ECAI 2002

JULY 21-26 2002, LYON, FRANCE

Organised by the European Coordinating Committee on Artificial Intelligence (ECCAI) and the Association Française pour l'Intelligence Artificielle (AFIA). Hosted by Claude Bernard University and INSA, Lyon.

15th European Conference on Artificial Intelligence

Proceedings

Edited by
Frank van Harmelen
Vrije Universiteit, Amsterdam

IOS Press
Ohmsha

Evolutionary Computation in MAS Design

Carles Sierra, Jordi Sabater, Jaume Agusti, Pere Garcia [1]

Abstract. This paper explores the existing gap between multiagent specification and implementation and the potential help that evolutionary programming techniques can bring in. We present a methodology to help the programmer in the transition from a set of desired global properties expressed as an equation-based model that a Multi-Agent System (MAS) must fullfil to an actual society of interacting agents. The evolutionary techniques are seen, within this methodology, as a procedure to tune the parameters of the population of agents in order that their aggregated behaviour maximaly approaches the desired global properties.

1 Introduction

The general goal of the research reported in this paper is to better understand the dynamics of large Multi-Agent Systems (or MAS, for short) with globally distributed and interconnected collections of human, software and hardware systems; each one of which with potentially thousands of components.

Within this ambitious goal this paper will explore two ideas. First, a particular approach to the principled design of MAS using Equation-Based Models (EBM) as a high level specification method, where equations model the aggregated behaviour of the agent populations abstracting from the interaction details of individual agents. Second, the use of evolutionary computation techniques to find out what agent structures produce the global emergent behaviour specified in the EBM maintaining certain restrictions in the design of the agents. These ideas will be framed within a design methodology called SADDE (Social Agents Design Driven by Equations).

In Section 2 we'll introduce the main steps in the SADDE design methodology. Section 3 will introduce a running example and will exemplify the methodology introduced in Section 2. Section 4 will extend the methodology with a step based on evolutionary computation whose concrete application to the example introduced in Section 3 will be done in Section 5. Section 6 concludes the paper, discussing it.

2 The SADDE Methodology

We take the stance that in order to build a model for a society containing thousands or millions of agents, the general view provided by an EBM provides succint descriptions of population-level behaviours which we then attempt to replicate using models consisting of a society of individual interacting agents, that is, the ABM. Our proposed lifecycle is graphically depicted in Figure 1.

An important characteristic of MASs design from a software engineering perspective is the decoupling of the interaction process between agents from the deliberative/reactive activity within each agent, [2, 17]. The notion of *electronic institution* [8, 10], as described in Section 3.2, plays this role in our methodology by establishing a framework that constraints and enforces the acceptable behaviour of agents.

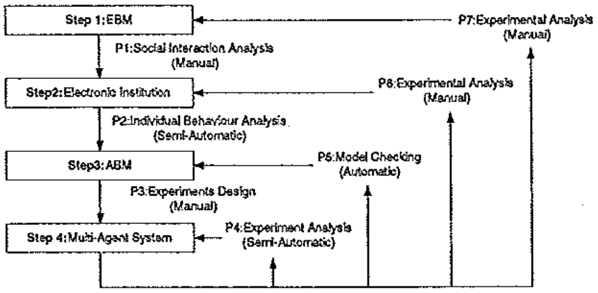

Figure 1. SADDE Methodology

The different phases within SADDE are:

[Step 1] **EBM – Equation-Based Model.** In this first step, a set of state variables and equations relating them must be identified. These equations have to model the desired global behaviour of the agent society and will not contain references to individuals of that society. Typically these variables will refer to values in the environment and to averages of predictions for observable variables of the agents. We model yet-to-exist artificial systems. The EBM is the starting point of the construction of a system that later on will be observed. Thus, a comparison between the EBM predicted behaviour and the actual ABM behaviour will be obtained.

[Step 2] **EIM – Electronic Institution Model.** In this step the interactions among agents are the focus. It is a first "zoom in" of the methodology from the global view towards the individual models. This step is not a refinement of the EBM but the design of a set of social interaction norms that are consistent with the relations established at Step 1.

[Step 3] **ABM – Agent-Based Model.** Here, we focus in the individual. We have to decide what decision models to use. This is the second "zoom in" of the methodology. New elements of the requirement analysis (new variables) will be taken into account here. For instance, some rationality principles associated to agents (*e.g.* producers do not sell below production costs), or negotiation models to be used (*e.g.* as those proposed in [14]) have to be selected.

[Step 4] **Multi-Agent System.** Finally, the last step of our methodology consists on the design of experiments for the interaction of very large numbers of agents designed in the previous step. For each type of agent the number of individuals and the concrete setting for the parameters will be the matter of decision here. The results of these experiments will

[1] Institut d'Investigacio en IA - CSIC, Campus UAB, Bellaterra, Barcelona, sierra,jsabater,agusti,pere@iiia.csic.es

ECAI 2002
F. van Harmelen (Ed.)
IOS Press, 2002

determine whether the requirements of the artificial society so constructed have been consistently interpreted throughout the methodology and thus whether the expected results according to the EBM are confirmed or not.

Once the experiments designed at Step 4 are run and analysed, several redesigns are possible as shown schematically in figure 1. In this paper we focus on the use of evolutionary computation to explore the space of possible MAS configurations. Section 4 addresses this. Further details on the SADDE methodology can be found at [16].

3 Supply chain example

Supply chains have been a traditional focus of attention in the design of multiagent systems [13] because of their important role in the structuring of the manufacturing economy and because they are a naturally distributed system where agents try to maximize their own profit, therefore permitting a classical economical analysis of their strategies. Thus, a MAS in supply chain will typically consist of a group of selfish agents that will trade by buying one level below in the chain, adding value to the purchased goods, and selling the manufactured good up to the next level in the chain. Although the real model is a supply tree or a supply graph, a simple supply chain is rich enough to show the potential complexities of a MAS design process. In this section we will follow the methodology presented in Section 2 to illustrate a MAS design —although necessarily in a summarised way.

3.1 Description of the Supply Chain EBM

In Figure 2 we see an example of a supply chain consisting of three levels: S_1, processing rough materials to produce goods to be sold to level S_2 which, in turn, processes the goods bought to S_1 to sell to the final consumers represented as level S_3. Each level is pictured as a rounded box that contains the model of the behaviour of that level in the chain. It should be emphasized that by behaviour we do not mean the behaviour of an agent at that level; rather, we mean a global summary of multitude of agents placed in that level of the chain.

The meaning of the variables in the example is quite straightforward. We classify them in three groups:

ENVIRONMENT VARIABLES: Variables whose value is fixed outside the EBM. The MAS designer can manipulate them only to see how the model reacts in view of environmental changes. In our example they are:

- *RoughMat* represents the generation of primary goods.
- *Demand* and *MinPrice* generate the price for the good.

MODEL VARIABLES: Their value is computed as a function over other variables or constants in the model, or is fixed by the MAS designer.

- $MaxStockIn_i$ represents the maximum storage capacity for income goods at level i.
- $MaxStockOut_i$ represents the maximum storage capacity for produced goods at level i.
- $ProdRate_i$ is the production rate in number of processed units per time unit at level i.
- $delay_i$ is the time required to transform a unit of good at level i.
- $PriceIn_i$ is the market price that level i in the chain would be willing to pay to level $i-1$ for a unit of good. We made it depend on the stocks at level i.
- $PriceOut_i$ is the market price at which level i in the chain would be willing to sell to level $i+1$ a unit of good. We made it depend on the stock of produced material, the maximum storage capacity and the price payed at level i $delay_i$ units of time ago.

STATE VARIABLES: Their value changes along time and they represent the system observable variables whose dynamic behaviour the MAS designer is interested in. We have:

- $StockIn_i$ represents the current stock of income goods at level i.
- $StockOut_i$ represents the current stock of produced goods at level i.
- $Cash_i$ represents the liquidity at level i of the supply chain.

The reader can use different tools to implement and then observe the behaviour of the system (we have used SIMILE [15]). The collective behaviour shows two main properties:

1. There is a moderate linear increase of the cash at levels S_1 and S_2 of the chain, and
2. There is a positive flow of goods along the chain, that is, there is commerce being made.

3.2 An Electronic Institution for supply chain

The next step in the methodology after the EBM has been specified is the specification of the electronic institution that will give structure to the interaction among the individuals [7]. Each chain level is a global view of a reality consisting of many individual agents, and the transactions modelled as flows of goods and money in the EBM are the result of the social interaction of the agents following particularly well established conventions. In our Institution specification we consider two scenes: One, *primary_market*, for the interaction between agents of levels S_1 and S_2 in figure 2 and another, *retailing*, for the interaction between S_2 and S_3. See in Figure 3 the performative structure of such institution. The links between scenes indicate the flux of agents; *root* is the initial scene to enter the institution, and *final* is the scene through which all agents leave the institution. For further details on the concept of Electronic Institution see [8, 10].

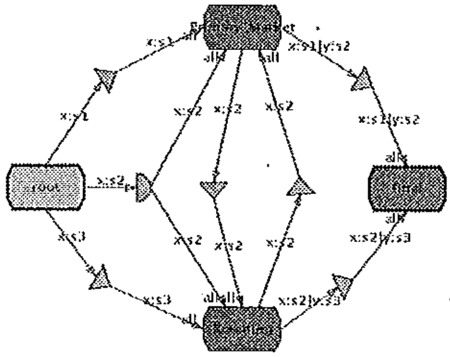

Figure 3. Supply chain scenes

By means of this specification we are, for instance, prescribing that there will be no interactions between agents of

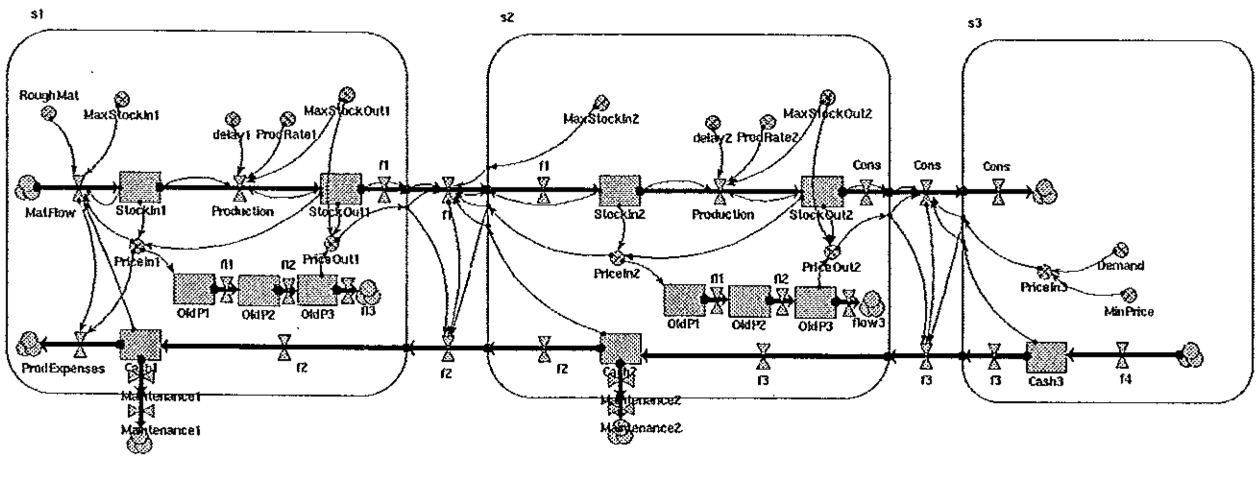

Figure 2. Diagrammatic representation of a supply chain.

S_1 and S_3 within the Institution. Also, we are fixing that the interaction in the two scenes will be based on a negotiation protocol as specified in [1] and that is presented in Figure 4.

3.3 Agents for the supply chain

The mechanism that the agents use to buy and sell products in the so specified institution is then a negotiation one-to-one as explained in detail in [1].

The agent model must specify a range of strategies and tactics that agents can employ to generate initial offers, evaluate proposals and offer counter proposals. To simplify, we only allow our agents the use of time-dependent tactics. In these tactics, the predominant factor used to decide which value to offer next is time. Thus these tactics basically consist of varying the acceptance value for an issue in the contract under negotiation depending on the remaining negotiation time. In our agent model, the negotiation issues are reduced to "price". We define t^a_{max} as the maximum time that an agent a can spend for a single negotiation process and β is the factor that fixes the type of tactic. The β factor modifies the shape of the function wich makes the dependence of price on time. When $\beta < 1$ we talk about *boulware* tactics. These tactics maintain the offered price until the time is almost exhausted, whereupon it concedes up to the reservation price. When $\beta > 1$ we talk about *conceder* tactics.

Figure 5 shows a schema of the internal structure of

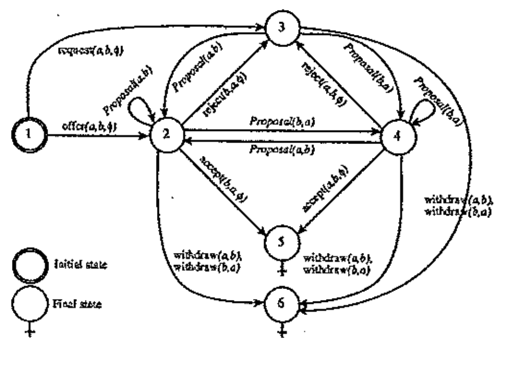

Figure 4. Agents' negotiation protocol.

each agent. The clouds with the word D-Making-In and D-Making-Out represent the decision making part of the agent. Using the negotiation model explained before, each agent tries to reach an agreement with the other partner to buy/sell the product. The parameters for this decision making processes are the parameters for the negotiation model, this is: $t$max and β. The rest of variables have the same meaning as in the EBM model but applied to a single individual instead as a full population.

4 Evolutionary programming within SADDE

Once an ABM is generated according to the SADDE methodology, what we have is a precisely defined way of interaction between agents, as restricted by the electronic Institution, and schemes of individual behaviour (determined by a concrete decision making system) of the agents playing the different roles. But there are still two important decisions to be made in order to have a running MAS: what values to assign to the parameters of the decision making apparatus of the agents, and what proportions of significantly different individual behaviours to use in order to conform to the MAS.

Evolutionary computing [5, 4] is the technique used to explore the space of possible configurations of MAS populations. In this respect we follow here the Pittsburgh approach [6] where an individual in the Genetic Algorithm corresponds, in our case, to the genetic material of a complete MAS population. Crossover between populations will mean that subsequent generations will explore the space of agent combinations and that mutation will, basically, generate new agents by mutating the parameters of the decision making apparatus of a particular agent. From the study of such populations we expect to obtain insights about the structure of the agents and their social interelationships in relation to the global behaviour. This study would eventually lead to the generation of a series of design rules that could reduce the currently existing gap between specification and implementation of MAS.

Figure 6 illustrates graphically the intended role of evolutionary computation. We want to use as the initial population of the evolutionary computation algorithm a set of MASs that fit with the schema obtained through the SADDE methodol-

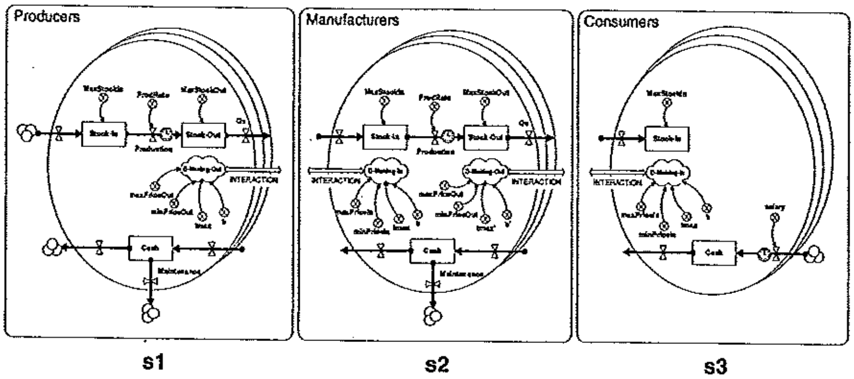

Figure 5. Agent models for the supply chain example.

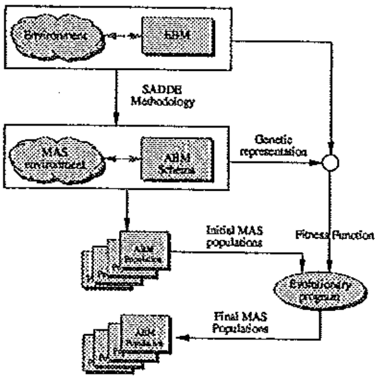

Figure 6. Evolutionary computing within SADDE.

ogy, and then use evolutionary computation to obtain a set of MASs that fit optimally with the EBM. It is thus natural that the fitness function is provided by the EBM in terms of the concrete genetic coding used. In our approach, one chromosome is the specification of a full MAS and the parameters that specify a single agent are codified in a gen of that chromosome. The parameters that represent an agent are the parameters of the negotiation model. Therefore a chromosome is a sequence of t_{max} and β.

5 Evolutionary computing in the Supply Chain example

We exemplify in this section the use of evolutionary computing in the particular context of our running example.

5.1 Fitness functions

One of the key design issues in the proposed methodology is how to obtain a fitness function from the global behaviour, as expressed in the EBM, and the behaviour of the agents as specified in the ABM. The right choice is essential to improve from the initially designed ABM populations into better ABM populations that fit the overall objective of guaranteeing certain properties of the societies that satisfy the EBM.

To illustrate this let us consider the Supply Chain example again. The EBM properties we want to match with the agent society are those explained in section 3.1. In order to determine the fitness function we have to put in relation these global properties with individual variables so that by selecting MASs that maximize some functions over those variables we approach the desired global behaviour. In general, if we have a set of properties we want the MAS to satisfy along time and we model each property to be satisfied as a function over time and a vector of state variables in the EBM, $f_i(t, \mathbf{X}_i)$, and we model the observed behaviour of the aggregated individual variables, $\mathbf{Y}_i$, corresponding to $\mathbf{X}_i$, as $h_i(t, \mathbf{Y}_i)$, we can define a fitness function as a weighted ($omega_i$) mean over a comparison function between the two along time:

$$f(EBM, ABM) = \sum_{i=1}^{n} \omega_i \cdot \sum_{0 \le t \le T} g(f_i(t, \mathbf{X}_i), h_i(t, \mathbf{Y}_i))$$

For instance, in our supply chain example, we want to satisfy the two properties refered at the end of subsection 3.1 that are modelled by two functions obtained from the execution of the EBM model. The variables used for those two functions are $X_1 = \langle Cash_1, Cash_2, Cash_3 \rangle$ and $X_2 = \langle Stock_1, Stock_2, Stock_3 \rangle$ with their corresponding functions, $f_1(t, X_1) = f_1(t, \langle Cash_1, Cash_2, Cash_3 \rangle) = Cash_1(t) + Cash_2(t) + Cash_3(t)$. Similarly for f_2. The corresponding Y_i vectors will correspond to the set of individual variables for cash and stock respectively, and the h_i functions will add up the values of the individual variables. In the experiments we have performed we have used a quadratic means error to compare the two behaviours. That is, $g(a, b) = \sqrt{|a - b|^2}$.

5.2 Genetic Algorithm settings

For a general introduction to GAs refer to [3]. We have made initial experiments with a population of $N = 30$ individuals, each individual representing $A = 60$ agents, with $\frac{A}{6}$ agents at the S_1 level of the chain, $\frac{A}{3}$ agents at the S_2 level, and finally $\frac{A}{2}$ agents acting as consumers. We have used this increased population distribution to make the co-evolution more realistic with respect to many real supply chains where the amount of manufacturers and final consumers increases along the chain. The exploration of other values is part of the future work. Each step in the algorithm consists of T iterations of the following process: for each MAS of the N MAS in the population do the following, for each agent at level S_3 randomly choose an agent in S_2 and engage them in a negotiation process, and for each agent at S_2 randomly choose an agent in S_1 and also engage them in a negotiation process. We have chosen $T = 10$. The algorithm terminates when the average fitness of the MAS individuals does not significantly change in 5 iterations. Each step consists then on $N * T * (\frac{A}{2} + \frac{A}{3})$ negotiations, that is 24900 negotiation processes per generation.

The genetic material is the sequence of agent parameters of each one of the agents in a MAS population as explained in section 4. The mutation probability is set to $p = 0.1$ and there are as many cross-over points as agents with a probability of cross-over of $p_c = 0.25$. We consider that a 10% of mutation is a good trade-off between exploitation and exploration. Finally, the selection is elitist.

6 Discussion and results

EBM and ABM are two well known styles of computer based modelling. EBM allows the modelling of the global behaviour of a population leaving implicit the behaviour and interaction of individuals. On the other hand in ABM we model explicitly these individuals and their interactions leaving the global behaviour of the population as an emergent result. There are numerous applications of each of these approaches [12, 11]. They have even been applied to the same problem in order to establish comparative criteria about their alternative use [9]. This competing view between EBM and ABM makes sense if you have a real system against which the model you build should be checked. However if the goal is to build an artificial system whose behaviour is to be inspired by a real system but not bound to simulate it faithfully, then the reasonable attitude is to take EBM and ABM as complementary approaches to be used at different levels of abstraction in the design lifecycle.

We have integrated both approaches into a methodology for MAS design and implementation. More specifically we have used EBM to identify desired global properties of the MAS.In the supply chain example these are the moderate increase in cash and the positive flow of goods.Then we analyzed how the flows of the EBM could be produced by the interactions between different types of agent. The structure of the EBM guided the definition of these interactions through an electronic institution. We then decided on the agent model we expect that will allow populations whose aggregate behaviour will meet the EBM. In our example this comprises the range of strategies and tactics that agents can employ to negotiate. These are reduced to two parameters, the maximum time of negotiation and the type of tactic, which are the genes of each agent in the MAS when exploring the space of models using evolutionary computing.

The application of this technique to a collection of MAS brings us two main preliminary results. First, the chosen agent model allows the convergence of the evolutionary process towards the production of a stable collection of MASs showing the EBM specified properties to an acceptable degree. Second, from the analysis of the distribution of the values of the parameters in each MAS we can establish a first design rule which relates them with the global properties specified by the EBM. Its informal expression can be put as follows. Slow agents can not be part of any MAS satisfying the mentioned properties of the supply chain while generous and greedy agents can coexist.

This is our first attempt to incorporate evolutionary computing in the SADDE methodology. We simplified the problem and reduced the agent model parameters to a minimum. We are aware that the design rule obtained does not brings much new insight on agents design. It was in fact foreseen from the beginning. However it helped to check the applicability of the approach. In the future we plan to work with a more complete supply chain in order to obtain a more informative set of design rules. We also plan to extend further the application of evolutionary computing by representing the decision processes themselves as genetic material. Finally, the electronic institution itself could also be represented as genetic material. In this growing complexity of the representation we go all the way back from the ABM to the second step in the methodology, that of the electronic institution specification.

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish CICYT project eINSTITUTOR (TIC2000-1414), and by the European research project SLIE (IST-1999-10948).

REFERENCES

[1] P. Faratin, C. Sierra, and N.R. Jennings, 'Negotiation decision functions for autonomous agents', *Robotics and Autonomous Systems*, (24), 159–182, (1997).

[2] J. Ferber and O. Gutknecht, 'A meta-model for the analisys of organizations in multi-agent systems', in *ICMAS98*, pp. 128–135, ICMAS98Addr, (1998).

[3] E. Fiesler and R. (eds.) Beale, *Handbook of Neural Computation*, IOP Publishing Ltd and Oxford University Press, 1997.

[4] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley.

[5] J. Holland, *Adaptation in Natural and Artificial Systems*, MI: University of Michigan Press.

[6] C.G. Langton, *Artificial Life*, Addison-Wesley, 1989.

[7] P. Noriega, *Agent-Mediated Auctions: The Fishmarket Metaphor*, number 8 in IIIA Monograph Series, Institut d'Investigació en Intel.ligència Artificial (IIIA), 1997. PhD Thesis.

[8] P. Noriega and C. Sierra, *Auctions and Multi-agent Systems*, 153–175, Springer, 1999.

[9] H.V.D. Parunak, R. Savit, and R.L. Riolo, 'Agent-based modeling vs. equation-based modeling: A case study and users' guide', in *Proceedings of Workshop on Modeling Agent Based Systems, MABS98, Paris*, pp. 1–15, (1998).

[10] J.A. Rodríguez, F.J. Martin, P. Noriega, P. Garcia, and C. Sierra, 'Towards a test-bed for trading agents in electronic auction markets', *AI Communications*, **11**(1), 5–19, (1998).

[11] J.A. Rodríguez-Aguilar, F. Martín, P. Noriega, P. Garcia, and C. Sierra, 'Towards a test-bed for trading agents in electronic auction markets', *AI Communications*, **11**(1), 5–19, (1998).

[12] M. Ruth and B. Hannon, *Modelling Dynamic Economic Systems*, Springer, 1997.

[13] N. Sadeh-Koniecpol, D. Hildum, D. Kjensta, and A. Tseng, 'Mascot: An agent-based architecture for coordinated mixed-initiative supply chain planning and scheduling', in *Workshop notes, Agent-Based Decision Support for Managing the Internet-Enabled Supply Chain, Third International Conference on Autonomous Agents (Agents '99)*, (1999).

[14] C. Sierra, P. Faratin, and N.R. Jennings, 'A service-oriented negotiation model between autonomous agents', in *MAAMAW'97*, number 1237 in LNAI, pp. 17–35, Ronneby, Sweden, (1997).

[15] SIMILE Web Page. http://www.simile.co.uk.

[16] W. W. Vasconcelos, D. Robertson, J. Agustí, C. Sierra, M. Wooldridge, S. Parsons, C. Walton, and J. Sabater, 'A lifecycle for models of large multi-agent systems', in *Proc. AOSE*, volume to be published of *Lect. Notes in Comp. Sc.*, Springer-Verlag, (2002).

[17] M. Wooldridge, N.R. Jennings, and D. Kinny, 'The gaia methodology for agent-oriented analysis and design', *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3), 285–312, (2000).