UNREAL GOAL Bots

Koen V. Hindriks [a] Birna van Riemsdijk [a] Tristan Behrens [b]
Rien Korstanje [a] Nick Kraayenbrink [a] Wouter Pasman [a]
Lennard de Rijk [a]

[a] *Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands*
[b] *Clausthal University of Technology, Julius-Albert-Straße 4, 38678 Clausthal, Germany*

Abstract

It remains a challenge with current state of the art technology to use BDI agents to control real-time, dynamic and complex environments. We report on our effort to connect the GOAL agent programming language to the real-time game UNREAL TOURNAMENT 2004. We focus in particular on the design of a suitable interface to manage agent-bot interaction and argue that the use of a recent toolkit for developing an agent-environment interface provides many advantages.

1 Introduction

Connecting cognitive or rational agents to an interactive, real-time computer game is a far from trivial exercise. This is especially true for agents that use logic to represent and reason about the environment they act in. There are several issues that need to be addressed ranging from technical to more conceptual issues. The focus of this paper is on the design of an interface that is suitable for connecting logic-based BDI agents to the real-time game UNREAL TOURNAMENT 2004 (UT2004, see [1]), but we also touch on some related, more technical issues and discuss some of the challenges and potential applications that motivated our effort.

The design of an interface for connecting logic-based BDI agents to a real-time game is complicated for at least two reasons. First, such an interface needs to be designed at the right *abstraction level*. The reasoning typically employed by logic-based BDI agents does not make them suitable for controlling low-level details of a bot. Conceptually, it does not make sense, for example, to require such agents to deliberate about the degrees of rotation a bot should make when it has to make a turn. This type of low-level control is better delegated to a behavioral control layer. At the same time, however, the BDI agent should be able to remain in control and the interface should support sufficiently fine grained control. Second, for reasons related to the required responsiveness in a real-time environment and efficiency of reasoning, the interface should not flood such an agent with percepts. Providing a logic-based BDI agent with huge amounts of percepts would overload the agents' processing capabilities. The *cognitive overload* thus produced would slow down the agent and reduce its responsiveness. At the same time, however, the agent needs to have sufficient information to make reasonable choices of action while taking into account that the information to start with is at best incomplete and possibly also uncertain.

We have used and applied a recently introduced toolkit called the Environment Interface Standard (EIS) [2] to implement an interface for connecting agents to a gaming environment, and we evaluate this interface for designing a high-level interface that supports relatively easy development of agent-controlled bots. We believe that making environments easily accessible will facilitate the evaluation and assessment of performance and the usefulness of features of agent platforms. Various projects have connected agents to UT2004 and in the full paper we discuss some of these projects and the differences with our approach. Most importantly, the design of the agent interface reported here has quite explicitly taken into account what would provide the right abstraction level for connecting logic-based BDI agents such as GOAL agents (see [4] for an introduction) to UT2004.

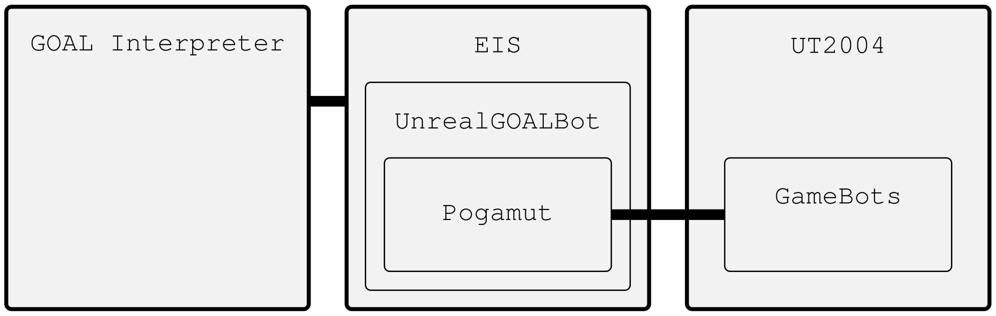

Figure 1: Schematic implementation overview. The GOAL-interpeter connects to the EIS, which wraps UnrealGOALBot. UnrealGOALBot wraps Pogamut, which connects to GameBots, an UNREAL-plugin.

2 Agent Interface for Controlling UNREAL Bots

The *Environment Interface Standard* (EIS) [2]is a proposed standard for interfaces between (agent-)platforms and environments. We have chosen to use EIS because it increases the reusability of environments and it provides ready support for event and notification handling and for launching agents and connecting to bots. EIS facilitates acting, active sensing (actions that yield percepts), passive sensing (retrieving all percepts), and percepts-as-notifications (percepts sent by the environment).

The connection established using EIS between GOAL-agents, which are executed by the GOAL-interpreter, and UT2004 bots in the environment consists of several distinct components (see Fig. 1). The first component is GOAL's support for EIS. Basically this boils down to a sophisticated MAS-loading-mechanism that instantiates agents and creates the connection between them and entities, together with a mapping between GOAL-percepts/actions and EIS ones. Entities, from the environment-interface-perspective, are instances of *UnrealGOALBot*, which is a heavy extension of the *LoqueBot* developed by Juraj Simlovic. LoqueBot on the other hand is built on top of Pogamut[3]. Pogamut itself is connected to *GameBots*, which is a plugin that opens UT2004 for connecting external controllers via TCP/IP.

3 Conclusion and Future Work

As is well-known, the UNREAL engine is used in many games and various well-known research platforms such as the USARSIM environment for crisis management. We believe that the high-level Environment Interface that we have made available to connect agent platforms with UT2004 will facilitate the connection to other environments such as USARSIM as well. The availability of this interface makes it possible to connect arbitrary agent platforms with relatively little effort to such environments which opens up many possibilities for agent-based simulated or gaming research. This is beneficial to put agent technology to the test. The framework, moreover, has been used for educational purposes.

The connection of an agent programming language for rational or BDI agents to UT2004 poses quite a few challenging research questions. A very interesting research question is whether we can develop agent-controlled bots that are able to compete with experienced human players using the same information the human players possess. The work reported here provides a starting point for this goal. Even more challenging is the question whether we can develop agent-controlled bots that cannot be distinguished by experienced human players from human game players.

References

[1] Unreal Tournament 2004. `http://www.unrealtournament2003.com/ut2004/index.html` (Accessed 30 Jan 2010).

[2] Tristan M. Behrens, Jürgen Dix, and Koen V. Hindriks. Towards an Environment Interface Standard for Agent-Oriented Programming. Technical report, Clausthal University of Technology, IfI-09-09, September, 2009.

[3] Ondrej Burkert, Rudolf Kadlec, Jakub Gemrot, Michal Bída, Jan Havlíèek, Martin Dörfler, and Cyril Brom. Towards fast prototyping of IVAs behavior: Pogamut 2. In *Proc. of the 7th Int. Conf. on Intelligent Virtual Humans*, 2007.

[4] Koen V. Hindriks. Programming Rational Agents in GOAL. In *Multi-Agent Programming: Languages, Tools and Applications*, chapter 4, pages 119–157. Springer, 2009.