A System Implementing Cooperation in Multi-Agent Networks *

F. Buccafurri, L. Palopoli, D. Rosaci and G. M. L. Sarnè
DIMET, Università "Mediterranea" di Reggio Calabria,
Via Graziella Loc. Feo di Vito, 89100 Reggio Calabria (Italy)
E-mail: {bucca,palopoli,rosaci,sarne}@ing.unirc.it

Abstract

In Multi-Agent Systems the main goal is providing fruitful cooperation among agents in order to enrich the support given to user activities. Cooperation can be implemented in many ways, depending on how local knowledge of agents is represented and consists in general in providing to the user an integrated view of individual knowledge bases. But the main difficulty is determining which agents are promising candidates for a fruitful cooperation among the (possibly large) universe of agents operating in the net. This paper gives a contribution in this context, by proposing a system for supporting cooperation in multi-agents networks. Semantic properties are here represented by coefficients and adaptive algorithms permit the computation of a set of suggested agents for cooperation. Actual choices of the users modify internal parameters in such a way that the next suggestions are closer to users expectancy.

1. Introduction

Coordinating the activities of multiple agents is a basic task for the viability of any system in which such agents coexist. Each agent in an agent community does not have to learn only by its own discovery, but also through a cooperation with other agents, by sharing individual learned knowledge. And practical situations need to be considered in an agent community. For example, an agent has often different objectives to achieve w.r.t. another agent, and no real intention to help it. In other cases, an agent a may have quite similar objectives as another agent b, but it can be also very busy in satisfying many requests, so b may prefer not to contact it for avoiding a long waiting time. Moreover, the problem of integrating heterogeneous knowledge bases has to be considered in order to implement cooperation [2, 1].

Many Multi-Agent Systems (hereafter MASs) have been recently proposed in the literature. In particular, cooperation is often considered as one of the key concepts of agent communities [5, 6, 7]. Researchers in Intelligent Agent Systems have recognized that learning and adaptation are essential mechanisms by which agents can evolve coordinated behaviours finalized to meet the knowledge of the interest domain and the requirements of the individual agents [10, 9]. In order to realize such a cooperation, some techniques developed in the field of Machine Learning has been introduced in various MASs [8, 4].

In such a context, this paper describes a new MAS, called SPY, able to inform the individual agent of a multi-agent network about which agents are the most appropriate to be contacted for possible knowledge integration.

2. Main Contributions

The main contributions of this paper are the following:

1. We point out some properties useful for driving the integration of the knowledge coming from non local agents and provide a framework in which such properties are represented as quantitative information by mean of a number of real coefficients.

2. By exploiting the above properties, we design an adaptive method for determining, for a given agent a of a multi-agent net, the most appropriate agents to cooperate with a. Such a method is adaptive in the sense that it takes into account some reactive properties of users, and, as such, its result *depends* on their behaviour.

3. On the basis of this model we design the architecture of a system, called SPY, supporting cooperation of agents operating in a multi-agent network. The main function of the system is providing the user with a number of agent lists, each containing the *most appropriate* agents for cooperation, from which the user can choice agents she/he want to contact for supporting her/his activity. The multiplicity of such choice lists depends on the multiplicity of the properties that can be used as preference criteria. Users are free to also partially

*This paper is a short version of the full report [3].

use the suggested lists, or can ignore them. In any case, user's behaviour induces a modification of some coefficients (describing reactive properties) in such a way that lists suggested in the future by the system are (hopefully) closer to real user needs. Therefore, the system *learns* from user's behaviour about how to provide the users with suggestions meeting as much as possible their expectancy.

3. The Model

3.1. Ontology and Local Knowledge

The ontology represents the common knowledge about the environment in which the agents work. However, each agent may have a partial view of the ontology representing the portion of the world the user monitored by the agent selects by her/his activity. Inside this portion of the ontology, different priorities for the classes can be inferred by exploiting user behaviour. This is encoded in the notion of *the Local Knowledge Base* (LKB for short). A Local Knowledge Base, representing the local view of the agent, is then obtained by extracting a sub-graph from the ontology graph including all the classes accessed by the user (and thus at least the root node). Moreover, arcs of the so obtained graph are weighted for assigning highest priority to most accessed classes.

3.2. Extraction of the Semantic Properties

Besides his/her local agent, each user looks at the other agents of the net as a source of potentially interesting information in order to enrich the support to his/her activity. Interest in agents can be defined by considering some semantic properties. Such properties, useful for driving users' choices are of two types: (i) *local properties*, taking into account information stored in the LKBs, and (ii) *global properties*, merging local properties with external knowledge extracted from the general context.

An important feature of the model is that the merge performed in the construction of global properties is based on adaptive learning involving some parameters taking into account user behaviour. In other words, global properties exploit an important kind of properties (encoded by some parameters) directly reflecting reactions of users to system advice. We call such additional properties *reactive properties*. Next we describe the set of properties used in the model.

The only local property we consider is the property we call *similarity* between two agents i and j, representing a measure of the similarity of the two corresponding LKBs. Such a coefficient is a real value ranging from 0 to 1.

Global properties merge local properties with knowledge extracted from the context. We introduce the notion of *interest coefficient*, representing just a measure of the global properties of a given agent as perceived by another one. Hence, for a pair of agents i and j, the interest coefficient, besides of the similarity between i and j, must take into account also knowledge extracted from the context. But *which kind of contextual knowledge has to be considered as meaningful?*. The choice we make in our model is the following: The knowledge extracted from the context, used by the agent i for defining the interest coefficient I_{ij} w.r.t. another agent j, is a measure of the global interest of all the other agents (different from i) w.r.t. the agent j, that is a measure of a sort of *attractiveness* of the agent j as perceived by the agent i .

Besides of the interest property, from the knowledge of the interest coefficients lists, agents can exploits a second type of property. Indeed, an agent can compare different agents on the basis of their *attractiveness coefficient*, representing the component of the interest capturing only the contextual knowledge.

Choice Lists. Suppose the user of an agent i has the intention of contacting other agents in order to establish a cooperation. Suppose the similarities between i and every other agent is known as well as both the interest coefficient of i w.r.t. every other agent and the attractiveness of all the agents perceived by i. Such values can be effectively computed once a number of parameter are set (actually, they are suitably initialized and their updating is learnt from the behaviour of the user). Thus, three agent lists can be presented to the user i associated to the agent i, each associated with a property among similarity, interest and attractiveness. We denote these lists $L_S(i)$, $L_I(i)$, and $L_A(i)$. $L_S(i)$ ($L_I(i)$, $L_A(i)$, resp.) is the list of the $n-1$ agents j (different from i) ordered by decreasing similarity (interest, attractiveness, resp.) coefficient S_{ij} (I_{ij}, A_{ij}, resp.). When the user i chooses an agent j from the list $L_S(i)$ ($L_I(i)$, $L_A(i)$, resp.), it means that she/he perceived only the property of similarity (interest, attractiveness, resp.) about the agent j. From the choices of the users, useful knowledge can be thus drawn, which is potentially usable as feedback for correcting advice given to them. This issue is sketched in the following.

Reactive Properties. For reactive properties we mean properties describing reactions of users to the suggestions received from the system at a given time, that must be taken into account for adapting future responses of the system. We implement such adaptation of the system to the user behaviour by including into the interest coefficient definition some specific coefficients that are automatically updated during system running depending on to the user behaviour. Properties considered are: the preference property, the benevolence property and the consent property.

The Preference Property. It is described by a real coefficient ranging from 0 to 1 denoted by P_i and called *pref-*

erence coefficient. The property measures how much for an agent i the similarity is more important than the attractiveness property for defining global properties. Now we define how the coefficient P_i is updated. Suppose that at a given time the user of the agent i makes a selection of agents. Denote by SI_i (SS_i, SA_i, resp.) the set of the agents that the user has selected from the list $L_S(i)$ ($L_I(i)$, $L_A(i)$, resp.). We interpret the behaviour of the user in the following way. The choice of an agent from a list, say $L_S(i)$, means that the user relies on the associated property, say similarity. We can then interpret the former choice as an implicit suggestion from the user to set the coefficient P_i to 1, while the latter as an implicit suggestion to set this value to 0. In case the user chooses from the list $L_I(i)$, we infer that the user accept the current value the coefficient P_i.

The Benevolence Property. This property measures a sort of availability of the agent j to which a user i requires to share knowledge. Such a property is used in order to weight the interest of i w.r.t. j. For instance, an agent j that recently, and for several times, has denied collaboration in favor of i should become of little interest for i. The parameter encoding such a knowledge is called *benevolence coefficient*, denoted by B_{ij}, and takes real values ranging from 0 to 1. $B_{ij} = 0$ (resp., $B_{ij} = 1$) means the agent j is completely unavailable (resp., available) to fulfill the requests of i. The response of j to requests of i updates the value of B_{ij} according to the following rules:

$$B_{ij} = \begin{cases} min(1, B_{ij} + \delta) & \text{if } j \text{ grants the request of } i \\ max(1, B_{ij} - \delta) & \text{if } j \text{ denies the request of } i \end{cases}$$

where δ is a (reasonably small) positive real value.

The Consent Property. This property describes how much the user of an agent i trusts suggestions of the system regarding another agent j done on the basis of the interest property. The coefficient associated with this property is denoted by C_{ij} and is called *consent coefficient*. The updating rules defining how to adapt the coefficients C_{ij} after a user selection step take into account only the portion of the selection performed on the list $L_I(i)$.

3.3. Integration of Local Knowledge Bases

Cooperation between two agents is implemented in our model by the integration of their LKBs. Thus, the user of an agent i which has selected an agent j from one of the three choice lists can exploit the cooperation of j by consulting the *Integrated Knowledge Base*, obtained by integrating the LKB of i with the LKB of j. Clearly, the integration process must take into account which choice list the user of i has used for selecting j.

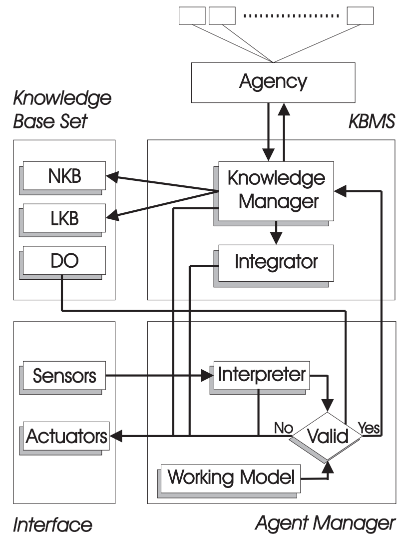

Figure 1. The SPY System

4. Discussion and Conclusion

The model is based on the extraction of some semantic properties capturing both local and contextual knowledge about agents. Such properties, encoded by suitable coefficients, drive users on selecting from the agent net the most promising candidate agents for fruitful cooperation. User choices are exploited as feedback for adapting coefficients in such a way that a trade-off among similarity and attractiveness, on the one hand, agent congestion and user dissatisfaction, on the other hand, is obtained. As example, consider: (i) An agent a with high similarity and low attractiveness perceived by another agent b. The user of b can decide to contact a less similar, but more attractive, agent c, and this means that the current similarity does not fully satisfy b. Since b has chosen c, probably it will make choices more similar to those of c than to those of a, and the similarity between a and b will decrease, coherently with dissatisfaction of the user. (ii) An agent a with high interest and low similarity (or low attractiveness) perceived by another agent b. The user of b can decide to contact a less interesting, but more similar (or more attractive) agent c. As a consequence, the interest for a perceived by b will decrease, due to the decreasing of the consent coefficient C_{ba}. (iii) An agent a with high interest and high attractiveness perceived by another agent b. The user of b knows that high attractiveness means probably long waiting time for obtaining answers from a and can decide to contact a less interesting agent c. As a consequence, the interest of b for a will decrease.

On the basis of the formal model, we developed a system whose architecture is just quickly illustrated here. For a detailed presentation of the system see [3]. The system (see Figure 3) is composed of four different functional modules, namely: (A) the *Interface*, that allows bidirectional communications between user and agent, (B) the *Knowledge Bases*, containing the whole knowledge the agent has captured about the user, (C) the *Knowledge Base Management System*, answering to both update and query requests (for the latter, integration of local knowledge bases has to be also performed), (D) the *Agent Manager*, transforming user action detected by interface sensors into requests for the (C) module and vice versa.

References

[1] S. Adali, R. Emery. A uniform framework for integrating knowledge in heterogeneous knowledge systems. In *Proc. of the 12th IEEE International Conf. of Data Engineering*, 1995.

[2] S. Adali, V. S. Subrahmanian. Amalgamating knowledge bases III: algorithms, data structures, and query processing. *Journal of Logic Programming*, 28(1), pp 45-88, 1996.

[3] F. Buccafurri, L. Palopoli, D. Rosaci, G.M.L. Sarnè. The SPY system. *Quaderno di dipartimento n.1/01, DIMET Univ. Reggio C. (I)*, 2001, available from the authors.

[4] Y. S. Choi and S. I. Yoo: Multi-agent web information retrieval: neural network based approach. In *Proc. of the 1st International Symposium - IDA'99, Amsterdam (NL)*, ed. Berthold, Hand and Kok, Springer, 1999.

[5] J. E. Doran, S. Franklin, N. R. Jennings and T. J. Norman. On cooperation in multi-agent systems. *The Knowledge Engineering Revue*, 12(3), pp 309-314, 1997.

[6] M. Fisher, J. Muller, M. Schroeder, G. Staniford and G.Wagner. Methodological foundation for agent-based systems. *The Knowledge Engineering Revue*, 12(3), pp 323-329, 1997.

[7] G. Iglesias, M. Garijo, J. Gonzales and J. G. R. Velasco. Analysis and design of multiagent systems using MAS-common KADS. *Agent Theories, Architectures and Languages*, pp 313-327, 1997.

[8] A. Moukas and P.Maes. Amalthaea: an evolving multi-agent information filtering and discovery system for the WWW. *Autonomous Agents and Multi-agent Systems*, 1(1), pp 59-88, 1998.

[9] M. Tan. Multi-agent reinforcement learning: indipendent vs. cooperative agents. In *Proc. of the 10th International Conf. on Machine Learning*, pp 330-337, 1993.

[10] G. Wei. Adaptation and learning in multi-agent systems: some remarks and a bibliography. *Adaptation and Learning in Multi-agent Systems*, pp 1-21, ed. G. Wei and S. Sen, 1996.