

Chapter 4

A Multiagent Approach to Teaching Complex Systems Development

Simon C. Lynch
University of Teesside, UK

Keerthi Rajendran
University of Teesside, UK

ABSTRACT

Teaching practical development of multi-agent systems (MAS) presents various difficulties. This is also the case for application areas, like multimodal dialog systems (MMD), which rely on agent technologies to provide their systems infrastructure. These areas are challenging for novices and require significant pre-requisite knowledge from both learners and educators. The authors aim to reduce these difficulties, targeting teaching towards undergraduate and postgraduate computing students from increasingly diverse educational backgrounds. The authors specify an adaptable multi-agent architecture and provide a set of general purpose agents to handle essential multimodal tasks like speech input/output, fusion and semantic analysis. They outline the design of this system and describe how it provides a framework for students to assemble complex systems and experiment with agent-level design patterns. They evaluate the usability of the resulting software and tools using the Cognitive Dimensions Framework and by examining students' experience of using this approach in computing courses and project work.

INTRODUCTION

In addition to numerous other advantages, the multi-agent systems (MAS) concept provides a level of abstraction above that of object orientation and is a suitable base technology for simplifying the complexity of development in various application areas (Ferber, 1999; Jennings, 2001;

Sycara, 1998). The research community has produced a choice of agent languages, platforms and frameworks to support MAS (Bordini, R. H. et al., 2006) but these have been developed primarily to support particular notions of agency in research or to address the requirements of specific target applications rather than to meet the needs of learners and to provide generic support for applications development. Adopting an existing agent language for teaching has the benefit that

DOI: 10.4018/978-1-60960-080-8.ch004

students learn to use a system which has a wider user-base and acquire knowledge which may be useful to them in future but this must be offset against any limitations imposed by the system and/or issues of learnability. Many languages (e.g. Jack, Jason, 3APL) embody specific flavours of agency (BDI for example) which constrain the types of agents that they can specify and limit the possibilities for students to experiment with agent concepts in general. Our interest is in finding a framework which can provide support for building various distributed, multi-platform software architectures including those that integrate with other systems (databases, language processors, etc) and which are suitable for building multimodal dialog (MMD) systems. BDI agents have been used as a basis for some courses in agency (Bordini, 2005) but these agents have their activities described in terms of goals and planning solutions (typically using a rule-based approach). This can be difficult for learners who are not used to specifying the behaviour of software in these terms and BDI agents can present interoperability issues with other (non-BDI) languages/subsystems. Other types of agent system, Netlogo (Wilensky, 1999) for example, have been used in teaching (Beer, 2005; Centola et. al., 2000) but Netlogo does not easily interoperate with other languages or subsystems either and the language used to program Netlogo agents is too constrained for general programming support.

Even with more generic platforms (e.g. JADE), the focus is in providing the facilities needed by experienced developers rather than those preferred for learners and educators. For many students, whose primary experience of practical software development has been with languages like C++, Java or C# and who have largely adopted a single-thread, non-distributed model for their software, the move to agents presents a variety of challenges. Agents are typically concurrent, existing in their own process thread, they are distributed and agent systems are conceived in terms of behaviours and communication. Additionally, many existing agent

languages present difficulties for learners at both the syntactic and conceptual levels because their design is based on theories of agent behaviour rather than on pre-existing knowledge we would expect from students. In some cases this situation is made more difficult because of the lack of tools available to help develop and debug agent systems.

To avoid these problems we have developed an agent framework designed specifically for learnability. In addition to our aims of providing an agent language that novices can use to build their own agents, we are also interested in providing a framework suitable for students to assemble complex/challenging systems from pre-existing agents. Due to our other research and educational interests we select multimodal dialog systems as the target application area for this and provide a library of predefined agents for this area. MMD systems are those which provide the user with multiple modes of user interaction which may be used concurrently. A MMD system may, for example, provide users with speech input and gesture recognition allowing them to speak and point, issuing statements like “*put that there*” which can only be processed when all input modalities are considered. MMD systems typically also provide multiple output modalities (e.g. speech output and animated graphics) and may process user interaction as continuing strands of conversation rather than as a series of discreet statements.

Incorporating practical methods and technologies into MMD courses can be difficult for several reasons. Firstly there is a requirement for varied and specialist technologies, such as speech recognition and gaze tracking, and using these is difficult without specific expertise. Modern complex applications such as multimodal systems are typically developed jointly by experts in a variety of technologies. Even in research environments, the development of prototype systems presents a “demanding challenge” (Herzog et al., 2003).

Secondly the processes involved in using a combination of modes to capture inputs, e.g. speech and facial expression, *in parallel* can be

16 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/multiagent-approach-teaching-complex-systems/50395

Related Content

An Architecture for Developing Multiagent Educational Applications for the Web

Tasos Triantis and Panayiotis Pintelas (2008). *Agent-Based Tutoring Systems by Cognitive and Affective Modeling* (pp. 307-323).

www.irma-international.org/chapter/architecture-developing-multiagent-educational-applications/5052

Epistemological Aspects of Simulation Models for Decision Support

Anthony H. Dekker (2013). *International Journal of Agent Technologies and Systems* (pp. 55-77).

www.irma-international.org/article/epistemological-aspects-of-simulation-models-for-decision-support/87166

Artificial Minds with Consciousness and Common sense Aspects

K.R. Shylaja, M.V. Vijayakumar, E. Vani Prasad and Darryl N. Davis (2017). *International Journal of Agent Technologies and Systems* (pp. 20-42).

www.irma-international.org/article/artificial-minds-with-consciousness-and-common-sense-aspects/201443

Hierarchical Multi-Agent Plans Using Model-Based Petri Net

Said Brahimi, Ramdane Maamri and Zaidi Sahnoun (2013). *International Journal of Agent Technologies and Systems* (pp. 1-30).

www.irma-international.org/article/hierarchical-multi-agent-plans-using-model-based-petri-net/87147

Analyzing Transactions Costs in Transport Corridors Using Multi Agent-Based Simulation

Lawrence Henesey and Jan A. Persson (2009). *Multi-Agent Systems for Traffic and Transportation Engineering* (pp. 342-356).

www.irma-international.org/chapter/analyzing-transactions-costs-transport-corridors/26947