Chapter X A Multi-Level Time-Committed Framework for Evolving Agents

Stefania Costantini, Pierangelo Dell'Acqua

and Luís Moniz Pereira

Introspection is understood in philosophy (cf., Gertler, 2003; Fieser & Dowden, 2009) as a conscious mental and usually purposive process aimed at the self-observation and reporting of conscious inner thoughts. Introspection relies upon thinking, reasoning, and examining one's own thoughts, and is contrasted with extrospection, the observation of things external to one's self. Introspection is in general used synonymously with self-reflection.

In Computer Science, reflection is usually understood as the process by which a computer program can observe and modify its own structure and behavior. Reflection, as discussed in (Costantini, 2002) can take a variety of forms according to the underlying programming paradigm, but in essence it can be seen as the act of introspecting (*upward reflection*) and, symmetrically, of resuming normal operation (*downward reflection*). In Computer Science there is instead no commonly-agreed definition of introspection. It is interesting to notice that philosophy provides, since Locke, an "observational model" of introspection where introspective capacity enables us to observe the inner world just as perceptual capacity enables us to observe the outer world. In analogy to philosophy, in this paper we shall understand introspection as the ability to create and access a

representation of the object-level knowledge and reasoning processes and to be able to perform metareasoning based on this representation that can be considered as a (partial) representation of the *self*.

According to (Anderson & Oates, 2007), recent work on metareasoning mainly concerns on the one hand scheduling and control of deliberation, and on the other hand generating and using higher-order knowledge about (abstractions of) reasoning processes. In past work we have explored the potential of metareasoning to enhance the expressivity of the object-level (or, synonymously, *base-level*) reasoning. (Costantini & Lanzarone, 1989) and (Barklund, Costantini, Dell'Acqua & Lanzarone, 2000) discuss the use of logical schemata to capture basic properties of the domain represented in an object-level language. This results in a meta-level abstraction of the object-level domain knowledge, which allows for greater expressivity and increased inferential power in the system as a whole. The approach has been extended in (Costantini et al., 2004b) to the meta-level enhancement of agents' social ability, by means of metareasoning on the structure and contents of incoming and outgoing messages.

This subject of this paper is how to realize by means of introspection forms of metacognition aimed at making an agent eager to check and correct its own behavior with respect to inadequacies related either to the expected results and performance or to changes in the domain. The concept of metacognition as introduced in (Flavell, 1979) and (Flavell, 1987) concerns the knowledge (i.e., awareness) of one's cognitive processes and the efficient use of this self-awareness to self-regulate these processes. In the present setting, we take the broader view proposed in (Cox, 2005), where metacognition encompasses reasoning about one's own thinking, memory and executive processes (i.e.,

not just about one own learning process, like in the original definition). As emphasized by Cox, metacognition differs from standard cognition in that the self is the referent of the processing or the knowledge.

Thus, while metacognition may have many diverse definitions, we consider interpreting it as the monitoring and regulation of cognitive processes. This may for example (Maheswaran & Szekely, 2007) consist in choosing between multiple reasoning strategies and determining the amount of resources (computation, memory, bandwidth, etc.) to use in pursuing each strategy. We will instead interpret the monitoring as self-observation over time and self-modification in case anomalies are detected, with the aim to restore correct functioning. Self-modification is intended here as advocated by Cox (Chapter 6), i.e., not in the form of self-modifying "spaghetti" code, but in the form of adaptive changes driven by metareasoning.

When should metareasoning come into play, and how should these modifications be performed? The problem that (according to Anderson & Oates, 2007 and many others) metareasoning principally tries to cope with is the brittleness of most artificial systems in contrast with the robustness of natural intelligent systems. We would like to design and build machines that, like humans, may occasionally not behave optimally but are able to cope with unwanted and unforeseen difficulties and changes (which first of all implies being able to detect these changes). Anderson & Perlis (2005) define *perturbation tolerance* as the ability of a system to quickly recover—that is, to re-establish desired/expected performance levels—after a perturbation, intended as any change, either in the world or in the system itself, that impacts performance. The method they propose is to equip artificial agents with the ability to notice when something is amiss, assess the

anomaly, and guide a solution into place by means of the ‘metacognitive loop’ (MCL) which involves the system actually monitoring, reasoning about, and, when necessary, altering its own decision-making components. In their approach, time plays a central role: in fact, the formal basis is active logic, in which aspects of the environment are represented as first order formulas but with an "evolving-during-inference model of time" in contrast to the (as they say) “time-frozen” characterization of time of traditional temporal logic (c.f. for instance Rescher & Urquhart, 1971).

We take a similar stance concerning the central role of metareasoning for perturbation-tolerance and about the importance of time. However, we argue that humans cope with perturbations by wondering, more of less often, about the state of affairs of the object-level reasoning. If performance is unsatisfactory or unexpected circumstances have occurred that make the present course of reasoning and acting inadequate, humans take measures to either *improve* or *correct* their behavior, so as to restore a satisfactory situation. This may imply changing their view and their manner of considering certain aspects of the domain at hand. Therefore we propose and discuss in this paper an approach, not in contrast but complementary with others approaches to metareasoning, where metareasoning patterns come into play at a certain frequency rather than within a pre-defined loop. Introspection may occur more or less often depending upon the criticality of the conditions that each pattern copes with. Defining this frequency is a part of the initial setting of an agent, and frequencies can be modified dynamically as a result of reinforcement learning. Metareasoning schemata include temporal-logic-like statements, where a well-known agent-oriented temporal logic is adapted to perform dynamic self-checking. Our approach is strongly based upon *episodic memory* (defined in

Tulving, 1972 and advocated in Cox, 2007) which concerns actual events or episodes in a person's life or in an agent's action history. As discussed later on, we assume that episodic memory includes the records of what has happened, together with the annotation of when it happened, and possibly of the context where it happened.

Background

In this section, we shortly summarize the notions of meta-language, metareasoning, naming, introspection and reflection in a logical setting. The knowledgeable reader may skip this part.

Metareasoning in logic

In logic, a language that takes sentences of another language as its objects of discourse is called a meta-language. The other is called the object language. Sentences written in the meta-language can refer to sentences written in the object language only by means of some kind of *description,* or *encoding,* so that sentences written in the object language are treated as data. As widely discussed in (Carlucci Aiello & Levi, 1984) , (Barklund et al., 2000) and (Costantini, 2002) and in the references therein, object-level reasoning manipulates elements of the domain of interest in which the agent operates: in logic, this means that constants denote elements of the domain, variables range over elements of the domain, and predicates manipulate these constants and variables. Meta-level reasoning, which is reasoning about reasoning, can take as its objects representations of object-level expressions, including formulas or entire theories. In the syntax of logic, this means that

variables (i.e., meta-variables) may range over (encodings of) object-level predicates and formulas, and meta-level predicates and formulas are defined and operate over these representations. In principle, we may have a meta-meta-language, and so on, which means that a logical theory can be syntactically and conceptually organized into an object (or *base*) level and any number of meta-levels.

We may notice that most software components implicitly incorporate some kind of metaknowledge and metareasoning: there are pieces of object-level code that "execute" what metaknowledge states. For instance, an object-level planner program might "know" (as implicit "knowledge" hidden in the code) that *near(b,a)* holds whenever *near(a,b)* holds, while this is not the case for *on(a,b)*. A planner with a meta-level could explicitly encode: (i) a meta-rule stating that whenever a relation $\mathcal{R}$ is symmetric, then $\mathcal{R}(a,b)$ is equivalent to $\mathcal{R}(b,a)$ and whenever instead a relation is antisymmetric this is never the case; (ii) a form of metareasoning that establishes that these properties can be applied to *any* symmetric and antisymmetric relation that one may have. Thus, if the planner knows that *near* is symmetric and *on* is antisymmetric the desired result is obtained, with the advantage that such a declarative meta-level specification (which is independent of the specific object-level knowledge or application domain) is fairly general and might be reused in future applications.

Connection among levels and encoding

A link must exist to correlate the levels, in one direction for producing meta-level representation and, in the other direction, to transpose a meta-level result to the object-level and put it in operation. Two kinds of organization are possible. One option is that

each meta-level continuously monitors the lower levels and makes interventions whenever needed. An alternative option instead is that control is exchanged among levels, for example as follows.

- Control will *shift up* from one level to the level above it periodically and/or upon certain conditions, by means of an act called *upward reflection*.

- Vice versa, control will *shift down* to the lower level by performing an act called *downward reflection*.

Forms of control based on reflection in computational logic are formally accounted for in (Barklund et al., 2000). The frequency as well as the conditions of each type of shift is control information that must somehow be associated to the logical theory. Conceptually, reflection acts initiate a phase of *introspection*, as the reasoning activity at each level is suspended so as to reason, at a higher level, about how that reasoning activity is going on and what can be done to affect/improve it.

Meta-level rules manipulate a *representation* of lower-level knowledge which is often called a *name*. A name of a language expression is built by means of an operation of *encoding* (or *reification* or *quoting*) that must have an inverse operation giving back the original expression. Names can be of several forms, ranging from a simple constant to a compound formula describing at various levels of detail the expression it refers to. Meta-variables range over expressions involving names (for a discussion if these issues see Barklund et al., 1995).

Meta-level Architecture

In the remainder of this paper, we assume a multi-layered or multi-level architecture where there is a base (or object) level, that we call BA for "Base Agent", and (at least) two meta-layers. Thus, referring to Figure 11, from the Manifesto (Cox & Raja, 2007), while the BA monitors the "ground level", the MA (Meta-Agent) and the IEA (Information Exchange Agent) comprise in our model the "Meta-level". We do not commit to specific agent languages or models: our sole (soft) commitment is to the adoption of computational logic. Many existing agent models and systems can be adopted as "building blocks", and even commitment to computational logic is not really strict.

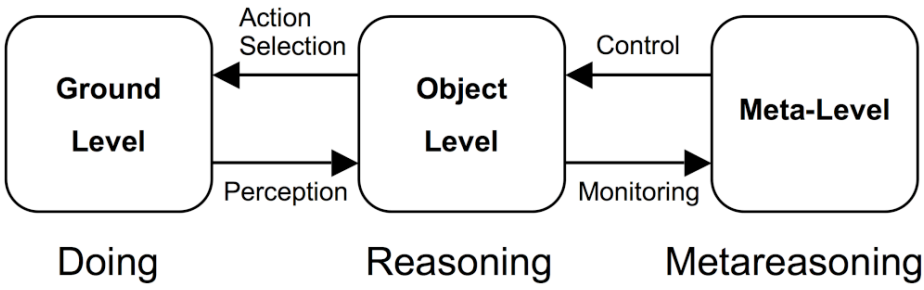

Figure 1: Multi-layered agent model

The MA performs metareasoning of various kinds and supervises the BA activities. The MA includes a meta-control component that on the one hand coordinates the BA activities, and on the other hand makes the MA decisions effective by acting on the BA. The actions that the MA will be able to undertake may include the modification of the BA in terms of adding/removing knowledge (modules) in the attempt at correcting inadequacies and generating a more appropriate behavior. The IEA will be put into action

whenever the need and opportunity of social interactions and knowledge exchange requires evaluation. The agent model that we have in mind and its operational account is discussed in (Costantini, Tocchio, Toni & Tsintza, 2007).

The role of Memory

As emphasized in (Cox, 2005), the role of memory and more generally that of an internal "history" is crucial for metareasoning and metacognition in agents. Through "memory" the agent is potentially able to learn from experiences and ground what it knows throughout these experiences (Liew & Gero, 2002a). The interaction between the agent and the environment can play an important role in constructing its "memory" and may affect its future behavior. Methods to design agent memorization mechanisms have often been inspired by models of human memory (see for instance Logie, 1995 and Pearson & Logie, 2000).

Atkinson and Shiffrin (1968) proposed a model of human memory which posits two distinct memory stores: short-term and long-term memory. This model has been adopted by Gero and Liew for implementing constructive memory (Liew & Gero, 2002b). Memory construction in this model occurs whenever an agent uses past experiences in the current environment in a situated manner. In a constructive memory system, any information about the current design environment, the internal state of the agent and the interactions between the agent and the environment is used as cues in its memory construction process.

Gero and Liew use the notion of working memory, a workspace for reflective and reactive processes where explicit design-based reasoning occurs. Items of information within the working memory are combined with the stored knowledge and experiences, manipulated, interpreted and recombined to develop new knowledge, assist learning, form goals, and support interaction with the external environment.

(Costantini & Tocchio 2006c) propose a method of correlating agent experience and knowledge by using a particular construct, the internal events, that has been introduced in the logic agent-oriented language DALI (Costantini & Tocchio 2004a, 2006a, 2006b, Costantini et al., 2004b) but could be in principle adopted in any computational logic setting. The agent memory in defined in a very simple way as composed of the original knowledge base augmented with *past events* that record the external stimuli perceived, the internal conclusions reached and the actions performed.

Past events can play a role in reaching internal conclusions. These conclusions, that are proactively pursued, take the role of "dynamic" memory that supports decision-making and actions: in fact, an agent can inspect its own state and its view of the present state of the world, so as to identify the better behavioral strategy in that moment. The agent re-elaboration of its experiences creates a particular view of the external world. By "particular" we mean that each agent, on the basis of its knowledge and experience, can interpret what has changed in the world in its peculiar manner. Therefore, an agent should record not only perceived external stimuli but also the internal conclusions reached by the entity and the actions performed. This allows in principle all aspects of agent behavior to be inter-related, thus potentially improving its capabilities and performance.

More specifically, we believe that *past events* may have at least three relevant roles:

- Describe the agent experience: if agent's actions and commitments are recorded in the set of past events with the annotation of when (and possibly in which context) they were performed, their sequence and significance can be reconstructed.
- Keep track of the state of the world and of its changes, possibly due to agent intervention.
- Induce and install preference rules and meta-rules about conflicting preferences (Pereira et al., 2009).

Notice that in case there are more "versions" of a past event, the last one will now be in force or "actual". Then, all the most recent past events may be seen as representing the current "state of affairs".

We presume to keep a set P of current "valid" past events that describe the state of the world as perceived by the agent but we also presume to maintain a set PNV where we store all previous ones. The content of set PNV can be seen as the agent "memory" in a broader sense (we do not cope here with practical efficiency reasons that might force the agent to "purge" PNV in some way so as to regain store). We assume a mechanism for keeping P up-to-date, which consists in defining in the agent program a set of constraints that express what and when to remove in a past event from P.

Self-Checking via Metareasoning

Metareasoning can be related to several aspects of agent operation (for example learning, negotiation, goal evaluation and setting). The quest for metareasoning is not only due to limited computational power, but also to the agents' *autonomy,* i.e., to the ability of agents to act without user intervention and to inhabit an open, changing and partially known environment. Autonomous agents should hopefully improve their performance and "competence" over time, and this can hardly be obtained without resorting to metareasoning.

Methods for checking that an agent will exhibit a correct behavior during its "life" are sometimes intended to be applied at the initial phase of agent operation. But agents live in open environments where events happen including interactions with other agents. Moreover, agents can learn new knowledge or rules. This makes the checking of behavior correctness by means of either model-checking (cf. Clarke & Lerda, 2007) or other static 'a priori' approaches quite impractical. However, metareasoning can be usefully exploited during operation to check that agent's activities are performed in a correct and timely manner (i.e., poor performance can be itself considered to be an anomaly). In case a malfunctioning or some kind of inappropriate/inadequate behavior is detected, suitable deliberations can be taken that may include stopping/starting object-level activities and/or replacing pieces of object-level code. Therefore, in our view the MA coordinates BA's operation as it triggers appropriate activities at the appropriate time, puts metareasoning deliberations into effect, but also continuously performs self-checking based upon historical information and takes appropriate corrective measures in case of anomalies.

Self-checking by means of metareasoning relies in general upon records that should be collected and maintained by the agent itself. These records (that we call, as discussed, *past events*) concern what has happened in the past to the agent (events perceived, conclusions reached, actions performed, new knowledge learned) and encode relevant aspects of an agent's *experience* that can be seen as a component of a description of the agent's *self*.

Given the variety of unforeseen circumstances, agents may performed better if their representation of the self is empowered with updatable overarching moral principles that specify general normative decision-making and behavior (Pereira & Saptawijaya, 2007, 2008). These principles can be expressed at the meta-level, and their respect can be again based on self-checking.

Agents may also model at the meta-level their hypothetical futures, in order to determine by means of metareasoning the best courses of evolution from their own present, and thence to prefer amongst those futures. In such 'prospective' agents (Pereira & Lopes, 2007, 2008 and Pereira & Anh, 2009), a priori and a posteriori preferences, embedded in meta-reasoning patterns, are used for preferring amongst hypothetical futures, or scenarios. The a priori ones are employed to produce the most interesting or relevant conjectures about possible future states, while the a posteriori ones allow the agent to actually make choices based on the imagined consequences in each scenario. Such agents need to be able to self-evaluate consequences of their decisions, based on the historical information as well as quantitative and qualitative a posteriori evaluation.

In our approach self-checking activities come into play by means of reflection as introspective activities. Therefore, in our view these activities observe and influence aspects of the agent's *self*. In fact, excluding self-consciousness, which is not discussed here, meta-axioms describe what the agent can answer to itself when asking "How can I try to understand my own behavior?", or, "Am I doing things properly or not?". Moreover, though for lack of space we cannot discuss this issue here, we assume a learning mechanism by which the agent learns not only object-level facts and rules but also meta-axioms, thus enlarging its capabilities and potentials, provided that the acquired knowledge is evaluated during a trial period (Costantini et al., 2008b).

Temporal-Logic-like Metareasoning Rules

We agree with Anderson & Perlis (2005) on the fact that time must be explicitly taken into account in meta-reasoning and metacognition. Therefore, in our perspective the MA will include self-checking rules inspired by temporal logic. We also agree on the need of an "evolving-during-inference" rather than "time-frozen" model of time. In fact, our rules are constructed so as to be re-evaluated over time and adapted to a changing context. The basic aim of the checks is the detection of either fulfillment or violations of time-dependent constraints that have to be worked out by some action of *improvement*.

These actions cannot be decided "a priori" since they will depend on the context. In order to be able to express in a time-dependent fashion meta-rules aimed at self-checking, we have introduced the AI-METATEM language, which builds upon and extends the

well-known METATEM temporal logic (Fisher, 2005). Operationally, AI-METATEM relies on a mechanism similar to that of the DALI internal events, thus losing the full power of modal logic though retaining a number of its useful features with a reasonable computational efficiency. For a formal definition the reader may refer to (Costantini et al, 2008a). Here we illustrate the kind of self-examination and self-improvement that can be performed.

For instance, the following is an AI-METATEM meta-rule. The atom before the "*:*" is the *name* of this meta-axiom. The part between the "*:*" and the "*::*" is the *check condition*. The part between the "*::*" and the "÷" is the *context*. The part after the "÷" is the *repair/improvement*.

Incr-commitment(G,T): N–NEVER(not achieved(G),dropped(G))::

goal(G), NOW(T), commitment_level(T,L) ÷

increment_commitmemt(T,L,New_L)

Suppose that at a certain time *t* the check condition

N-NEVER(not achieved(G),dropped(G))

holds for some specific goal *g*, where *N–NEVER* stands for "Not never", i.e., "Sometimes". This means that goal *g* has been dropped without having been achieved.

The context part on the one hand instantiates the meta-variable *G* to appropriate values, on the other hand sets the current time *t* and determines the current commitment

level. Then, the MA executes the action after the "÷", which consists in increasing the commitment level adopted by the BA. The execution of the action (that can be seen on the one hand as a repair of a malfunctioning, on the other as an improvement over current agent behavior) will in general allow the MA to perform the specified run-time re-arrangement of the program, thus attempting to cope with the unwanted situation. Finally, the rule name instantiated with the values at hand will be recorded as a part event. Here, the past event is *Incr_commitmentP(g,t)*.

The recording of past events allows AI_METATEM meta-axioms to be defined on these records. For instance, the following meta-axiom will have the effect of increasing the trust evaluation of each agent *A* that has been observed to be reliable during an interval.

Reliable_agentP(A,T1,T2) ÷ Increase_Trust_Level(A).

The monitoring has been performed by another meta-axiom, that if successful for specific agent *a* in interval *(t1,t2)* has led to the record *Reliable_agentP(a,t1,t2)*.

AI-METATEM axioms are activated by an act of introspection that shifts control from the MA to the BA. Introspection may occurs either periodically or also upon certain conditions. For instance, the axiom in the latter example above could be attempted periodically, while the former whenever a *drop* action is performed by the BA. The frequency or the conditions can be associated to each meta-axiom as control information.

As discussed in (Costantini et al, 2008a), AI-METATEM meta-level axioms can be useful for many forms of self-checking and self-evaluation. The repair/improvement action may include self-modification in the sense of replacing some of the agent

component modules with others that are deemed more appropriate in the present context. AI_METATEM meta-level axioms can also be very useful for performing metacognition in the strict sense, i.e., for monitoring the agent learning processes.

Concluding Remarks

In (Alexander et al., 2007) and (Raja & Lesser, 2007) and in the references therein, meta-level control is defined as the ability of complex agents operating in open environments to sequence domain and deliberative actions so as to optimize expected performance. Deliberative control actions may include for instance scheduling domain-level actions and coordinating with other agents to complete tasks requiring joint effort.

The meta-control component is triggered based on the dynamic occurrence of pre-defined conditions. Meta-control generates an abstract meta-level problem based on the agent's current problem solving context and available deliberative actions, and chooses the appropriate deliberative action which is executed, possibly resulting in further domain-level actions. In their view, meta-level control supports for example decisions on when to accept, delay, or reject a new task; when it is appropriate to negotiate with another agent; whether to renegotiate when a negotiation task fails; how much effort to put into scheduling when reasoning about a new task; and whether to reschedule when actual execution performance deviates from expected performance. In their view, meta-level control can be understood as the process of deciding how to interleave domain and deliberative control actions such that the agent functioning is correct and appropriate also in terms of resource usage. Each agent has its own meta-level control component.

Our work can be intended as complementary rather than alternative, as we propose a method for "local" self-adaptation to perturbations by means of periodical introspection activities. This can be in principle integrated with their more "global" meta-level control.

We have experimented the practical application of temporal-logic like metareasoning rules by adapting the implementation of internal events in DALI. The first experiments are promising, as they show that introducing these meta-rules improves the agents' behavior in accuracy and adequacy to their tasks and also in performance.

References

Anderson, M. L., & Perlis, D. R. (2005). Logic, Self-Awareness and Self-Improvement: The Metacognitive Loop and the Problem of Brittleness. *Journal of Logic and Computation*, *15(1)*, 21-40.

Alexander, G., Raja, A., Durfee, E., & Musliner, D. (2007). Design paradigms for meta-control in multi-agent systems. In *Proceedings of AAMAS 2007 Workshop on Metareasoning in Agent-based Systems* (pp. 92–103).

Anderson, M., & Oates, T. (2007). A Review of Recent Research in Metareasoning and Metalearning. *AI Magazine, 28(1),* 7-16.

Atkinson, R.C, & Shiffrin, R.M. (1968). *Human memory: A proposed system and its control processes*. New York, NY, Academic Press.

Barklund, J., Costantini, S., Dell'Acqua, P., & Lanzarone, G.A. (1995). Semantical properties of encodings in logic programming. In Lloyd, J.W. (Ed.) *Logic Programming – Proc. of the 1995 International Symposium* (pp. 288-302), Cambridge, MA, MIT Press.

Barklund, J., Costantini, S., Dell'Acqua, P., & Lanzarone, G.A. (2000). Reflection principles in computational logic. *J. of Logic and Computation, 10(6)*, 743–786.

Carlucci Aiello, L., & Levi, G. (1984). The uses of metaknowledge in AI systems. In: *Proc. European Conf. on Artificial Intelligence* (pp. 705–717).

Clarke, E.M. & Lerda, F. (2007). Model Checking: Software and Beyond. *Journal of Universal Computer Science, 13(5),* 639–649.

Costantini, S. (2002). Metareasoning: a survey. In Kakas, A.C, & Sadri, F. (Eds.). *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Lecture Notes in Artificial Intelligence* (pp. 2407-2408). Berlin, Germany, Springer-Verlag.

Costantini, S., & Lanzarone, G. A. (1989). A Metalogic Programming Language. In G.Levi & M. Martelli (Eds.), *Logic Programming, Proc. of the 6th Int. Conf., The MIT Press*, USA, 1989.

Costantini, S., & Tocchio, A. (2004a). A Logic Programming Language for Multi-agent Systems. In S. Flesca, S. Greco, N. Leone, & G. Ianni (Eds.), *Logics in Artificial*

Intelligence, Proc. of the 8th Europ. Conf., JELIA 2002, LNAI 2424, Berlin, Germany, Springer-Verlag.

Costantini, S., Tocchio, A., & Verticchio, A. (2004b). Communication Architecture in the DALI Logic Programming Agent-Oriented Language. In *Proceedings of CILC'04, Italian Conference on Computational Logic.* http://www.cs.unipr.it/CILC04/

Costantini, S., & Tocchio, A. (2006a). The DALI Logic Programming Agent-Oriented Language. In J. J. Alferese & J. Leite (Eds.), *Logics in Artificial Intelligence, Proceedings of the 9th European Conference, Jelia 2004, LNAI 3229*, Berlin, Germany, Springer-Verlag.

Costantini, S., & Tocchio, A. (2006b). About declarative semantics of logic-based agent languages. In Baldoni, M., Torroni, P. (Eds.), *Declarative Agent Languages and Technologies, LNAI 3229*. Berlin, Germany, Springer-Verlag.

Costantini, S., & Tocchio, A. (2006c). Memory-driven dynamic behavior checking in Logical Agents. In *Proceedings of CILC'06, Italian Conference of Computational Logic*. Bari, Italy: URL: http://cilc2006.di.uniba.it/programma.html.

Costantini, S., Tocchio, A., Toni, F., & Tsintza, P. (2007). A multi-layered general agent model. In: *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing, 10th Congress of the Italian Association for Artificial Intelligence. LNCS 4733*. Berlin, Germany: Springer-Verlag.

Costantini, S., Dell'Acqua, P., & Pereira, L.M. (2008a). A Multi-layer Framework for Evolving and Learning Agents. In Cox, M. & Raja, A. (Eds.), *Proceedings of the AAAI-08 Workshop on Metareasoning: Thinking about Thinking.*

Costantini, S., Dell'Acqua, P. Pereira, L.M., & Tsintza, P. (2008b). Specification and Dynamic Verification of Agent Properties. In Fisher, M., Sadri, F. & Thielscher, M. (Eds.), *Proceedings of the International Workshop on Computational Logic in Multi-Agent Systems (CLIMA-IX)* (pp. 61-76).

Cox, M. (2005). Metacognition in computation: A selected research review. *Artificial Intelligence*, *169(2)*, 104-141.

Cox, M. T., Raja, A. (2007). Metareasoning: a manifesto. *Technical Report BBN TM-2028, BBN Technologies*. URL www.mcox.org/Metareasoning/Manifesto.

Fieser, J. & Dowden, B. H. (2009) The Internet Encyclopedia of Philosophy, http://www.iep.utm.edu/.

Fisher, M. (2005). Metatem: The story so far. In Bordini, R.H., Dastani, M., Dix, J., & Fallah-Seghrouchni, A.E. (Eds.), *3rd Int. W. on Programming Multiagent Systems (PROMAS), LNAI 3862* (pp. 3-22), Berlin, Germany, Springer-Verlag.

Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive developmental inquiry. *American Psychologist*, *34*, 906–11.

Flavell, J. H. (1987). Speculations about the nature and development of metacognition. In F. E. Weinert & R. H. Kluwe (Eds.), *Metacognition, Motivation and Understanding* (pp. 21–9). Hillside, New Jersey, Lawrence Erlbaum Associates.

Gertler, B. (2003). Self-Knowledge entry in the Stanford Encyclopedia of Philosophy. http://plato.stanford.edu/entries/self-knowledge/

Liew, P-K., & Gero, J.S. (2002a). A memory system for a situated design agent based on constructive memory. In Eshaq, A., Khong, C., Neo, K., Neo, M., & Ahmad, S. (Eds.), *Proc. CAADRIA2002* (pp. 199-206), New York, Prentice Hall.

Liew, P-K., & Gero, J.S. (2002b). An implementation model of constructive memory for a design agent. In Gero, J.S. & Brazier, F. (Eds), *Proc. W. Agents in Design 2002* (pp. 257-276). Sydney, Australia: Key Centre of Design Computing and Cognition, University of Sydney, Australia.

Logie, R.H. (1995). *Visuo-Spatial Working Memory*. Hove, UK: Lawrence Erlbaum.

Maheswaran, R. T., & Szekely, P. Metacognition for Multi-Agent Systems, In *Proc. AAMAS W. on Meta-Reasoning in Agent-Based Systems,* Honolulu, HI, May 14, 2007.

Pearson, D.G., & Logie, R.H. (2000). Working memory and mental synthesis. In O'Nuallan, S. (Ed.), *Spatial Cognition: Foundations and applications.* John Benjamins Publishing Company.

Pereira, L.M., & Anh, H.T. (2009). Evolution Prospection. In Nakamatsu, K. et al. (Eds.), Procs. First KES Intl. Symposium on Intelligent Decision Technologies (KES-IDT'09), Studies in Computational Intelligence vol.199, (pp. 51-64), Berlin, Germany, Springer.

Pereira, L.M., & Lopes, G. (2007). Prospective Logic Agents. In Neves, J.M., Santos, M.F., Machado, J.M. (Eds.), *Progress in Artificial Intelligence, Proceedings of the 13th Portuguese International Conference on Artificial Intelligence (EPIA'07), LNAI 4874* (pp. 73-86), Berlin, Germany, Springer-Verlag.

Pereira, L.M., & Lopes, G. (2008). Prospective Logic Agents. *International Journal of Reasoning-based Intelligent Systems (IJRIS)*, to appear.

Pereira, L.M., & Saptawijaya, A. (2007). Modelling Morality with Prospective Logic. In Neves, J.M., Santos, M.F., Machado, J.M. (Eds.), *Progress in Artificial Intelligence, Proceedings of the 13th Portuguese International Conference on Artificial Intelligence (EPIA'07), LNAI 4874* (pp. 99-111), Berlin, Germany, Springer-Verlag.

Pereira, L. M., & Saptawijaya, A. (2008). Modelling morality with prospective logic. *International Journal of Reasoning-based Intelligent Systems (IJRIS)*, to appear.

Pereira, L.M, Lopes, G., & Dell'Acqua, P. (2009) On Preferring and Inspecting Abductive Models, Invited paper in Gill, A., & Swift, T. (Eds.), *Proceedings of the 11th Intlernational Symposium on Practical Aspects of Declarative Languages (PADL'09), LNCS*, Berlin, Germany: Springer-Verlag.

Raja, A., & Lesser, V. (2007). A framework for meta-level control in multi-agent systems. *Autonomous Agents and Multi-Agent Systems, 15(2),* 147–196.

Rescher, N., & Urquhart, A., (1971). *Temporal Logic*. New York, Springer-Verlag.

Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donaldson (Eds.), *Organization of memory*, (pp. 381-403). New York, Academic Press.