early development stage, the architecture does not have an explicit planning mechanisms, although the reinforcement learning (by always aiming at user's satisfaction), top-down learning (episode retention) and action decision are present and should allow the model to find better and better solutions over time and encode sequences of actions into episodes. During the future development, different approaches will be considered for inclusion like BDI features and emotionally guided planning.

As stated in the beginning of this section, messages from the Body are added to the WM and become source of activation for the SAE that starts to send similarity assessments to the RE. Those assessments are checked for obvious flaws and inconsistencies due to the fact that SAE is supposed to make analogies as well. If no flaws are found, they are transformed to established correspondences. When the correspondences between the target set and LTM are established the IE is used to verify and evaluate them and eventually the candidates are rejected or confirmed. Based on the existing correspondences, parts of past episodes are evaluated by the IE and transferred to the target set until eventually an action transfer is chosen and the appropriate action structure is added to the target set.

When an action structure is added to the target episode, it is automatically sent to the Body, along with its canonical representation. Each and every Tool that is going to receive and process the action command expects a specific format, and so a canonical message structure is needed. Those structures are kept in the Mind and are used when an action command is sent. Usually, they contain a sub-graph (always the same) that identifies the type of Tool to process the command, any additional information (as there might be Tools, carrying on various tasks) and the actual command.

Finally the user is provided with the result of the task and could give a feedback. If the task is considered completed (e. g. the question is answered and the user is satisfied with it) the whole episode with the task and its completion is stored in LTM as an experience episode. Any new knowledge acquired in the scope of the current task is isolated as general knowledge. In the current implementation, the user has two buttons in the interface – " praise" and "scold". If there is no button pressed after the answer is provided by the agent, the system assumes the user was satisfied and records the episode as successful. If the user presses the "praise" button, the episode is recorded as "more than successful" and if the user presses the "scold" button, the system records the episode as not-successful. In the latter case, this episode is recorded in the WM with very low chance of being retrieved in the future and as a negative example.

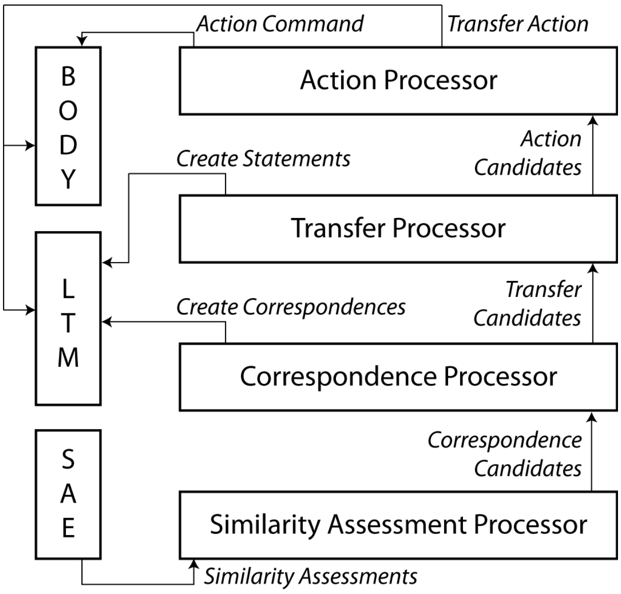

Fig 3: Reasoning Engine Modules.

III RE modules

The RE is made of several procedures for handling incoming knowledge structures and with several modules that are called "processors." The latter are used asynchronously, serially and iteratively within the scope of a single task. They are briefly described here, as the implementation details are not so important for the model and as they are still subject of changes, assessment and fine-tuning.

The modules of RE are as follows:

- Similarity Assessment Processor: handles the initial similarity assessments from SAE, eliminates the inconsistency and establishes correspondence hypotheses (CH). The latter are sent to the Correspondence Processor. If there are no correspondences established, the module aborts execution of the current reasoning process and invokes again the SAE;
- Correspondence Processor: the correspondence hypotheses are processed by this module. Each correspondence hypothesis has a score assigned on the basis of the activation of the nodes it puts into correspondence and of its consistency with other correspondence hypotheses. The most active correspondence hypotheses are used as a base for the formation of a list of transfer requests that are sent to the Transfer Processor. Before this, the correspondence hypotheses are checked against the contents of the LTM (fast, low-cost and efficient process). It is still a point of research if the check against the LTM contents is more efficient if done only in SAP, only in the Correspondence Processor, or in both of them;
- Transfer Processor: removes inconsistent transfer requests, deals with contradictory ones and evaluates them on the basis of the activation of the corresponding nodes and relatedness to the current goal. This is the module that actually adds transferred knowledge to WM. This module also creates a list of action requests, which are sent to the Action Processor;
- Action Processor: receives a list of possible actions to execute from the Transfer Processor. Contradictory actions are evaluated, based on their activation and the one with the highest activation is executed. If the last action has proven unsuccessful (e.g. no answer from an information search tool after a fixed amount of time).

IV Inference Engine (IE)

The RDF-based agent LTM provides the permanent storage of the agent knowledge. It is updated during operation of the model and can scale up to hundreds of thousands of statements. The current working prototype uses a LTM of less than ten thousand statements. The knowledge is expressed entirely with RDF triples and the model has mediated connection to the environment, as it receives and sends (to the 'Body') coalitions of RDF triples.

As mentioned above, the RDF representation provides the basis for extremely efficient reasoning-based (both by inference and analogy) augmentation of the goal and the most active part of the WM (which has the focus of attention of the agent). This allows the system to "transfer" specific and relevant information at very high speeds. The inference capabilities and the RDF storage as a whole are also used for verification and consistency check by the memory retrieval and transfer of knowledge for task completion purposes (e.g. answering a user question). The episodes that are stored in the LTM are identified by the contexts of the statements. As one statement can be part of more than one context, there are nesting and overlapping contexts, allowing flexible behavior and less data multiplication.

The IE uses mechanisms which are used by many platforms. However the novelty here is the combination with connectionist activation spreading which selects the knowledge in LTM relevant to the task at hand. This mechanism is inherited from DUAL/AMBR [1] but, as explained bellow, is further developed to include an attentional focus and make specific use of inference based memory augmentation.

V Conclusion

In this paper, the progress in the development of a new model for an embodied conversational agent is presented. Only the main ideas behind the model, its basic modules and the interplay among them are presented, although the largest part of the mechanisms have been implemented. The main focus was to outline the architecture of the model, the basic mechanisms allowing to increase its efficiency and the role of the reasoning engine in the interplay between the engines.

The main idea behind this model is to combine a rich cognitive model which would bring flexibility, context sensitivity, and selective attentional mechanisms with cutting edge fast machine learning algorithms like logical inference over ontologies and fast matrix calculations. At the core of the model is the combination of an connectionist similarity assessment and emotion modules which run continuously in parallel and a serial reasoning engine, which is based on inference over RDF triples. The main principle behind the connectionist engine is to combine activation spreading and semantic relevance and relational information in order to focus any further operations only on the most useful part of LTM. This fast selection of only a small part the agent's knowledge allow for its further augmentation by efficient inferences.

The three main 'engines' of the model – the reasoning, the emotional, and the similarity assessment engines – have been already implemented and parts of them tested. Although the results are very promising they are too preliminary to be reported here. The elaboration of the model, its full integration within a full fledged agent platform and tests with real users are currently in progress and will be the subject of a next papers. One of the latter will be focused on the detailed presentation and tests of the SAE and the other on the role of the EE as evidenced by simulations and usability tests.

Acknowledgment

We would like to gratefully acknowledge the discussions with Dr. Hristo Laftchiev, Kiril Kiryazov, Atanas Kyriakov, and Damian Ognianoff about the presented model.
The work is supported by the project RASCALLI under FP6 of EC.

References

[1] B. Kokinov, "A hybrid model of reasoning by analogy," in K. Holyoak and J. Barnden (Eds.), *Advances in connectionist and neural computation theory: Vol. 2. Analogical connections,* Norwood, NJ: Ablex, 1994, pp. 247 – 18.

[2] K. Kiryazov, G. Petkov, M. Grinberg, B. Kokinov, and C. Balkenius, "The Interplay of Analogy-Making with Active Vision and Motor Control in Anticipatory Robots," *Anticipatory Behavior in Adaptive*

Learning Systems: From Brains to Individual and Social Behavior, LNAI 4520, 2007.

[3] S. Kostadinov, G. Petkov, and M. Grinberg, "Embodied conversational agent based on the DUAL cognitive architecture," in *Proc. of WEBIST 2008 International Conference on Web Information Systems and Technologies*, Madeira, Portugal.

[4] S. Kostadinov and M. Grinberg, "The Embodiment of a DUAL/AMBR Based Cognitive Model in the RASCALLI Multi-Agent Platform," in *Proc. 8th International Conference on Intelligent Virtual Agents*, Tokyo, LNCS 5208, 2008, pp. 35–363.

[5] C. Becker, S. Kopp, and I. Wachsmuth, "Why emotions should be integrated into conversational agents," in *T. Nishida (Ed.), Conversational Informatics: An Engineering Approach*, Chichester: John Wiley & Sons, 2007, pp. 49–68.

[6] A. Kiryakov, D. Ognyanoff and D. Manov, " OWLIM – A Pragmatic Semantic Repository for OWL," in *Proc. Information Systems Engineering – WISE 2005 Workshops* , LNCS 3807, pp. 182 – 192 .

[7] J. Dias and A. Paiva, "Feeling and Reasoning: A Computational Model for Emotional Characters," *Progress in Artificial Intelligence*, Berlin, Springer, 2005.

[8] J. Dias, K. Kiryazov, A. Paiva, M. Grinberg, and S. Kostadinov, "Integration of Emotional Mechanisms in the Triple Agent Model,." In preparation.

[9] A. Ortony, G. Clore, and A. Collins, "*The Cognitive Structure of Emotions,"* Cambridge University Press, UK, 1988.

[10] I. Vankov, K. Kiryazov, and M. Grinberg, " Impact of emotions on an analogy-making robot," in *Proceedings of CogSci 2008,* Washington DC, July 22–26.

[11] A. Kabbaj, "Development of Intelligent Systems and Multi-Agents Systems with Amine Platform", in Proc. *ICCS 2006*, LNCS 4068, pp. 286-299.

[12] A. Kabbaj et al., "Ontology in Amine Platform: Structures and Processes", in Proc. *ICCS 2006* , LNCS 4068, pp. 300-313.

[13] J. Cassell, "Embodied conversational agents: representation and intelligence in user interfaces," in *AI Magazine archive. Volume 22, Issue 4,* pp. 67–8, 2001.

[14] N. Le β mann, S. Kopp, and I. Wachsmuth, "Situa ted interaction with a virtual human perception, action, and cognition," in G. Rickheit and I. Wachsmuth (Eds.), Situated Communication, Berlin: Mouton de Gruyter, 2006, pp. 287–323.

[15] B. Krenn, "RASCALLI. Responsive Artificial Situated Cognitive Agents Living and Learning on the Internet", in *Proc. of the International Conference on Cognitive Systems (CogSys 2008)*, LNCS 4131, pp. 535-542.

[16] NEBULA Engine (2007-2008), http://www.radonlabs.de/technologynebula2.html

[17] Feiyu Xu, H. Uszkoreit, Hong Li, "A Seed-driven Bottom-up Machine Learning Framework for Extracting Relations of Various Complexity," in *Proc. of ACL 2007,* Prague.

[18] MARY Text-to-Speech Engine (2008) – http://mary.dfki.de .

[19] M. Ramscar and D. Yarlett, "Semantic grounding in models of analogy: an environmental approach," Cognitive Science 27, 2003, pp. 41 – 71.

[20] L. B. Larkey and A. B. Markman, "Processes of similarity judgment", in *Cognitive Science 29, 2005*, pp. 1061–1075.

[21] R. Lopez de Mantaras, D. McSherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M. L. Maher, M. Cox, K. Forbus, M. Keane, A. Aamodt, I. Watson, "Retrieval, Reuse, Revision, and Retention in CBR", in *Knowledge Engineering Review, 20(3), 2005*, pp. 215–240.

[22] K. A. Ericsson and W. Kintsch, "Long-term working memory," *PsychologicalReview*, v. 102, 1995, pp. 211 – 245.

[23] W. Kintsch, V. L. Patel and K. A. Ericsson, " The role of long-term working memory in text comprehension," *Psychologia*, v. 42, 1999, pp. 186-198.