An architecture for rational agents interacting with complex environments

A. Stankevicius   M. Capobianco   C. I. Chesñevar

Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Av. Alem 1253 - B8000CPB Bahía Blanca - ARGENTINA
E-mail: {ags,mc,cic}@cs.uns.edu.ar

Keywords: rational agency, argumentation, logic programming

1 Introduction

Constructing an *intelligent agent* is the main goal of artificial intelligence. To this purpose, the notions of *agency* and *intelligence* need to be precisely defined. It is well known that intelligence is hard to formalize as an abstract concept. To overcome this, Stuart Russell —among others— suggested the term *rational agency* as an alternative characterization for intelligent agency. In his own words [10], *"rational agents [...] are agents whose actions make sense from the point of view of the information possessed by the agents and its goals"*. Simply put, a rational agent is an agent that "does the right thing based on its beliefs" [6]. Naturally, the construction of rational agents is by no means an easy task.

Software Engineering has developed the notion of *architecture* in order to provide proper modularization for coping with large projects in a successful way. As formalizing rational agency is a complex task, different *agent architectures* have been proposed. The agent architecture supplies a separation of concerns, inducing a tentative modularization. However, at the present state of affairs, to build a fully rational agent is rather unattainable; several important issues remain overlooked, or are just partially addressed (*i.e.*, addressed under too unrealistic assumptions). Rao and Wooldridge [8] underscore the following aspects:

> *Rational agents are not solitary entities living in an uninhabited static environment; they are embedded in a continuously changing environment and have to constantly interact with other agents. Hence, communication with other agents and interaction with the environment are key concerns within this field....*

In this paper we sketch an agent architecture suitable to be used as a tool for exploring agent perception and multiagent interaction. Nowadays, there is no strict correspondence between the theoretical work in rational agents and their implementation. In this respect, it is our intention to reach a good trade-off between expressiveness and implementability.

Argumentative frameworks [3] constitute an appropriate formalism for knowledge representation and reasoning (mainly) as a consequence of its ability to deal with incomplete and/or contradictory information. An argumentative system underpins our architecture in order to represent the agent's knowledge, and perform inferences from it. The next section delineates the proposed architecture, and discusses its main components. Finally section 3 resumes the related work.

2 Agent architecture

As stated in the introduction, there is no general consensus on what rational agency is. However, most approaches share the following features:

- **Reactivity**: the ability to sense and act selectively.
- **Autonomy**: goal-directed, proactive, and self-starting behaviour.
- **Inferential capabilities**: the agent reasons from its knowledge and goes beyond the information given; it may have explicit models of itself, the environment, and/or other agents.
- **Social capabilities**: the ability to cooperate and/or coordinate with other agents to achieve common goals.

Any architecture for rational agents is likely to accommodate many of these features. Our proposed architecture (see Figure 1) is grounded on these four items. In the sequel, we detail the role of each component.

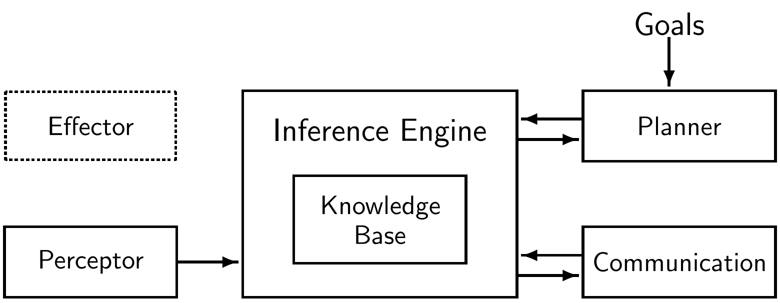

Figure 1: An sketch of the agent architecture.

2.1 Knowledge base

The knowledge base is the information repository of the agent. Even though multi-modal logics are usually the first choice for modelling an agent's mental state, the expressiveness of these approaches tends to be lost in the transition towards practical systems [8]. Logic programming provides a useful model for the epistemic state of an agent, as its formal specification and the corresponding implementation are closely related. Unfortunately, conventional logic programming cannot deal with partial and potentially contradictory information [4], a recurring situation when modelling practical agents. This restriction has motivated the development of extensions to logic programming that deal with these inconveniences. Observation-based defeasible logic programming (ODeLP) [2] is a formalism recently developed to model the epistemic state of agents in dynamic domains. ODeLP combines the advantages of logic programming with defeasible argumentation and perception mechanisms. Accordingly, we have chosen the language of ODeLP to conform the knowledge base of our agent.

2.2 Inference engine

To draw a conclusion from an ODeLP program arguments for and against a certain literal are obtained. These arguments undergo a dialectical analysis that establishes their current statuses. Hence, a query q can obtain as answer YES if there exists a justified argument for q, NO if there exists a justified argument for the complement of q, or UNDECIDED if neither q nor its complement are supported by a justified argument. In our architecture, the inference engine uses an ODeLP program as a knowledge base to solve queries posed by the other units.

2.3 Planning unit

The planning unit exercises the control of the agent. It constructs plans to achieve agent's goals based upon the current epistemic state. In this setting, we understand a plan to mean any sequence of actions. The basic actions available to the agent involve consulting other agents about their epistemic state, engaging them in negotiations, and any other elementary operation performed by its effectors.

2.4 Communication unit

The communication unit embodies the *high-level* perceptors and effectors that manage the interaction among agents. Two levels can be identified. The first one, *atomic interaction*, includes both querying agents to gain insights about their epistemic states, and accomplishing queries from other agents. In contrast *composite interaction* comprehends any non-atomic interaction among agents, such as coordination, cooperation, and negotiation.

Atomic interaction is easily achieved: queries can be posed by means of the communication unit, and the inquiries received can be easily answered through the inference engine. We assume that every agent is aware of the existence of its counterparts: the communication unit maintains the information required to access any of them. Note that atomic interaction among agents exchanges unstructured information only. Thus, no specific protocol is required.

For composite interaction, we focus our attention on deliberation. In our approach most of the other high level interactions (such as cooperation, persuasion, negotiation, etc.) as by-products of deliberation. As deliberation we understand the process through which a set of entities come to mutually accepted position regarding a given issue. In contrast to atomic interaction, the deliberation process involves the exchange of elaborated evidence about the topic being considered. For instance, in this context the parties exchange arguments either supporting or rebutting a given claim. Clearly, a protocol governing these transactions needs to be defined.

2.5 Perceptors and effectors

A rational agent residing in a complex dynamic environment must be able to gather information by sensing its surroundings [6]. The perceptor unit is responsible for handling this interaction, reporting environment changes so they can be incorporated into the knowledge base. The effector unit executes a set of actions over the world, as to follow the requirements of the planner. This unit often owns a variety of actions. As we concentrate on agent perception and multi-agent negotiation, the architecture does not require any particular action besides those provided by the communication unit. Consequently, Figure 1 depicts the effector unit in dashed lines.

3 Current and related work

In this section we survey the main ideas on agent perception and multi-agent negotiation being developed using the proposed architecture. These results comprise the research being conducted in two different, but interconnected, research lines.

3.1 Agent perception

We already stressed the importance of considering a rational agent as an entity interacting in a changing environment. Clearly, it is impossible to provide the agent with all the required information from its conception; this prompts for the definition of mechanisms suitable for agent perception. Our perceptor unit senses the environment, providing concrete facts about the world. However, the agent cannot monitor the entire state of the world at all time; the best perception can do is to provide the agent with a sort of sampling at discrete times, or over shot time intervals [6].

Consistency is an important issue when formalizing perception. The world, as it evolves, need not be consistent with its previous states (*i.e.*, a new perception may contradict a previous one). An agent sensing its environment must be able to deal with the inconsistencies arising from its perceptions. Consequently, the argumentative framework encoding the agent's knowledge must be able to distinguish its previous knowledge from the new information acquired through perception. In order to cope with this problem, the ODeLP framework is provided with the ability to update the knowledge base as it receives new perceptions. An in-depth treatment of these issues is outside the scope of this paper. More details concerning this approach can be found in [2].

3.2 Interaction among agents

The traditional approach for modelling multiagent interaction resorts to game theory [9]. Although several issues have been explored under this conception, most of them rely on a strong assumption: each agent should be aware of the complete pay-off matrix *before* the interaction even starts. The implications of this strong assumption undermines the applicability of the whole approach.[1]

In a prominent paper, Parsons *et al.* [5] develop the idea that multiagent interaction can be seen as an argumentative process. In an attempt to follow their intuitions, we also adopt an argumentative framework to model multi-agent negotiation (note that an argumentative framework is already present in our architecture).

Recent progress in defeasible argumentation favours dialectical characterisation, such as Prakken's dialogue games [7], Simari's dialectical trees [11], etc. Under this particular view,

[1] It should be noted that there also exist variants where such a matrix can be computed dynamically.

two parties take opposing positions regarding a certain matter, and alternatively pose reasons backing their stances. This view extends quite naturally to a *set* of three or more agents interacting with each other.

We content that, instead of circumscribing only to the actual interaction, the entire process should be taken into account. At the current state of the research, the following steps have been identified:

1. An agent A decides that it needs to interact with another agent B. A's planning unit is responsible for making this decision.

2. Agent A engages the chosen counterpart B. Interaction between them can start once A's communication unit contacts B's.

3. The actual interaction takes place. In this step, the interaction is conducted according to an argumentative protocol.

4. The outcome of the interaction is accounted. Note that when a given agent does not succeed in a negotiation process, it accepts the issue under debate. Its knowledge base becomes contradictory. In this step, the ability to deal with contradictory information becomes indispensable.

Part of our current research work involves further refinement of the above steps. In particular, the interested reader is referred to [1] for a detailed discussion concerning steps 3 and 4.

References

[1] A.G.Stankevicius, and G.R.Simari. An abstract model for the process of deliberation within multiagent systems. In *Proceedings del 7mo Congreso Argentino de Ciencias de la Computación (CACIC)* (2001), Universidad Nacional de la Patagonia Austral, pp. 1017–1026.

[2] Capobianco, M., Chesñevar, C. I., and Simari, G. R. An argumentative formalism for implementing rational agents. In *Proceedings del 7mo Congreso Argentino de Ciencias de la Computación (CACIC)* (2001), Universidad Nacional de la Patagonia Austral, pp. 1051–1062.

[3] Chesñevar, C. I., Maguitman, A., and Loui, R. Logical Models of Argument. *ACM Computing Surveys 32*, 4 (2000), 337–383.

[4] Gelfond, M., and Lifschitz, V. Logic Programs with Classical Negation. In *Proceedings of the 7th International Conference on Logic Programming* (June 1990), D. H. D. Warren and P. Szeredi, Eds., pp. 579–597.

[5] Parsons, S., Sierra, C., and Jennings, N. Agents that reason and negotiate by arguing. *Journal of Logic and Computation 8*, 3 (1998), 261–292.

[6] Pollock, J. L. Rational Cognition in OSCAR. In *Proceedings of the 6th International Workshop on Agent Theories, Architectures, and Languages* (1999). To appear.

[7] Prakken, H. *Logical Tools for Modelling Legal Argument.* Kluwer Academic Publishers, 1997.

[8] Rao, A., and Wooldridge, M. Foundations of Rational Agency. In *Foundations of Rational Agency*, no. 14 in Applied Logic Series. Kluwer Academic Publishers, 1999, pp. 1–10.

[9] Rosenschein, J. R., and Zlotkin, G. *Rules of Encounter.* MIT Press, 1994.

[10] Russell, S. J. Rationality and Intelligence. *Artificial Intelligence 94*, 1–2 (1997), 57–77.

[11] Simari, G. R., Chesñevar, C. I., and García, A. J. The Role of Dialectics in Defeasible Argumentation. In *Proceedings of the XIV Conferencia Internacional de la Sociedad Chilena para Ciencias de la Computación* (Nov. 1994), pp. 111–121. `http://cs.uns.edu.ar/giia.html`.