

# Extending agent languages for autonomy

Felipe Rech Meneguzzi  
King's College London  
Department of Computer Science  
London, United Kingdom  
felipe.meneguzzi@kcl.ac.uk

## ABSTRACT

BDI agent languages provide a useful abstraction for complex systems comprised of interactive autonomous entities, but they have been used mostly in the context of single agents with a static plan library of behaviours invoked reactively. These languages provide a theoretically sound basis for agent design but are very limited in providing direct support for autonomy and societal cooperation needed for large scale systems. Some techniques for autonomy and cooperation have been explored in the past in *ad hoc* implementations, but not incorporated in any agent language. In order to address these shortcomings we extend the well known AgentSpeak(L) BDI agent language to include behaviour generation through planning, declarative goals and motivated goal adoption. We also develop a language-specific multiagent cooperation scheme and, to address potential problems arising from autonomy in a multiagent system, we extend our agents with a mechanism for norm processing leveraging existing theoretical work. These extensions allow for greater autonomy in the resulting systems, enabling them to synthesise new behaviours at runtime and to cooperate in non-scripted patterns.

## 1. INTRODUCTION

Agents as a metaphor for complex systems has been gaining popularity for years, and research on agent programming languages has followed in two main veins: one focusing on adapting traditional object-oriented languages to the task of creating multiple agents, and one focusing on creating simple, yet theoretically sound agent languages with a view towards proving certain properties of systems created using them. Examples of the first type of agent language include JADE [9] and JACK [1], whereas AGENT0 [11] and AgentSpeak(L) [10] represent examples of the second type. The strong connection to a lower-level object-oriented language makes the first type of language more flexible in creating real systems, but loses some of the advantages that the agent abstraction could provide, since a designer needs to recreate certain agent techniques in the lower level language. Alternatively, the simple languages of the second type were designed for single simple agents and lack many features for developing complete software solutions.

Padgham, Parkes, Mueller and Parsons (eds.): Proc. of AAMAS 2008, Estoril, May, 12-16., 2008, Portugal, pp. XXX-XXX. **Cite as:** Title, Author(s), Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems, pp. XXX-XXX, 2008.  
Copyright © 2008, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved..

Progress on the development of practical agent languages has not occurred at the same pace as efforts in developing new architectures and *ad hoc* techniques. While concepts such as autonomous behaviour and techniques for cooperation have been recognised as paramount to practical autonomous agents, few of them have been integrated into existing agent languages. For example, declarative goals have been advocated by Winikoff *et al.*[12] as paramount for autonomous agent behaviour, their inclusion in a practical agent language came only in Hübner *et al.*[2]. Similarly, multiagent planning techniques have been developed [3], but none have been incorporated in an agent language so that a designer needs not to re-implement it in a lower-level programming language.

In our work, we consider the issues of autonomy and cooperation in agent architectures and incorporate them into the popular AgentSpeak(L) language, tooling it for the development of complete multiagent systems. In order to facilitate the creation of autonomous agents, we take the concept of declarative goals supported by planning and extend the language to reason about desired world states rather than simply adopting procedural plans with no knowledge of the desired result, synthesising new plans to allow the agent to achieve these world states. We then proceed in allowing goals to be generated by an underlying model of motivations, which allows an agent to generate its own goals instead of strictly responding to events in the environment. Finally, in order to facilitate describing the *macro* level of an agent system we extend the language to allow non-scripted cooperative behaviour, as well as the processing social norms.

## 2. AGENTSPEAK(PL)

Research on practical models of autonomous agents has largely focused on a procedural view of goal achievement. This allows for efficient implementations, but prevents an agent from reasoning about alternative courses of action for the achievement of its design objectives. In [4, 5] we show how a procedural agent model can be modified to allow an agent to compose existing plans into new ones at runtime to achieve desired world states. This new agent model can be used to implement a declarative goals interpreter, since it allows designers to specify only the desired world states in addition to an agent's basic capabilities, enhancing the agent's ability to deal with failures. Moreover our approach allows the new plans to be included in the plan library, effectively enabling the agent to improve its runtime performance over time.

### 3. MOTIVATIONS IN META-REASONING

In agent systems, meta-level reasoning is commonly used in enforcing rationality in the choice of goals and actions performed by an agent, ensuring that an agent behaves as effectively and efficiently as possible. Through meta-reasoning an agent is able to explicitly consider goals before committing to them, and consider courses of action before executing plans. We argue that although seldom considered, a flexible meta-level reasoning component is a valuable addition to any agent architecture. In [6], we describe such a component for use in BDI architectures, underpinned by a model of motivation and a motivation-based description language, and demonstrate its effectiveness empirically.

### 4. SOCIAL AGENTSPEAK(L)

The development of practical agent languages has progressed significantly over recent years, but this has largely been independent of distinct developments in aspects of multiagent cooperation and planning. In particular, multiagent cooperation in practical agent languages is seldom addressed. For example, while the popular AgentSpeak(L) has had various extensions and improvements proposed, it still is essentially a single-agent language. In response [7], we address the situation in which an agent can make use of other cooperative agents by constructing new plans that involve subplans adopted from, and performed by, these cooperative agents. We describe a simple, yet effective, technique for multiagent planning that enables an agent to take advantage of cooperating agents in a society. In particular, we build on a technique that enables new plans to be added to a plan library through the invocation of an external planning component, and extend it to include the construction of plans involving the chaining of subplans of others. Our mechanism makes use of *plan patterns* that insulate the planning process from the resulting distributed aspects of plan execution through local *proxy plans* that encode information about the preconditions and effects of the *external plans* provided by agents willing to cooperate. Proxy plans encapsulate the communication and cooperation necessary for the remote execution of external plans, so that from the planner's perspective, all capabilities seem to be local. In this way, we allow an agent to discover new ways of achieving its goals through local planning and the delegation of tasks for execution by other agents that possess abilities not available to the planning agent, allowing it to overcome individual limitations.

### 5. NORMATIVE PROCESSING IN AGENTSPEAK(L)

Though autonomy is touted as one of the main advantages of agent-based systems, completely autonomous agents may create problems when they engage in unexpected behaviour. In order to design a system based on autonomous agents, a designer must be able to define rules to prevent undesirable behaviours from taking place [8]. One method upon which recent efforts have focused is based on the definition of social norms that agents must obey or else face sanctions. We provide a practical method of incorporating norm-based controls in AgentSpeak agents through the use of a meta-level toolkit that modifies an agent's plan library to make it comply with a set of norms that can be received while the agent runs. Modifications to the plan library involve sup-

pressing plans that violate prohibitions and creating new plans to comply with obligations.

### 6. REFERENCES

- [1] N. Howden, R. Rönquist, A. Hodgson, and A. Lucas. Jack: Summary of an agent infrastructure. In *Proceedings of the 5th International Conference on Autonomous Agents*, 2001.
- [2] J. F. Hübner, R. H. Bordini, and M. Wooldridge. Plan patterns for declarative goals in agentspeak. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1291–1293, 2006.
- [3] D. Kalofonos and T. J. Norman. An investigation into team-based planning. In *2004 IEEE International Conference on Systems, Man and Cybernetics*, pages 5590–5595, 2004.
- [4] F. Meneguzzi. Motivations and declarative goals as cornerstones of autonomy. In *Student Session of the Eight European Agent Systems Summer School*, 2006.
- [5] F. Meneguzzi and M. Luck. Composing high-level plans for declarative agent programming. In *Proceedings of the Fifth Workshop on Declarative Agent Languages*, pages 115–130, 2007.
- [6] F. Meneguzzi and M. Luck. Motivations as an abstraction of meta-level reasoning. In H.-D. Burkhard, G. Lindemann, R. Verbrugge, and L. Z. Varga, editors, *Proceedings of the 5th International Central and Eastern European Conference on Multi-Agent Systems*, volume 4696 of *LNAI*, pages 204–214. Springer, 2007.
- [7] F. Meneguzzi and M. Luck. Interaction among agents that plan. In *Sixth International Workshop "From Agent Theory to Agent Implementation"*, 2008. (submitted).
- [8] F. Meneguzzi, S. Miles, C. Holt, M. Luck, N. Oren, N. Faci, and M. Kollingbaum. Electronic contracting in aircraft aftercare: A case study. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, 2008. To Appear.
- [9] A. Pokahr, L. Braubach, and W. Lamersdorf. Jadex: A BDI reasoning engine. In R. H. Bordini, M. Dastani, J. Dix, and A. E. Fallah-Seghrouchni, editors, *Multi-Agent Programming: Languages, Platforms and Applications*, pages 149–174. Springer, 2005.
- [10] A. S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In W. V. de Velde and J. W. Perram, editors, *Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, volume 1038 of *LNCS*, pages 42–55. Springer, 1996.
- [11] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1):51–92, 1993.
- [12] M. Winikoff, L. Padgham, J. Harland, and J. Thangarajah. Declarative & Procedural Goals in Intelligent Agent Systems. In D. Fensel, F. Giunchiglia, D. L. McGuinness, and M.-A. Williams, editors, *Proceedings of the Eighth International Conference on Principles and Knowledge Representation and Reasoning*, pages 470–481. Morgan Kaufmann, 2002.