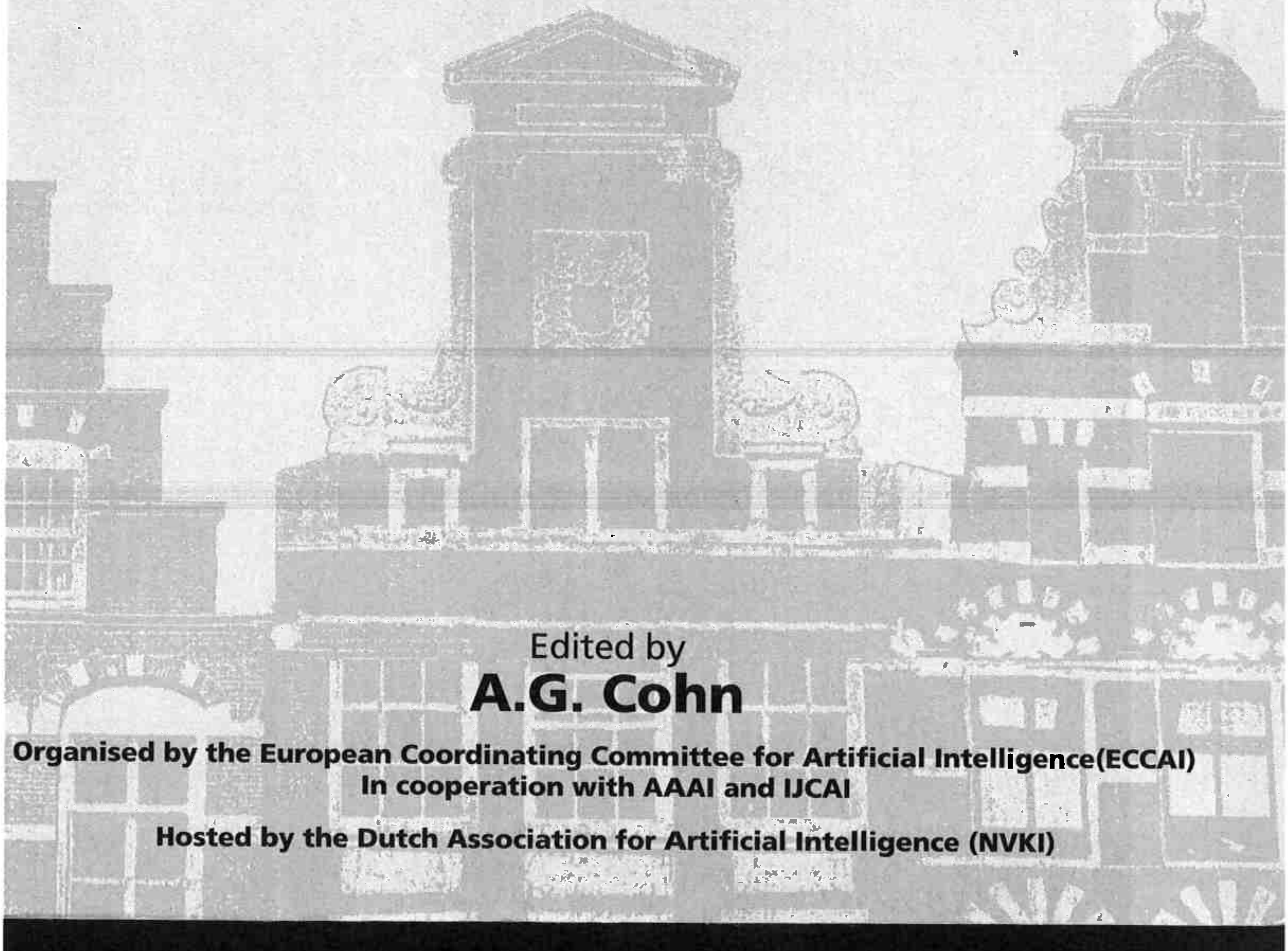
11th European Conference on Artificial Intelligence

ECAI 94

August 8-12, 1994, Amsterdam, The Netherlands

Proceedings



Edited by

A.G. Cohn

Organised by the European Coordinating Committee for Artificial Intelligence(ECCAI)
In cooperation with AAAI and IJCAI

Hosted by the Dutch Association for Artificial Intelligence (NVKI)

Index of papers

Application Technologies

Automated Reasoning

Cognitive Modelling

Connectionism and PDP

Distributed AI

A Social Reasoning Mechanism Based On Dependence Networks

Jaime Simão Sichman [1]
Yves Demazeau [1]

Rosaria Conte [2]
Cristiano Castelfranchi [2]

Abstract. This paper describes the fundamental concepts of a *social reasoning mechanism*, designed to be part of an agent's internal model, in a multi-agent systems (MAS) context. It enables an agent to reason about the others using information about their goals, actions, resources and plans. Every agent stores this information in a data structure called external description. We have formally defined and implemented the concepts of *external description, dependence relation*, and *dependence network*. One of the main contributions of this work is that an agent can infer his dependence on others using either his own plans or those of the others. As a result, we have defined a preliminary taxonomy of *dependence situations* regarding the goal being analysed (unilateral, mutual or reciprocal) and the sets of plans used in this reasoning mechanism (mutually or locally believed). We have used this model to build a dependence network simulator, called DEPNET, which is also briefly described in this paper.

Keywords: multi-agent systems, cognitive modelling, communication and cooperation, integrating several AI components.

1 Introduction

The main goal of this work, developed in a scientific cooperation program between the LIFIA/IMAG and the IP/CNR, was to combine the complementary expertise of these groups in the multi-agent systems (MAS) domain. In particular, we have designed and implemented a computational model of the Social Power Theory [4], using the concept of dependence relation [3]. This model, as well as some of its extensions, will be used in some modules of the LIFIA's MAS platform [2] and agent models [10], mainly involving conflict management in MAS. On the other hand, we have used this model to build a dependence network simulator, called DEPNET. This simulator will enable the IP/CNR's research staff to experiment and validate some future theoretical results.

The ability of reasoning about the others is an essential issue in a so called "intelligent" agent. Moreover, if we consider a MAS as an open system [7], this feature enables an agent to adapt himself to an evolving environment, to take into account information about new members of the agency (as an example, one can think of a robot's agency, when new robots may arrive and robots may leave due to failures). Therefore, in our point of view, an agent must have a *social reasoning mechanism* in order to react properly when faced to such situations. On the other hand, structural analysis approaches have been extensively used in the last years, specially in social and political sciences [8]. More recently, these approaches are beginning to be used in Computer Science. An interesting work, closely related to this paper, is described in [12], where a model of dependence was designed to treat business reengineering problems. The difference between this work and the one presented in this paper is that in the former some of the types of dependences introduced are closely related to the target domain, while our approach claims to be domain-independent.

In section 2, we present the concept of *external description*, a data structure where an agent stores the information he has about the others. This information may be used to infer his *dependence relations* and to construct his *dependence networks*, as it is shown in section 3. Once constructed such networks, an agent may identify which is his *dependence situation* regarding the other agents for a specific goal. A preliminary taxonomy of these dependence situations is described in detail in section 4. Section 5 briefly presents the DEPNET simulator, a software tool we have built in order to test our ideas. Finally, we present in section 6 our conclusions and further work.

2 External Description

As presented in [2] [10], we consider that an essential functionality an agent has to have in order to be really autonomous (in a broader sense) is a *social reasoning mechanism*. We call social any reasoning mechanism that uses *information about the others* in order to infer some conclusions. Therefore, any agent (despite the possible different internal models an agent may have) must have a data structure where this information about the others is stored. At the LIFIA/IMAG, we call such data structure an *external description* [6]. In this paper, we define it as composed of the following elements:

- *goals:* the goals an agent wants to achieve. An agent may have more than one goal, and in this point we do not make any reference if a goal is currently active or not, this discussion is out of scope of this paper;
- *actions:* the actions an agent is able to perform;
- *resources:* the resources an agent has control on;
- *plans:* the plans an agent has, using any actions and resources, in order to achieve a certain goal. These actions

[1] LIFIA/IMAG, 46 av. Félix Viallet, 38031 Grenoble Cedex France
[2] PSCS/IP/CNR, Viale Marx 15, 00137 Rome Italy

ECAI 94. *11th European Conference on Artificial Intelligence* Edited by A. Cohn
Published in 1994 by John Wiley & Sons, Ltd.

ıd resources do not necessarily belong to his own set of :tions and resources, and therefore an agent may *depend n others* in order to carry on a certain plan.

et us introduce a formal notation in order to describe concept. In the rest of this paper, we will denote ag_i . generic agent whose social reasoning mechanism we are lysing. Therefore, for an agent ag_i, his external description t_{ag_i} is defined as follows:

$$Ext_{ag_i} \stackrel{def}{\equiv} \bigcup_{j=1}^{n} Ext_{ag_i}(ag_j)$$

ɜre $Ext_{ag_i}(ag_j)$ is the corresponding entry in ag_i's external cription when the information about ag_j is stored. In the vious formula, we are supposing that the agency is com- ɜed of n agents, so each agent has n entries in his external cription, corresponding to each agent that belongs to the ɜncy, including himself.

An external description entry $Ext_{ag_i}(ag_j)$ is by its way ined as follows:

$$:t_{ag_i}(ag_j) \stackrel{def}{\equiv} \{G_{ag_i}(ag_j), A_{ag_i}(ag_j), R_{ag_i}(ag_j), P_{ag_i}(ag_j)\}$$

ıere $G_{ag_i}(ag_j)$ is the set of goals g_i , $A_{ag_i}(ag_j)$ is the set of tions a_i , $R_{ag_i}(ag_j)$ is the set of resources r_i and $P_{ag_i}(ag_j)$ the set of plans p_i ag_i believes ag_j has. At this point, ɜ must clarify that we are not interested in the moment how an agent can gather this information about the oth- s. This may be done by explicit communication whenever ı agent enters the agency (for instance, using an introduc- on protocol as described in [1]) or by the agent's perception echanisms. The important point is that this information orresponds to the beliefs an agents has regarding the oth- :s, and that these beliefs may be *neither necessarily true or complete*. Without lacking of generality, and in order ɔ illustrate the power of the proposed framework, we will dopt the hypothesis of external description compatibility $Ext_{ag_i}(ag_i) = Ext_{ag_j}(ag_i) \wedge Ext_{ag_i}(ag_j) = Ext_{ag_j}(ag_j)$) in he rest of this paper, when analysing the social reasoning nechanism of two related agents.

Regarding the description of the plans, $P_{ag_i}(ag_j, g_k)$ refers o the set of plans $p_{ag_{i_l}}(ag_j, g_k)$ ag_i believes that ag_j has in ırder to achieve the goal g_k. Each plan $p_{ag_{i_l}}(ag_j, g_k)$ can be ormally described as:

$$p_{ag_{i_l}}(ag_j, g_k) \stackrel{def}{\equiv} \{g_k, R(p_{ag_{i_l}}(ag_j, g_k)), I(p_{ag_{i_l}}(ag_j, g_k))\}$$

where g_k is the goal to be achieved by this plan, the second term $R(p_{ag_{i_l}}(ag_j, g_k))$ is a (possibly empty) set of resources used in this plan and $I(p_{ag_{i_l}}(ag_j, g_k))$ is a (possibly empty) sequence of instantiated actions used in this plan. By its way, each instantiated action $i_m(p_{ag_{i_l}}(ag_j, g_k))$ is defined as follows:

$$i_m(p_{ag_{i_l}}(ag_j, g_k)) \stackrel{def}{\equiv} \{a_m, R_{a_m}(p_{ag_{i_l}}(ag_j, g_k))\}$$

where a_m is an action and $R_{a_m}(p_{ag_{i_l}}(ag_j, g_k))$ is a (possibly empty) set of resources used in this action, with the following constraint $R_{a_m}(p_{ag_{i_l}}(ag_j, g_k)) \subseteq R(p_{ag_{i_l}}(ag_j, g_k))$.

For simplicity of notation, in the next sections we will drop the subscript referring to the generic agent ag_i we are analysing (for instance, we will use $G(ag_j)$ instead of $G_{ag_i}(ag_j)$). We will also use P_{jk} and p_{lk} respectively instead of $P_{ag_i}(ag_j, g_k)$ and $p_{ag_{i_l}}(ag_j, g_k)$. Whenever these notation assumptions are to be dropped, this fact will be explicitly stated.

3 Dependence Relations and Dependence Networks

Using the definitions cited above, an agent ag_i will be **a-autonomous** for a given goal g_k, *according to a set of plans* P_{qk} if there is a plan that achieves this goal in this set and every action appearing in this plan belongs to $A(ag_i)$:

$$a_{aut}(ag_i, g_k, P_{qk}) \stackrel{def}{\equiv} \exists g_k \in G(ag_i)\ \exists p_{lk} \in P_{qk}$$
$$\forall i_m(p_{lk}) \in I(p_{lk})\ a_m \in A(ag_i)$$

Analogously, an agent ag_i will be **r-autonomous** for a given goal g_k, *according to a set of plans* P_{qk} if:

$$r_{aut}(ag_i, g_k, P_{qk}) \stackrel{def}{\equiv} \exists g_k \in G(ag_i)\ \exists p_{lk} \in P_{qk}$$
$$\forall r_m \in R(p_{lk})\ r_m \in R(ag_i)$$

Finally, an agent an agent ag_i will be **s-autonomous** for a given goal g_k, *according to a set of plans* P_{qk} if he is both a-autonomous and r-autonomous for this goal:

$$s_{aut}(ag_i, g_k, P_{qk}) \stackrel{def}{\equiv} a_{aut}(ag_i, g_k, P_{qk}) \wedge r_{aut}(ag_i, g_k, P_{qk})$$

The major contribution of this work is the fact that this notion of autonomy is closely related to the set of plans used in the reasoning mechanism. In other words, an agent can use either *his own* set of plans (when $q = i$ in the definitions above) or *those of the others* in other to infer his own autonomy. Doing so, an agent can **simulate the reasoning of the others**, by using *their knowledge* (in our case, their plans).

If an agent is not autonomous for a given goal, he will depend on others to achieve it. An agent ag_i **a-depends** on another agent ag_j for a given goal g_k, according to a set of plans P_{qk} if he has g_k in his set of goals, he is not a-autonomous for g_k and there is a plan in P_{qk} that achieves g_k and at least one action used in this plan is in ag_j's set of actions. Formally:

$$a_{dep}(ag_i, ag_j, g_k, P_{qk}) \stackrel{def}{\equiv} \exists g_k \in G(ag_i) \neg a_{aut}(ag_i, g_k, P_{qk}) \wedge$$
$$\exists p_{lk} \in P_{qk} \exists i_m(p_{lk}) \in I(p_{lk}) a_m \in A(ag_j)$$

In a similar way, we define that an agent ag_i **r-depends** on another agent ag_j for a given goal g_k, according to a set of plans P_{qk} if:

$$r_{dep}(ag_i, ag_j, g_k, P_{qk}) \stackrel{def}{\equiv} \exists g_k \in G(ag_i) \neg r_{aut}(ag_i, g_k, P_{qk}) \wedge$$
$$\exists p_{lk} \in P_{qk} \exists r_m \in R(p_{lk}) r_m \in R(ag_j)$$

Finally, an agent ag_i **s-depends** on another agent ag_j for a given goal g_k, according to a set of plans P_{qk} if he either a-depends or r-depends on this latter:

$$s_{dep}(ag_i, ag_j, g_k, P_{qk}) \stackrel{def}{\equiv}$$
$$a_{dep}(ag_i, ag_j, g_k, P_{qk}) \vee r_{dep}(ag_i, ag_j, g_k, P_{qk})$$

Once defined these dependence relations, an agent can construct a *dependence network* to represent in a same structure all of his action/resource dependences regarding the others. These networks can be used later in the agent's social reasoning mechanism, in particular to detect the dependence situations regarding two agents for a given goal, as described in the next section.

4 The Dependence Situations

Since an agent has constructed his dependence networks, he can use this information when reasoning about the others. In other words, for a given goal g_k, an agent ag_i can calculate for each other agent ag_j which is the *dependence situation* relating them for this goal. A preliminary taxonomy of dependence situations are presented in figure 1. In the rest of this section, we will analyse these situations considering only a-dependences. We will call **mutual dependence** a situa-

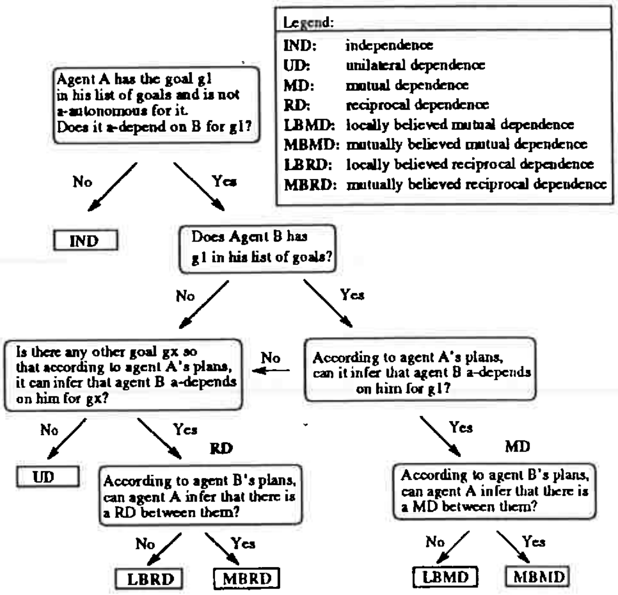


Figure 1. A Preliminary Taxonomy of Dependence Situations

tion where an agent ag_i infers that he and other agent ag_j a-depend on each other for the *same goal* g_k, according to a set of plans P_{qk}:

$$MD(ag_i, ag_j, g_k, P_{qk}) \stackrel{def}{\equiv} a_{dep}(ag_i, ag_j, g_k, P_{qk}) \wedge a_{dep}(ag_j, ag_i, g_k, P_{qk})$$

On the other hand, we will call **reciprocal dependence** a situation where an agent ag_i infers that he and other agent ag_j a-depend on each other, but for *different goals* g_k and g_l, according to the sets of plans P_{qk} and P_{ql} (both sets belonging to the same external description entry):

$$RD(ag_i, ag_j, g_k, g_l, P_{qk}, P_{ql}) \stackrel{def}{\equiv} a_{dep}(ag_i, ag_j, g_k, P_{qk}) \wedge a_{dep}(ag_j, ag_i, g_l, P_{ql}) \wedge g_l \neq g_k$$

In the case of mutual dependence, a possible *cooperation* regarding this goal can happen. On the other hand, in the case of reciprocal dependence, one of them will have to adopt the other's goal first in order to achieve his own one in the future. This mechanism is called *social exchange*. These concepts are better explained in [3] [4].

An agent **locally believes** a given dependence (either mutual or reciprocal) if he uses exclusively his *own plans* when reasoning about the others. If he uses both *his own plans* and *those of the others* to reach such a conclusion, it will be said that there is a **mutual believed** dependence between them.

The difference between the locally and the mutually believed situations (either in mutual or reciprocal dependence) is very subtle: in the first case, one of the agents may not be aware of this dependence (for instance, when one of them has a different set of plans to achieve the considered goal and in neither of these plans he a-depends on the other). In this case, a plan negotiation will have to be made in order to achieve the desired goal.

Let us consider two agents ag_i and ag_j. If $g_k \in G(ag_i)$ and $\neg a_{aut}(ag_i, g_k, P_{ik})$, there are six different dependence situations which may occur, considering ag_i's reasoning mechanism. These situations are described next, and we will use as an example to illustrate them the external description presented in table 1, corresponding to an agency composed of 7 agents (*jaime, rosaria, cristiano, vittorio, maria, amedeo* and *paola*):

Table 1. Example of an external description

ag_j	$G(ag_j)$	$A(ag_j)$	$R(ag_j)$	$P(ag_j)$
jaime	g1 g2 g3 g4	a1	r1	g1:=a3(r1). g2:=a6(r1),a7(r2). g3:=a1(r1),a2(r3). g4:=a1(r1),a4(r5),a5(r4).
rosaria	g1	a2	r2	g1:=a2(r2).
cristiano	g5	a3	r3	g5:=a3(r3).
vittorio	g4	a4	r4	g4:=a4(r5),a5(r4).
maria	g4	a5	r5	g4:=a1(r1),a5(r7).
amedeo	g3	a6	r6	g3:=a6(r6).
paola	g2 g3	a7	r7	g2:=a5(r1),a7(r2). g3:=a1(r2),a6(r3).

1 **Independence:** using his own plans, ag_i infers that he does not a-depend on ag_j for g_k:

$$IND(ag_i, ag_j, g_k) \stackrel{def}{\equiv} \neg a_{dep}(ag_i, ag_j, g_k, P_{ik})$$

In our example, agent *jaime* is independent on agent *rosaria* regarding g_1:

```
Regarding agent:(rosaria)

Agents (jaime) and (rosaria) are independent
regarding goal g1
Agent (rosaria) is s-autonomous for goal g1
```

2 **Locally Believed Mutual Dependence:** using his own plans, ag_i infers that there is a mutual dependence between himself and ag_j for g_k, but he can not infer the same using ag_j's plans:

$$LBMD(ag_i, ag_j, g_k) \stackrel{def}{\equiv} MD(ag_i, ag_j, g_k, P_{ik}) \wedge \neg MD(ag_i, ag_j, g_k, P_{jk})$$

In our example, agent *jaime* locally believes that there is a mutual dependence between himself and agent *vittorio* for g_4 (because of a_1 and a_4):

```
Regarding agent:(vittorio)
```

```
jaime
--------- g4
        |---------- g4:=a1(r1),a4(r5),a5(r4).
vittorio          |---------- a4
                           |---------- vittorio
--------- g4
        |---------- g4:=a4(r5),a5(r4).|----------
                  |---------- a5
                           |---------- maria
                                     |----------
```

Mutually Believed Mutual Dependence: using his own plans, ag_i infers that there is a mutual dependence between himself and ag_j for g_k. Moreover, using a_j's plans, he infers the same mutual dependence:

$$MBMD(ag_i, ag_j, g_k) \stackrel{def}{\equiv}$$
$$MD(ag_i, ag_j, g_k, P_{ik}) \wedge MD(ag_i, ag_j, g_k, P_{jk})$$

If we adopt the hypothesis of external description compatbility described in section 2, the two agents ag_i and ag_j will nfer the same result, since it is quite easy to prove the following identity: $MBMD(ag_i, ag_j, g_k) \Longleftrightarrow MBMD(ag_j, ag_i, g_k)$.

In our example, there is a mutually believed mutual deendence between agents *jaime* and *maria* for g_4 (because of a_1 and a_5):

```
Regarding agent:(maria)

 jaime
---------- g4
        |---------- g4:=a1(r1),a4(r5),a5(r4).
 maria            |---------- a5
---------- g4              |---------- maria
        |---------- g4:=a1(r1),a5(r7).|----------
                  |---------- a1
                           |---------- jaime
                                     |----------
```

One must note that despite this mutually believed dependence, we are not supposing that *the plans involved must be the same*, as in this example. An agent may start, for instance, an interaction protocol in order to persuade to other to adopt his own plan. This feature has the advantage of not restricting unnecessarily the proposed framework.

4 **Locally Believed Reciprocal Dependence:** using his own plans, ag_i infers that there is a reciprocal dependence between himself and ag_j for g_k and g_l, but he can not infer the same using ag_j's plans:

$$LBRD(ag_i, ag_j, g_k, g_l) \stackrel{def}{\equiv}$$
$$RD(ag_i, ag_j, g_k, g_l, P_{ik}, P_{il}) \wedge \neg RD(ag_i, ag_j, g_k, g_l, P_{jk}, P_{jl})$$

In our example, agent *jaime* locally believes that there is a reciprocal dependence between himself and agent *amedeo* regarding g_2 (because of a_6) and g_3 (because of a_1):

```
Regarding agent:(amedeo)

Agent (amedeo) depends on agent (jaime) for goal: g3
 jaime
---------- g2
        |---------- g2:=a6(r1),a7(r2).
                  |---------- a6
                           |---------- amedeo
                                     |----------
Agent (amedeo) is s-autonomous for goal g3
```

5 **Mutually Believed Reciprocal Dependence:** using his own plans, ag_i infers that there is a reciprocal dependence between himself and ag_j for g_k and g_l. Moreover, using a_j's plans, he infers the same reciprocal dependence:

$$MBRD(ag_i, ag_j, g_k, g_l) \stackrel{def}{\equiv}$$
$$RD(ag_i, ag_j, g_k, g_l, P_{ik}, P_{il}) \wedge RD(ag_i, ag_j, g_k, g_l, P_{jk}, P_{jl})$$

Once more, the two agents ag_i and ag_j will infer the same result if we adopt the hypothesis of external description compatibility, since it is also quite easy to prove the following identity: $MBRD(ag_i, ag_j, g_k, g_l) \Longleftrightarrow MBRD(ag_j, ag_i, g_l, g_k)$.

In our example, there is a mutually believed reciprocal dependence between agents *jaime* and *paola* regarding g_2 (because of a_7) and g_3 (because of a_1):

```
Regarding agent:(paola)

Agent (paola) depends on agent (jaime) for goal: g3
 jaime
---------- g2
        |---------- g2:=a6(r1),a7(r2).
 paola            |---------- a7
---------- g3              |---------- paola
        |---------- g3:=a1(r2),a6(r3).|----------
                  |---------- a1
                           |---------- jaime
                                     |----------
```

Once again, as in the MBMD situation, we are not supposing that *the plans involved must be the same*, as in the proposed example.

6 **Unilateral Dependence:** using his own plans, ag_i infers that he a-depends on ag_j for g_k, but this latter does not a-depend on him for any of his goals:

$$UD(ag_i, ag_j, g_k) \stackrel{def}{\equiv} a_{dep}(ag_i, ag_j, g_k, P_{ik}) \wedge$$
$$\neg \exists g_l \in G(ag_j)\ a_{dep}(ag_j, ag_i, g_l, P_{il})$$

In our example, agent *jaime* is unilaterally dependent on agent *cristiano* regarding g_1 (because of a_3):

```
Regarding agent:(cristiano)

 jaime
---------- g1
        |---------- g1:=a3(r1).
                  |---------- a3
                           |---------- cristiano
                                     |-----------
Agent (cristiano) have not goal g1 in his current
list of goals
```

5 The DEPNET Simulator

We have implemented a simulator, called DEPNET, which calculates both the dependence relations and situations between agents, and constructs the dependence networks of a given agent. According to the hypothesis of external description compatibility presented in section 2, there is only one external description which is shared by all agents. This simulator is composed of the following facilities:

- *agent edition module:* the user can dynamically create new agents and edit their goals, actions, resources, and plans, or modify the entry of an existing agent in the external description;
- *dependence network constructor:* this module constructs the various dependence networks of a given agent, either related to a specific goal or to all of his goals. It can construct both the a-dependence and the r-dependence networks for both cases;
- *dependence situation constructor:* this module calculates the dependence situations regarding a given agent and one of his goals. The user must specify the type of dependence situation he is interested in analysing.

The DEPNET simulator runs in UNIX workstations, and it was developed using the C++ programming language. The total number of lines of code is approximately 5800.

6 Conclusions and Further Work

In this paper, we have stressed the importance of a **social reasoning mechanism** in an agent's internal model, considering a MAS context. This feature is essential if an agent has to adapt himself to an evolving environment, to take into account information about new members of the agency, in an open system context [7].

The most obvious consequence of using such a mechanism is *decreasing the overall agency communication flow*. Even if every agent sends a broadcasting message to introduce himself, this is done only once, when he enters the agency. Since the others can take into account this information, there is no more need to send a broadcasting message every time an agent needs a given action or resource, as in [11], as he can know a priori the agents he should address.

On the other hand, the proposed framework allows an agent who wants to achieve a given goal to reason about the others in *two* different levels: *whom do I depend on* (in this case he may use only the dependence relations) and *who depends on me* (in this case he can use the dependence situations). Depending on the nature of the agents, these both levels may be used or not. Normally, the second level is hardly used in a benevolent world, where every agent wants to cooperate with the others. On the other hand, self-interested agents that want to achieve their own goals could benefit from the dependence situations in order to get their needed actions/resources more quickly, as described in [9]. Anyway, just the fact of using the first level has already a great impact in reducing the overall agency communication flow.

We intend to improve the *DEPNET simulator* (graphical interfaces, log files for storing sessions, unification mechanism for the plans) and extend the *dependence situations* (taking into account r-dependences as well). The computational models will be *integrated in LIFIA's MAS platform and agent models* (in particular by implementing both an introduction protocol [1] and an internal model of agent that uses the dependence networks in his social reasoning mechanism) and the DEPNET simulator will be used in more sophisticated *micro-social simulations* (to validate some theoretical results, specially concerning three or more party dependences [5]). Finally, we want to propose a model for the *quantification of dependence* (using goals' importance, number of actions and resources involved in a plan and so on). The main idea is to use this quantification to guide a decision mechanism in solving conflicts.

ACKNOWLEDGEMENTS

Most of the work described in this paper was developed between July/October 1993, when Jaime Sichman was working as a visiting researcher at IP/CNR. The authors would like to thank Olivier Boissier (LIFIA/IMAG) and Prof. Helder Coelho (INESC/Lisbon) for their useful comments during the revision of this paper. Yves Demazeau is a research fellow at CNRS, France. Jaime Simão Sichman is on leave from PCS-EPUSP, Brazil, and supported by FAPESP, grant number 91/1943-5.

REFERENCES

[1] S. Berthet, Y. Demazeau, and O. Boissier, 'Knowing each other better', *Proc. 11th. Int. Workshop on Distributed AI*, Glenn Arbor, USA, 1992, p. 23-41.

[2] E. Cardozo, J. S. Sichman and Y. Demazeau, 'Using the active object model to implement multi-agent systems', *Proc. 5th. IEEE Int. Conf. on Tools with AI (TAI'93)*, Boston, USA, 1993, p. 70-77.

[3] C. Castelfranchi, M. Miceli and A. Cesta, 'Dependence relations among autonomous agents', in E. Werner and Y. Demazeau (eds.) *Decentralized A. I. 3*, Elsevier Science Publishers B. V. 1992, p. 215-227.

[4] C. Castelfranchi, 'Social Power: A missing point in Multi-Agent, DAI and HCI', in Y. Demazeau and J. P. Muller (eds.) *Decentralized A. I.*, Elsevier Science Publishers B. V., 1990. p. 49-62.

[5] R. Conte, 'Three-party dependence and rational communication', *Atti del 2do. Incontro AI*IA su IA Distribuita*, Rome, Italy, 1992, p. 105-114.

[6] Y. Demazeau and J. P. Muller, 'From reactive to intentional agents', in Y. Demazeau and J. P. Muller (eds.) *Decentralized A. I. 2*, Elsevier Science Publishers B. V., 1991, p. 3-10.

[7] C. Hewitt, 'Offices are open systems', in A. Bond and L. Gasser (eds.) *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann Publishers Inc., 1988, p. 321-329.

[8] D. Knoke *Political Networks: The Structural Perspective*, Cambridge University Press, 1990.

[9] M. Miceli, A. Cesta and R. Conte, ' Others as resources: cognitive ingredients for agent architecture', in K. Ryan and R. F. Sutcliffe (eds.) *AI and Cognitive Science'92*, Springer-Verlag, 1992. p.84-98.

[10] J. S. Sichman, Y. Demazeau and O. Boissier, 'When can knowledge-based systems be called agents?', *Anais do 9o. Simpósio Brasileiro de IA (SBIA'92)*, Rio de Janeiro, Brazil, 1992, p. 172-185.

[11] R. G. Smith, 'The contract net protocol: high-level communication and control in a distributed problem solver', *IEEE Transactions on Computers*, v. 29, n. 12, p. 1104-1113, Dec. 1980.

[12] E. S. K. Yu and J. Mylopoulos, 'An actor dependency model of organizational work with application to business process reengineering', *Proc. of the Conference on Organizational Computing Systems (COOCS'93)*, Milpitas, USA, 1993.