Demo paper: AGADE

Using communities of agents to provide realistic feedback in business simulations

Thomas Farrenkopf[1], Michael Guckert[1]
and Neil Urquhart[2]

[1] KITE - Kompetenzzentrum für Informationstechnologie,
Technische Hochschule Mittelhessen, Germany
{thomas.farrenkopf, michael.guckert}@mnd.thm.de
[2] School of Computing, Edinburgh Napier University, Scotland
n.urquhart@napier.ac.uk

Abstract. The need to provide realistic feedback against decisions made within business games is a requirement if business games are to continue to remain relevant in training towards increasingly complex business scenarios. We attempt to address this problem by using software agents to simulate individuals and to model their actions in response to business decisions. In our initial studies we use agent technologies to simulate consumers who will make buying decisions based on their own preferences and those within their social network. Other applications are search for structure in complex contexts, or verification of predicted values based on theoretical considerations. In this demo paper we present the tool set AGADE (Agile Agent Development Environment) which incorporates agent based and semantic technologies to address this. It is applied to simulate different market mechanisms in a mobile phone market.

Keywords: Multi-Agent System, BDI, OWL Ontology, Market Simulation, Human Behaviour

1 Introduction

Traditionally, business games attempt to model real world business scenarios. They are typically based on mathematical models where cause-and-effect relationships are basically represented in difference and differential equations. Agent-based systems are perfectly suited to describe individuals and their behaviour rather than effects through equations and thresholds. Behaviour can be modelled for each agent and its effects on the comprising environment can be determined by running simulations. The BDI software model (belief-desire-intention) provides a common well established approach for building multi-agent simulations. Moreover ontologies can be used to make world knowledge available to the agents which can then determine their actions in accordance with this knowledge. We demonstrate how ontologies can be integrated into the BDI concept namely into the popular agent framework Jadex [6]. Each agent will maintain its knowledge

in an individual ontology and then access this dynamically during runtime. The generic approach will allow the simulation of different scenarios which can easily be modelled by creating appropriate ontologies. The approach is implemented in an interactive round based multi-agent simulation tool (AGADE). It was applied to simulate different market mechanisms in a mobile phone market, which will be presented in this demonstration.

2 Main purpose

AGADE aims at providing an integrated environment in which round based multi-agent simulations can be modelled and executed. Its emphasis lies on modelling individuals that are aware of their environment, have knowledge and are able to learn. A means for that is integrating ontologies and multi-agent systems so that the modelling of knowledge is shifted to writing OWL ontologies. Powerful inference engines are then available and can be exploited from within the BDI infrastructure.

3 Demonstration

AGADE is a round based multi-agent tool set designed to support the development and calibration of dynamic business scenarios. It is based on the Jadex framework which allows the definition of BDI agents. Agents are active parts of a complex social structure, allowing them to not only communicate but also to permanently learn from each other.

In our case study of a mobile phone market we have to distinguish between customer and seller agents. Therefore we have to define two types of agents and the number of agents each type that are to take part in the simulation. Before starting a simulation we have to define the social structure comprised of the mutual relations of all agents. We provide an adjacency matrix (see Fig. 1) in which we can manually define the *who knows whom* relations or calculate them using an appropriate algorithm such as Barabasi's preferential attachment [1]. Influence matrices quantify the mutual influence agents may have. Additionally other relational aspects (each with its own adjacency and influence matrix respectively) can be built e.g. random graph like structures. These matrices can be used to define the degree of technical understanding an agent a attributes to another agent b or the quantified degree to which one agent is affected by another.

Each agent is equipped with its own inference engine (reasoner) and private ontology which is accessed using the OWL API [4]. Social aspects and information about the agent's current state are mapped to the ontology. The simulation itself can be controlled using the GUI displayed in Fig. 2. On the top of the GUI the control buttons are located: Because of AGADE's round-based approach, between any two rounds a simulation can be halted so that further inspections of the current state of affairs are possible. Simulation data is displayed continuously.

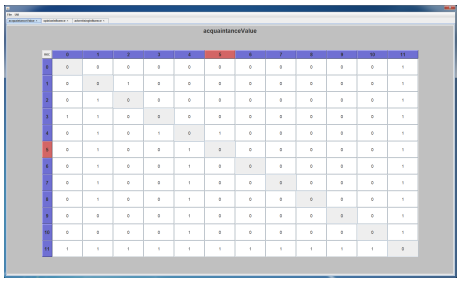

Fig. 1. Adjacency matrix setup.

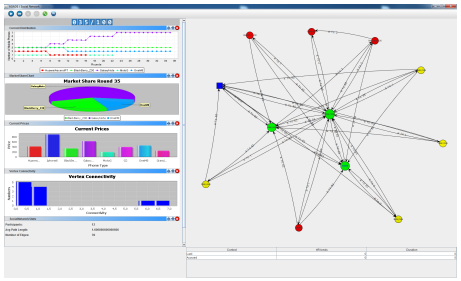

Fig. 2. AGADE simulation GUI.

On the right hand side of the screen there is a graphical display of the social structure formed by all participating agents. The vertices of the graph represent the agents using different shapes for different agent types and different sizes to indicate the popularity (the amount of edges connected to them) of an agent. The agent types are visualised by different geometrical forms: squares represent sellers and consumers are represented by circles. The edges between the vertices depict the relations of the agents giving a precise description of each relation as they are labelled with the respective relation indices. The colours of the consumer vertices indicate the state of the respective agents e.g. the happiness with their current product.

On a very abstract level AGADE (see Fig. 3) knows two different kinds of BDI agents: A director type agent that acts as some kind of conductor for the simulation which triggers the beginning of each new round and participant type that comprises any kind of agent participating in the simulation (i.e. consumer and seller).

Each round is composed of four phases:

Control phase The director processes commands issued by the user during execution of the last round. GUI components are updated.

Calculation phase The director tells every single participant to make its necessary calculations (e. g. updating its happiness value).

Socialisation phase Agents update their mutual relationships and possibly build new relationships.

Acting phase The next agent actions are triggered. This depends on the individual state and the rules expressed in the agent's ontology.

Note that the ontology based belief base leads to a very flexible architecture, because important aspects of the agent do not have to be coded statically any more but may be expressed in the rules of the ontology.

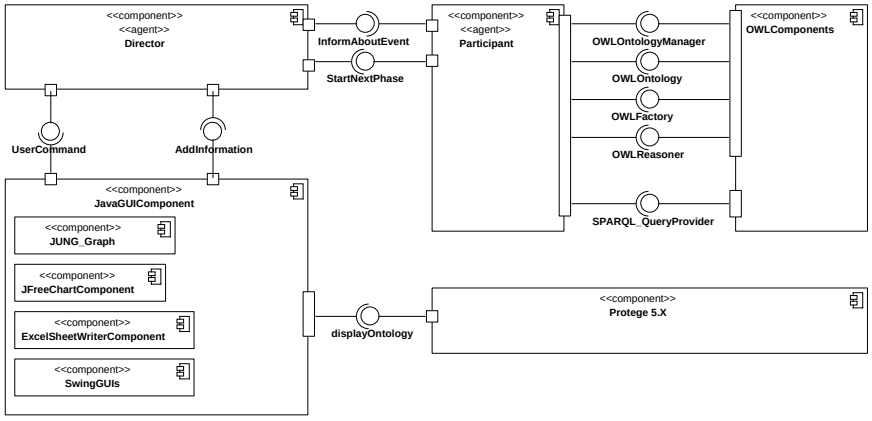

Fig. 3. Abstract view on the AGADE architecture.

4 Conclusion

The demonstration shows that multi-agent simulations can indeed be created by modelling agent behaviour through OWL ontologies and integrating that with the BDI concept. This reduces programming efforts as it is no longer necessary to code the complete agent logic in a programming language. Together with AGADE's ability to define social environments for the agents realistic simulations of real world scenarios can be built. Proof of concept was given in a case study with a scenario where agents are part of a typical social structure (small world network) [2]. The tool is designed to carry arbitrary simulations as long as the underlying structures of the scenario can be modelled by means of ontologies (individual aspect) and social structure.

References

1. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Reviews of Modern Physics 74(1), 47–97 (2002)
2. Farrenkopf, T., Guckert, M., Hoffmann, B., Urquhart, N.: Agade - how individual guidance leads to group behaviour and how this can be simulated. In: Müller, J., Weyrich, M., Bazzan, A. (eds.) Multiagent System Technologies, Lecture Notes in Computer Science, vol. 8732, pp. 234–250. Springer International Publishing (2014)
3. Horridge, M., Bechhofer, S.: The owl api: A java api for owl ontologies. Semantic Web 2(1), 11–21 (2011)
4. Pokahr, A., Braubach, L., Jander, K.: The jadex project: Programming model. In: Ganzha, M., Jain, L.C. (eds.) Multiagent Systems and Applications, Intelligent Systems Reference Library, vol. 45, pp. 21–53. Springer Berlin Heidelberg (2013)