DALI, RASP, Mnemosine: la Logica Computazionale in azione
DALI, RASP, Mnemosine: Computational Logic at Work

Stefania Costantini, Alessio Paolucci, Arianna Tocchio, Panagiota Tsintza

SOMMARIO/*ABSTRACT*

In questo articolo presentiamo le linee di ricerca più recenti del gruppo di Intelligenza Artificiale dell'Universita di L'Aquila. Tali attività riguardano la Logica Computazionale e principalmente gli Agenti Intelligenti logici. Infatti, il gruppo ha sviluppato negli ultimi anni il linguaggio logico orientato agli agenti DALI. Tuttavia, vi sono attività anche in altre aree, come ad esempio il ragionamento non monotono e l'elaborazione del linguaggio naturale. L'attenzione è posta in particolare sulle nuove prospettive di lavoro.

In this paper, we briefly describe recent research directions of the Artificial Intelligence group of the University of L'Aquila, Italy. Research activities concern Computational Logic and mainly Intelligent Logical Agents. In fact, in the last years the group has developed the logical agent-oriented language DALI. However, work is under way also in other areas, like, e.g., Non-Monotonic Reasoning and Natural Language Processing. We particularly emphasize recent and future work directions.

Keywords: Intelligent Agents, Negotiation, Answer Set Programming, Natural Language Processing for the Web.

1 Introduction

Intelligent Agents, computational logic, agents cooperation and negotiation, non-monotonic reasoning, natural language processing and, very recently, biologically inspired models have been the key words of our research activity in the last years[1].

Intelligent agents in computational logic form the "core" of our research. A main achievement of our group has been the definition and development of the DALI language [10, 11, 17, 12], an Active Logic Programming language designed in the line of [15] for executable specification of logical agents. DALI is a prolog-like logic programming language with a prolog-like declarative and procedural semantics. The reactive, proactive and social behavior of DALI agents is triggered by several kinds of events: external, internal, present and past events. The DALI Interpreter has been fully implemented in Sicstus Prolog. DALI agents have been put at work in several real-world applications. An application where the role of DALI agents is particularly relevant has been developed in the context of the CUSPIS European project[2], where DALI agents have been adopted for supporting users during their visit to museums or archeological areas. The system has been practically demonstrated in Villa Adriana (Tivoli, Rome) [8]. However, we have also experimented DALI agents in the context of hybrid architectures and in negotiation scenarios, as summarized in Sections 2 and 3.

Another research line of the group is concerned with Answer Set Programming (ASP for short), which is a form of logic programming based on the answer set semantics [14], where solutions to a given problem are represented in terms of selected models (answer sets) of the corresponding logic program. We have recently proposed RASP, an extension of ASP that permits declarative specification and reasoning on consumption and production of resources, shortly presented in Section 4.

A recent research direction concerns the problem of improving natural language processing by means of parallelizing syntactic and semantic analysis. As a case-study, we have developed a prototype semantic search engine, called *Mnemosine*. This work is outlined in Section 5.

Finally, in the last period we have spent some effort (in cooperation with the biologists of our University) in the study of the human brain as, in view of the growing complexity of computational tasks and their design, many researchers are considering whether interactive systems may be better designed by exploiting computational strategies

[1] The list of publications of the research group can be found at the URL http://www.di.univaq.it/stefcost/pubbls_stefi.htm

[2] CUSPIS, "A Cultural Heritage Space Identification System"(GJU/05/2412/CTR/CUSPIS).

based on the understanding of the human brain. In fact, up to now artificial cognitive systems have been designed without any reference to biology. However, with the predicted increases in computational power and storage capacity over the next decades, it may be important to investigate whether the study of natural cognitive systems can lead to design artificial systems, and in particular agent architectures, with better cognitive capabilities. If so, agent architectures will evolve and agents will become a more effective and useful support for the human activities.

2 Modeling Intelligence in Multi-Layer Cooperative Systems

Nowadays, myriads of heavily networked devices interact with the physical world in multiple ways and at multiple scales. Many of these devices are highly mobile and must adapt to the surrounding environment in a totally unsupervised way.

Biological systems are able to handle many of these challenges with an effectiveness still far beyond current human artifacts: therefore, our long-term goal is to investigate biologically inspired methods on how to engineer intelligent agent systems, so as to exhibit similar high stability and efficiency.

For reconciliating scalability and intelligence, one way is that of creating systems where various degrees of intelligence are distributed over various levels of the architecture. Aim of our research work in this direction has been that of creating a society composed of high-level intelligent agents, aided in their tasks by bunches of elementary agents. The high-level agents are responsible of overall system strategies and plans, to be possibly devised in cooperation. Bunches of elementary agents are supposed to assist each high-level agent in activities where massive parallelism is in order, such as environment exploration, pattern-recognition, classification, action selection and action execution. In fact, as in social insects colonies intelligence emerges from the cooperation activities among individuals, in software environments "intelligence" can be the result of a distribution of roles among different kinds of agents.

We have developed an architecture based on DALI intelligent agents and IBM Aglet mobile agents [1], where the Aglets are Java objects that can move from one host on the Internet to another, and are equipped with communication capabilities: i.e., the Aglets are a very simple kind of agent. The integration of DALI and the Aglets into a colony has been made possible by exploiting those social aspects that are present in both agents platforms. Instead of using standard interfaces, we have implemented a more efficient and flexible communication level not tailored to specific formalisms, that will in perspective allow the integration of other agent platforms into the framework.

As a first experiment of this new kind of architecture, we have tried to model a colony of social insect, though we have widely reinterpreted its structure. We have tried to suitably exploit the features of each platform and to distribute roles among the various entities in order to obtain a kind of "social intelligence" in the artificial colony. Roles that require more "intelligence" are assigned to DALI agents due to their reasoning and learning abilities, while roles requiring communicative and reactive abilities are assigned to Aglets due to their mobility and social nature. In particular, proactivity of DALI agents allowed us to introduce in the artificial colony an entity capable of supervising all the activities which are crucial for the community life like, for example, the planning of some kind of supply or the generation of new individuals useful for the community. Reactivity and mobility of Aglets suggested that their ideal job was that of being the "actuators" of the basic steps of plans devised at the higher level.

We have experimented the architecture in two basic scenarios. In the first one, DALI agents are totally responsible of planning and communicate directly their directions to the actuators. In the second one, DALI agents delegate some planning activity to an intermediate entity that communicates with the actuators. For performing the experiments, we have simulated the activity of bees relative to honey production (where the "honey" the Aglets try to produce may in practical applications correspond to any kind of resource).

The difference between the two scenarios is mainly the degree of interaction, intelligence and sharing among the components. In the first scenario, all managerial and intelligent roles are reserved to the DALI "queen" that organizes the collectivity for producing a certain quantity of honey. According to the queen's indications and using the resources of the society, the colony starts moving, so that every member completes its task with the maximum efficiency. In the second scenario, we have introduced a new role inside the society, i.e., the "courtier". Here, the managerial role previously reserved only to the queen is distributed among the queen and the courtiers. Both scenarios show that the common effort of different entities such as the DALI intelligent agents and the mobile IBM Aglets succeeds in producing coordination so as to reach a common goal.

The number of experiments that we have performed so far does not allow us to establish a general statement. However, the second scenario seem to behave better in the sense that the same quantity of honey is produced in less time. In perspective however, we mean to assess by means of experiments which is the "optimal" distribution of intelligence among levels in relevant classes of applications.

Work is under way in the development of an application of this hybrid architecture in the field of security. As a first step, we have used the Aglets for exploring registries of a computer processor and detecting if an undesired program has been installed. We plan to tackle in the near future other security-related issues.

3 Agents and Negotiation

In the context of Proposal-Based negotiation we have proposed, implemented and experimented an extension of a negotiation approach originally introduced by Marco Cadoli [4].

In the original approach the negotiation areas, representing the admissible values of the negotiation issues, are considered to be convex ones. In particular, the negotiation process is considered as proposal-based with the restriction that at least one of the parts involved in the process is bound to offer only proposal corresponding to vertices of the negotiation area.

This restriction entails that at least one of the negotiation areas has to be polyhedral, while the restriction on the areas to be convex entails that any point of the line connecting two admissible proposals has to be an admissible proposal as well. Possible agreements are represented by the intersection of the two areas. The goal of the approach is to conclude the process, i.e., to find an agreement, by involving the minimum number of interaction between the parties.

The proposed extension [13] is based on relaxing the condition of at least one agent offering vertices of the the negotiation area, which on the one hand may lead to problems if, e.g., the intersection area does not include vertices and on the other hand excludes non-polyhedral areas such as circles. In the extension, proposals can be internal points of the negotiation areas. The points to propose are selected based on the last offer, increased or decreased by a delta margin. A large number of experiments have shown that the proposed extension works properly and that the algorithms performance, in terms of interactions, is reasonable.

Our recent research work on negotiation is related to Argumentation-Based negotiation and its use to cope with contract violations: an agent that has violated an already-signed contract will try to justify this fact by exposing some arguments while the opponent agent will try to undermine their truthfulness and acceptability, by finding attacks against them. As a response, the justifying agent needs in turn to devise a counter-attack. The main theoretical tools that we adopt are logic programming, argumentation and modal logics. Among the objectives of this research are the specification of an appropriate language to support/depict the arguments/justifications used by the agents (we are presently considering dynamic epistemic logic) and the definition of algorithms or mechanisms for performing dialectical disputes among agents.

4 Non-Monotonic Reasoning

Rich literature exists on applications of ASP in many areas, including problem solving, configuration, information integration, security analysis, agent systems, semantic web, and planning (see, e.g., [2] and the references therein). The ASP formulations in these and other fields may take profit from the possibility of performing (at least to some extent) forms of *quantitative* reasoning like those that are possible in, e.g., Linear Logics and Description Logics.

Then, together with Andrea Formisano (University of Perugia) we have recently proposed RASP, an extension of Answer Set Programming that allows for declarative specification and reasoning on consumption and production of resources. Resources are modeled by introducing *amount-atoms*, involving *quantities* that represent the available amount of resources. Processes that use resources are easily described through program rules: in fact, the firing of a RASP-rule can both consume and produce resources. Different solutions correspond to different possible allocations of available resources.

The approach also allows the declarative specification of preferences among alternative uses of available resources. In particular, in realizing the same process (modeled through the firing of a rule), one may prefer to produce a certain product rather than another one and/or to consume certain available resources rather than others. This extension can be particularly useful in planning/configuration applications.

Semantics for RASP programs is provided by combining usual answer set semantics with an interpretation of resource amounts, where different allocation choices correspond to different answer sets.

5 Mnemosine

Menmosine [9] is a prototype semantic search engine based on an extension to the well-known DCGs (Definite Clause Grammars) so as to perform syntactic and semantic analysis to some extent in parallel, and generate semantically-based description of the sentence at hand. The parallelization of syntactic and semantic analysis can help solving some functional deficiencies of classical NLP solutions [3, 16], that often cannot properly cope with ambiguous propositions. In fact, in classical automated language processing methodologies, syntactic and semantic analysis do not have, in many cases, sufficient information neither to determine with certainty the syntactic aspects and details nor to give the correct semantic evaluation. Even probabilistic methods have difficulties.

Thus, for coping with many practical cases NLP systems must at least include ontological reasoning, and thus must become to some extent “intelligent”. Mnemosine relies on an extension to classical DCG’s where a background knowledge base is accessed during the analysis, which is no more divided into separate stages, but can be considered to be “syntactic-semantic”. The input of the analysis is (as usual) a sequence of tokens obtained from lexical analysis. The results of syntactic-semantic analysis consist in (i) establishing the syntactic correctness of the sentence; (ii) creating a formal representation of extracted knowledge; (iii) adding to the knowledge base this representation, as well as the consequences that can be drawn form it. I.e.,

the objective is to elicit the structure and the meaning of the natural language expression at hand, and to properly exploit it for enlarging or improving the available knowledge.

Mnemosine has been fully implemented and has been applied to a practical case-study, i.e., to the WikiPedia Web pages. We have chosen to use a real data sets from third parties as the choice of data is of primary importance in experiments: in the field of artificial intelligence in fact, many solutions operate properly and efficiently on the data on which they have been developed and tested and then their efficiency collapses dramatically as soon as switched to a real operating environment.

The architecture of Mnemosine has been designed so as to be ready for a timely transformation from a research prototype to an actual industrial product.

6 Conclusions

Throughout the world we are seeing an increased interest in Artificial Intelligence and Computational Logic, despite the relative crisis of Computer Science "per se". Successful results in agents, search and language technology, robotics and web applications are starting the transition to industry. For all this, we have to thank those researchers, like Alberto Martelli, whose important work in computational logic has significantly contributed to the successful development of this field.

REFERENCES

[1] Aglets web site. http://www.trl.ibm.com/aglets/, 2008.

[2] C. Baral. Knowledge representation, reasoning and declarative problem solving. Cambridge University Press, 2003.

[3] P. Blackburn and J. Bos. Representation and Inference for Natural Language. *Natural Language, Center for the Study of Language and Information, StanfordUniversity*, Lecture Notes, 2005.

[4] M. Cadoli, Proposal-Based Negotiation in Convex Regions, *Proceedings of Cooperative Information Agents VII (CIA)*,p.93-108,2003.

[5] S. Costantini, M. S. Del Greco and A. Tocchio. Social bugs communities and Intelligent Agents: an experimental architecture. *Proc. of CILC'08, Ital. Conf. on Computational Logic, Perugia, Italy, July 2008.*

[6] S. Costantini and A. Formisano. Modeling resource production and consumption in answer set programming. Technical Report 04, Dipartimento di Matematica e Informatica, Univertità di Perugia, 2008. Preliminary version in *Proc. of ASP'07, Int.l Worksh. on Answer Set Programming Oporto, Portugal, Sept. 2007.*

[7] S. Costantini and A. Formisano. Modeling preferences on resource consumption and production in answer set programming. *Proc. of CILC'08, Ital. Conf. on Computational Logic, Perugia, Italy, July 2008.*

[8] S. Costantini, L. Mostarda, A. Tocchio and P. Tsintza. DALICA: Agent-Based Ambient Intelligence for Cultural-Heritage Scenarios. *IEEE Intelligent Systems*, vol.23, n.2, pp. 34-41, 2008.

[9] S. Costantini and A. Paolucci. Semantically Augmented DCG Analysis for Next-generation Search Engines. *Proc. of CILC'08, Ital. Conf. on Computational Logic, Perugia, Italy, July 2008.*

[10] S. Costantini and A. Tocchio. A logic programming language for multi-agent systems. *Logics in Artificial Intelligence, Proc. of the 8th Europ. Conf.,JELIA 2002*, LNAI 2424, Springer-Verlag, Berlin,2002.

[11] S. Costantini and A. Tocchio. The DALI logic programming agent-oriented language. *Logics in Artificial Intelligence, Proc. of the 9th European Conference, Jelia 2004*, LNAI 3229,Springer-Verlag, Berlin, 2004.

[12] S. Costantini and A. Tocchio. About declarative semantics of logic-based agent languages. *Declarative Agent Languages and Technologies, Post-Proc. of DALT 2005*, in M. Baldoni and P. Torroni Eds., LNAI 3229, Springer-Verlag, Berlin, 2006.

[13] S. Costantini, A. Tocchio, and P. Tsintza. A Heuristic Approach for P2P Negotiation. *Proc. of the Eighth-Workshop on Computational Logic in Multi-Agent Systems (CLIMA-VIII), September 2007, Porto, Portugal*, http://research.nii.ac.jp/climaVIII.

[14] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. *Proc. of the 5th Intl. Conference and Symposium on Logic Programming*, pages 1070–1080. The MIT Press, 1988.

[15] A. Kowalski. How to be Artificially Intelligent - the Logical Way *Draft, revised February 2004, Available on line, http://www-lp.doc.ic.ac.uk/UserPages/staff/rak/rak.html*, 2006.

[16] R. Mitkov. The Oxford Handbook of Computational Linguistics. *Natural Language, Center for the Study of Language and Information, StanfordUniversity*, Oxford University Press, 2005.

[17] A. Tocchio. Multi-Agent Systems in Computational Logic. *Ph.D. Thesis, Dipartimento di Informatica, Universitá degli Studi di L'Aquila*, 2005.

7 Contacts

Stefania Costantini (Corresponding Author)
Affiliation: Dept. of Computer Science, Univ. of L'Aquila
Address:Via Vetoio, Loc. Coppito, I-67010 L'Aquila, Italy
Telephone number:+390862433135 E-mail: stefcost@di.univaq.it.

Arianna Tocchio
Affiliation: Dept. of Computer Science, Univ. of L'Aquila
Address:Via Vetoio, Loc. Coppito, I-67010 L'Aquila, Italy
Telephone number:+390862433252
E-mail: tocchio@di.univaq.it.

Panagiota Tsintza
Affiliation: Dept. of Computer Science, Univ. of L'Aquila
Address:Via Vetoio, Loc. Coppito, I-67010 L'Aquila, Italy
Telephone number:+390862433252
E-mail: panagiota.tsintza@di.univaq.it.

8 Biography

Stefania Costantini is a full professor in the University of L'Aquila's Dept. of Computer Science. Her research interests include computational logic and AI, agents and multiagent systems, and answer set programming. She received her master's degree (the highest possible degree in Italy at the time) in computer science from the University of Pisa. She has been Co-hair for the Answer Set Programming 2007 conference.

Alessio Paolucci is a postgraduate student in computer science at the University of L'Aquila. His research interests mainly include Natural Language Processing. He received his Master's degree in Computer Science from the University of L'Aquila.

Arianna Tocchio is a postdoctoral student in computer science at the University of L'Aquila and a research assistant at the university. Her research interests include logical agents and multiagent systems and learning. She received her Master's degree in Computer Science from the University of L'Aquila.

Panagiota Tsintza is a PhD student at the University of L'Aquila. Her research interests are multiagent systems, computational logic agents and automated negotiation. She received her Master's degree in Computer Science from the University of L'Aquila.