Rule-Based Supervisor and Checker of Deep Learning Perception Modules in Cognitive Robotics

Abeer Dyoub and Giovanni De Gasperis

DISIM, Università degli Studi dell'Aquila, Italy

Abstract. This paper proposes the adoption of rule-based supervisor with logical agents as implementation of a reasoning layer embedded in cognitive robotic controllers where the world model representation is built from the output of a layer of deep learning perception modules. In this way the logical augmented cognitive robot could self-check in situations were safety and security issues have an impact in applications like user monitoring and robot navigation in shared spaces with humans. In previous related work we have suitably augmented the implementation of the DALI language, and we are presently developing promising sample applications in real robotic settings on which we shortly report here.

1 Introduction

Off-the-shelf deep learning technologies allow nowadays to readily implement perception modules for cognitive robots: image/scene description, object recognition, non conventional computer vision, dead reckoning positioning, sensor fusion, trajectory planning. This advanced input layer, made of sub-modules specialized to each perception task, is capable as a whole to build an internal digital and complex model of the world made of point clouds, 3d paths, moving labeled objects and signal time series that, considered together, provide world perception to the robot. This architecture has already demonstrated its potential in the autonomous car industry, where the driving decision about how to stay on the road is mostly determined by probabilistic inference systems in a timely manner, concerning several different aspects such as, e.g., obeying signals, avoiding obstacles and car accidents prevention. However, a limit of the approach is that the results of deep learning modules are applied unconditionally, and the decision of which particular course of action to take in case of conflicting indications does not exploit all rational elements that might be considered. Also, actual usefulness and adequacy of such results in the present setting is not evaluated in real-time, rather deferring the evaluation to future learning phases; instead, immediate timely adaptation would often be in order. Given a robotic system with a perception layer mostly based on deep learning modules, this work proposes the introduction of rule-based supervisor agents that can improve consistency of the perception layer; the rule based checker feeds a cascaded decision-maker, which could be either probabilistic, heuristic or deterministic in nature. The rule based perception checker layer is designed to increase the overall system trustworthiness towards one of possible states of the surroundings in order to anticipate the outcome of the robot actions and their impact on the environment in the immediate future time-frame. To implement such supervisors, that constitute a robot's "cognitive part", we envisage the adoption of Intelligent Agents, defined via rule-based declarative agent-oriented languages. There are many rule-based agent-oriented languages and architectures based on computational logic that could be apt to these aims, among which MetateM, 3APL, GOAL, AgentSpeak, Impact, KGP and DALI (the reader may refer to the surveys[1–3] and to the references therein), that might be exploited in robotics; in fact, several among the examples provided in the related literature concern potential robotic applications. We choose to experiment the use of the DALI language, as it has been developed by our research group and has been empowered over time with many features that can be useful to our aim. Specifically, DALI is equipped with: capabilities for the definition and management of an agent's memory and experience [4, 5], which can be exploited to compare the situation at hand with previous analogous ones; capabilities for user monitoring and training [6, 8]; DALI agents are also able to perform complex event processing [9, 10], and to dynamically modify their own behavior [11, 12]. These capabilities, though experimented in software agents, had never before been applied to robotics because the DALI implementation lacked the necessary features; we have recently developed such features, which are presented in [13]. Thus, via the extended implementation a DALI MAS (Multi-Agent System) can control a deep learning robot [14] platform by exchanging asynchronous JSON events over the new multi-standard DALI network bus. In addition, the ServerDALI module allows DALI agents and MASs to be located on a server, so as to be accessible from an external environment, or also via web or mobile applications. A cloud solution eliminates the need of equipping the (possibly diverse) robot hardware with sophisticated software; in fact, computationally heavy automated reasoning tasks can be more efficiently executed on the cloud, sharing an "event bus" with all involved robots. The novel contribution of the present paper is twofold: on the one hand, the re-thinking and extension of past work on DALI in the perspective of mobile robotic applications in unconstrained environments; on the other hand, the experimentation of the proposed approach on the extended DALI framework illustrated in [13]. In Section 2 we recall the basic DALI language, while in Section 3 we propose a discussion by means of small though significant examples about the potential applicability of DALI in robotics. Finally, in Section 4 we conclude.

2 The basic DALI Language and Architecture

DALI [15, 16] is an agent-oriented rule-based logic programming language which extends prolog. The DALI Multi Agent System Language and Framework has been developed at University of L'Aquila since 1999 and is publicly available at GitHub [17] . DALI agents are able to deal with several kinds of events: external events, internal, present and past events. **External events** are syntactically indicated by the postfix *E*. Reaction to each such events is defined by *reactive rules*, characterized by the special token :>. **Actions** (indicated with postfix *A*) may have or not preconditions, specified by special *action rules*. **Internal events** makes DALI agents proactive. An internal event is syntactically indicated by the postfix *I*, and its description is composed of two rules. The first one contains the conditions (knowledge, past events, procedures, etc.) that must be true so that the reaction (in the second rule) may happen. Thus, a DALI agent is able to react to its own conclusions. Internal events are automatically attempted with a default frequency customizable by means of directives in the initialization file. An agent remembers events and actions by recording **past event** (postfix *P*). The DALI communication architecture implements the DALI/FIPA protocol, which consists of FIPA primitives, plus few primitives which are particular to DALI; it is possible to define meta-rules for filtering incoming and outgoing messages, and for using external ontologies. A DALI MAS consists in a community of distributed agents' **instances**, derived from behavioral **types**; as a MAS starts, agent instances are generated from their respective type reference so to have sets of agents that share analogous behavior, even if instantiated with different initialization parameters, similarly to objects in the object-oriented programming paradigm. Each agent can have its own communication rules so as to encapsulate several communication models and design patterns.

3 Rule-Based Supervisors for Cognitive Robotics: basic ideas

The robotic applications that we envisage and that we are developing and experimenting concern user monitoring in diverse contexts. Our effort is especially intended for mobile robotics applications in unconstrained environments shared with humans. Below we show sample code excerpts concerning experiments that we are presently performing about the adoption of DALI for the implementation of robots' reasoning layer. Agents, as the robots' "brain", are generally aware by prior knowledge or via some form of learning of the behavioral patterns that other agents in the scene, such as humans and other robots, are adopting; they can also be able to learn rules and plans from other agents (by imitation or being told [8]). A mobile robot can use off-the shelf deep learning modules to achieve important perception tasks via its sensors, such as: scene description, people recognition and counting, path planning avoiding fixed obstacles and walking people, target recognition, speech recognition. All inputs coming from the perception layer can be encoded as asynchronous events with a data payload. However, such perceptions must be interpreted and can be reasoned about, so as to determine suitable robot's action. In the following simple example, when a certain door gets closed (it can be the door of a bus, of a train carriage, or the door of a room where a meeting will be held) then the robot can conclude that the number of people which is present will not change. To this conclusion, the robotic agent may react (special token "$:>$") by preparing and offering (the correct number of) drinks.

$people\_count\_can\_change$:- $door\_open$
$people\_count\_should\_be\_constant$:- $door\_closed$
$people\_count\_constantE$ $:> prepare\_and\_offer\_drinksA$
$people\_count\_should\_be\_constant \wedge people\_count\_changedE :> checkA$

Instead, whenever the deep learning computer vision module that is performing the people counting task would record an increasing number of persons while the door is still closed, it should conclude that either there is an error derived from sensor noise (such as a sudden different illumination of the room, e.g., from the sun appearing at the window after a rain event) or an improper behavior (intrusion) that may be occurring.

About learning from user's behavior, assume as a simple example that our agent has been somehow able to observe that the user normally (in a certain percentage of times) takes a drink when coming back home. This can be represented by a rule such as: $drink$:- $arrive\_home$ This learned rule can possibly be associated with a certainty factor. When the rule is later confronted with subsequent experience, its certainty factor will be updated, accordingly. Whenever this factor exceeds a threshold, this may lead to assert new meta-knowledge, such as: $USUALLY\ drink\ WHEN arrive\_home$ User monitoring can be performed via temporal-logic-like rules like the following sample one: $NEVER\ drink\_alchool\ AND\ take\_medicine$ where the semantics of such expressions is defined in [12]: the agent memory can be managed according to context-sensitive constraints valid in a time interval, where some events/actions must or must not occur. The care robot case is discussed in [13]. Below is a sample reactive pattern similar to those discussed in [11]. In particular, given background module M (that in our experiments is an Answer Set Programming module), reaction to event evE can be either any action which can be inferred (from M) as a possible reaction, or a *necessary* action, again according to M. The latter is preferred in a critical situation, with the connective $>$ expressing simple conditional preference order: the former option is preferred over the latter if the condition after the :- holds. Otherwise, any of the two options can be indifferently taken. $evE :> necessary(M) | action(M).$ $necessary(M)\ >\ action(M)\text{:-}critical\_situation.$ The monitoring component can however also include meta-axioms such as for instance the following, which states that a user action which is necessary to reach a basic objective should be undertaken: $ALWAYS\ do(user, A)$ $WHEN\ goal(G), necessary(G, A)$ Such a meta-rule could be applied to practical cases such as the following: $goal(healthy).$ $necessary(healthy, take\_medicine).$ The above small examples are instances of DALI constructs in the perspective of cognitive robotics applications; the aim is to show that DALI has indeed the potential for acting as an agent language for designing logic supervisors in the cognitive robotics realm. We have been performing several real examples, also by means of projects assigned to students of the "Intelligent Systems and Robotics Laboratory" course held at the University of L'Aquila (Master track in Computer Science).

4 Conclusions

In this paper we have showed the applicability of the DALI logical agent-oriented programming language in the cognitive robotic domain; we particularly envisage applications for user monitoring and robot navigation in shared space with humans. In our view, DALI agents can implement effective logic supervisors of deep learning robotic modules so as to provide consistent interpretations of perception layer's results. So far, we have performed a number of experiments in order to test the new implementation and to check its usability. At present, work is under way about the use of DALI for defining the cognitive self-checking sub-system of the deep learning robot architecture in an e-Health application (outlined in [18]). We do not consider either physical aspects concerning sensors, actuators, vision, etc., or deep learning input/output setup, that are however widely studied by specialists.

References

1. Fisher, M., Bordini, R.H., Hirsch, B., Torroni, P.: Computational logics and agents: a road map of current technologies and future trends. Comput. Int. J. **23**(1) (2007) 61–91
2. Bordini, R.H., Braubach, L., Dastani, M., ElSeghrouchni, A.F., Gomez-Sanz, J., Leite, J., O'Hare, G., Pokahr, A., Ricci, A.: A survey of programming languages and platforms for multi-agent systems. Informatica (Slovenia) **30**(1) (2006)
3. d'Inverno, M., Fisher, M., Lomuscio, A., Luck, M., de Rijke, M., Ryan, M., Wooldridge, M.: Formalisms for multi-agent systems. Knowledge Eng. Review **12**(3) (1997) 315–321
4. Costantini, S.: Defining and maintaining agent's experience in logical agents. In: Proc. of 7th Latin American Worksh. on Non-Monotonic Reasoning, LANMR 2011. Volume 804 of CEUR Workshop Proceedings., CEUR-WS.org (2011)
5. Costantini, S., De Gasperis, G.: Memory, experience and adaptation in logical agents. In: Management Intelligent Systems: Second Intl. Symposium, Proc. Advances in Intelligent and Soft Computing, Springer (2013)
6. Costantini, S., Dell'Acqua, P., Pereira, L.M., Toni, F.: Towards a model of evolving agents for ambient intelligence. In: Proc. of the Symposium on "Artificial Societies for Ambient Intelligence (ASAmI'07). (2007)
7. Costantini, S., Dell'Acqua, P., Moniz Pereira, L.: Conditional learning of rules and plans by knowledge exchange in log. agents. In: Proc. of RuleML 2011. LNCS 6826, Springer (2011)
8. Costantini, S., Riveret, R.: Event-action modules for complex reactivity in logical agents. In: Proceedings of AAMAS 2013, 13th Intl. Conf. on Autonomous Agents and Multi-Agent Systems, IFAAMAS/ACM (2014) 1503–1504 Extended Abstract.
9. Costantini, S.: ACE: a flexible environment for complex event processing in logical agents. In: Engineering Multi-Agent Syst., Third International Works., EMAS 2015, Revised Selected Papers. Volume 9318 of LNCS., Springer (2015)
10. Costantini, S., De Gasperis, G.: Complex reactivity with preferences in rule-based agents. In: Rules on the Web: Research and Applications, RuleML 2012, Proceedings. Volume 6826 of Lecture Notes in Computer Science., Springer (2012) 167–181
11. Costantini, S., De Gasperis, G.: Runtime self-checking via temporal (meta-)axioms for assurance of logical agent systems. In: Proc. of LAMAS 2014, 7th Works. on Logical Aspects of Multi-Agent Systems, held at AAMAS 2014. (2014) 241–255
12. Costantini, S., De Gasperis, G., Nazzicone, G.: DALI for cognitive robotics: Principles and prototype implementation. In: Practical Aspects of Declarative Languages - 19th International Symposium, PADL 2017, Proceedings. Volume 10137 of Lecture Notes in Computer Science., Springer (2017) 152–162
13. : Automous deep learning robot product ADLR: `http://www.autonomous.ai/deep-learning-robot`.
14. Costantini, S., Tocchio, A.: The DALI logic programming agent-oriented language. In: Logics in Artif. Int., Proc. of the 9th Europ. Conf. JELIA 2004. LNAI 3229, Springer (2004)
15. Costantini, S., Tocchio, A.: DALI: An architecture for intelligent logical agents. In: Proceedings of the Int. Workshop on Architectures for Intelligent Theory-Based Agents (AITA08). AAAI Spring Symposium Series (2008)
16. De Gasperis, G., Costantini, S., Nazzicone, G.: DALI multi agent systems framework (July 2016) DALI: `http://github.com/AAAI-DISIM-UnivAQ/DALI`.
17. Aielli, F., Ancona, D., Caianiello, P., Costantini, S., De Gasperis, G., Di Marco, A., Ferrando, A., Mascardi, V.: Friendly & kind with your health: Human-friendly knowledge-intensive dynamic systems for the e-health domain. In: Int. Conf. on Practical Appl. of Agents and Multi-Agent Systems. Comm. in Computer and Inf. Sc., Springer (2016) 15–26