Towards a Cooperation Knowledge Level For Collaborative Problem Solving

N.R.Jennings[1]

Dept. of Electronic Engineering, Queen Mary & Westfield College

London E1 4NS, UK

Abstract. The *cooperation knowledge level* is a new computer level specifically for multi-agent problem solvers which describes rich and explicit models of common social phenomena. A cooperation level description (called *joint responsibility*) is developed to describe how participants should behave during interactions in which groups of agents collaborate to solve a common problem. The utility of this model is highlighted in the real-world environment of electricity transport management in which agents have to make decisions using partial, imprecise views of the system and cope with the inherent dynamics of the environment. In such situations the tracking of social action becomes a primary consideration; joint responsibility provides evaluation criteria and the causal link to behaviour upon which such assessment can be based. Keywords: Multi-Agent Systems, Distributed AI, Joint Intentions, Knowledge Level

1. Introduction

Sophisticated problem solving is based upon knowledge. In advanced systems (typified by expert systems), this knowledge can be divided into two distinct categories: about the domain and about problem solving per se. Early expert systems had many important drawbacks including brittleness, weak explanation and unclear boundaries during knowledge acquisition [1] - characteristics attributed to their sole use of *surface knowledge*. To overcome these problems, second-generation systems use rich and explicit (deep) models of knowledge. However such *knowledge-level* [2] approaches have yet to be transferred into Distributed AI (DAI). In multi-agent systems, a *cooperation knowledge level* would be concerned with those aspects of problem solving specifically related to interacting with others - offering rich and explicit models of various social phenomena (cooperation, conflicts, competition, etc.). In Newell's taxonomy of computer levels, the cooperation level would be above the knowledge level. Like the others, it can be reduced to the level directly below it - ultimately being expressed in terms of single agents and individual goals, actions and knowledge states. It differs from the individual knowledge level in that it describes actions of groups of agents not just individuals (ie groups are the *system*). Also a notion of "team rationality" is required for the behavioural law; individual rationality is inappropriate for joint problem solving.

Some recent trends in contemporary DAI can be viewed as moving towards cooperation level solutions; though there is still greater need for recognition of this new level. The most widespread use of deeper models of social phenomena occurs within the context of communication; in speech act theory communication primitives and their effects are explicitly represented and reasoned about by the sender in order to try and bring about specific mental states in the hearer [3]. Other illustrations occur in conflict resolution [4] in which resolution strategies are categorized and selected according to the desired objective and prevailing circumstances; in persuasion/negotiation [5] in which agents reason about how to induce greater cooperativeness in other community members and in the definition of likes, goals and values based on physical dynamics [6].

The benefits of cooperation knowledge level systems include: enhanced explanation facilities, greater generality (and hence software reusability) and easier knowledge acquisition for the multi-agent system designer. Explanation can be enhanced because group activities can be described at a meta-level rather than at a task or message level (eg A1 and A2 have a conflict about Y). The advances in explanation facilities offered by such systems are especially important in environments in which the user plays an active problem solving role. Software reusability is enhanced by separating out the domain independent principles from the domain-dependent knowledge which they make use of. The generic component embodying the cooperation level can be applied to new problems merely by providing appropriate domain knowledge. Finally, the multi-agent system developer is aided by having a focused set of questions, strategies and options with which to confront the organization which commissioned the system.

Here we concentrate on one particular form of social interaction, namely the solving of a common problem by a team of agents - eg several agents lifting a heavy object or driving in a convoy. A complete cooperation level description would need to cover the following aspects: the detection of when team problem solving is required/beneficial, what

[1] The work described in this paper has been partially supported by the ESPRIT II project P2256 (ARCHON) whose partners are: Atlas Elektronik, JRC Ispra, Framentec, Labein, IRIDIA, Iberdrola, EA Technology, Amber, Technical Univ. of Athens, Univ. of Amsterdam, Volmac, CERN & Univ. of Porto. In particular Erick Gaussens's help has been greatly appreciated.

organizational form the team will take (will there be a single controller, a committee or will all members be equal?, will decisions require unanimous or majority support?), who should be in the team (is it best to have small teams of major contributors or larger teams with less active members?), how to recruit community members to the team (will they join out of benevolence or will they need convincing?, if so how?), how to construct the team plan (single planner or multiple partial planners?), how to divide the labour within the team, how to behave once team activity has begun and how team activity should be terminated.

This type of problem solving is a sophisticated form of collaboration; interactions may be protracted, involve several exchanges of information and opinions or require agents to modify their stances to accommodate the desires of others. During such activity there is significant scope for errors, misunderstandings and changing opinions, especially if the application domain is itself complex and dynamic. To operate in such environments agents have to take decisions based on partial, imprecise views of the system which they may wish to alter at a later stage as more information becomes available. To cope with this inherent uncertainty, incompleteness and dynamicity, it is important that the collaborators have a well specified description of how to evaluate (track) their ongoing problem solving and a prescription of how to behave should it run into difficulties.

Joint responsibility provides a cooperation level model of collaborative problem solving, based on the notion of joint intentions (ie a commitment to perform collective action while in a certain shared mental state [7]). Particular emphasis is given to defining conditions under which joint activity may falter and the actions which must be taken in such circumstances in order to maintain group coherence. It is, therefore, appropriate for defining that part of the agent architecture which has to track joint actions. Previous formulations (eg [8], [9]) are of limited value in dynamic and unpredictable environments because they fail to address this problem, concentrating predominately on what it means for a joint intention to exist. Responsibility subsumes the work of Cohen and Levesque [7]; defining joint commitment for both plan and goal states.

2. The Responsibility Framework

Responsibility defines conditions which must be satisfied before joint action can start and specifies a code of conduct for agents once problem solving has commenced. It uses first order logic (^ AND, v OR, ~NOT) and the model operators BEL, GOAL and MB. BEL(x,p) and GOAL(x,p) mean agent x has p as a belief and a goal respectively, MB({x, y}, p) that x and y mutually believe p[1]. The standard temporal operators: ❑ (always) and ◊ (eventually) are also used. We use p?;a to mean "action a with p holding initially" and a;p? to mean "action a with p holding as a consequence".

2.1 Common and Joint Persistence of Goals

Before joint action can commence a group of agents must realise they have a common objective that they wish to fulfill collaboratively. Recognition may occur through necessity or through belief that a team approach is best. Once the objective has been agreed, a joint persistent goal (JPG) exists and individuals become committed to achieving it [7]. However commitments are not irrevocable; they can be dropped if one team member believes: the goal has been achieved, its motivation is no longer present or that it will never be attained. If such events occur, the agent who is no longer committed cannot simply disregard the remaining group members; rather it must endeavour to inform them of its lack of commitment. The rationale for this being that if one participant is no longer committed, then there must be a good cause for this and hence the others ought to be made aware so they do not waste effort unnecessarily.

2.2 Solution Commitment

JPGs are not sufficient for obtaining joint action. They only specify that agents have a common desire to reach a target state, they do not specify *how* to reach this state. At the cooperation level we are concerned with underlying principles related to agents' plans, not implementation specific details. Relevant issues include: the fact that participants must agree to the *principle* of a common solution, enumerating conditions under which commitment to it can be dropped and defining how team members should behave towards each other in such circumstances.

2.2.1 Multi-Agent Planning Syntax

The adopted representation formalism defines points in the search space as partially elaborated plans, traversed using plan transformations. Plans are represented as an action ordering in which the actions, described by operators, are strung together with temporal ordering relations [10]. There are two types of action: those which can be undertaken by individuals (*primitive actions*) and those in which groups of agents work together (*social actions*)[2]. Throughout this section, let the set of agents in the community be represented by A, the set of primitive actions which can be performed by some agent in A by P and the set of social actions which community A can perform by S. Note $P \subseteq S$ since a primitive act can trivially be performed by 2 or more agents.

Group problem solving requires some actions to be synchronized; there will be *relationships* between them. Relationships can involve arbitrary numbers of actions and may be composed entirely of primitive actions, or of social actions or a mixture of the two. So if $s_1, s_2 \in S$; $p_1, p_2 \in P$ and $\Re_{a,b}(a,b)$ the relationship between actions a and b then, the

[1.] Mutual belief is taken to be the infinite conjunction of beliefs about the other agents' beliefs, about the other agents' beliefs (and so on to an infinite depth) about a proposition.

[2.] Social actions ultimately give rise to primitive actions because it is the individuals who have the ability to act.

following relationship may exist: $\Re_{s1,s2}(s_1, s_2)$, $\Re_{p1,p2}(p_1, p_2)$, $\Re_{s1,p1}(s_1, p_1)$ and $\Re_{p1,s1,s2}(p_1, s_1, s_2)$. Two actions are independent if $\sim\Re_{a,b}(a, b)$. All actions within a sequence are also subject to at least one relationship - $\Re_{a,a}(a, a)$. Relationships between actions are as important as the actions themselves - eg when moving an object in which all parties are required to lift at the same time, failing to satisfy the relationship "SIMULT" means the lift will not occur.

Actions can be combined into finite sequences to specify more complex interactions. Sequences are composed of at least one action and may contain mixtures of primitive/social actions and related/independent actions. A sequence Σ containing 4 actions (3 social $[s_1, s_2, s_3 \in S]$, 1 primitive $[p_1 \in P]$); two of which are related (s_2 and s_3) and two of which are not (p_1, s_1) is denoted by: $\Sigma = \{p_1, s_1, \Re_{s2,s3}(s_2, s_3)\}$. Primitive actions are assumed to be solved by action sequences of length one (i.e. p_1 is solved by $\Sigma = \{p_1\}$). In goal directed systems, actions are carried out in order to attain particular objectives; so Σ_σ means that action sequence Σ is executed in order to fulfill objective σ.

It is useful to distinguish the actions to be performed from the agents who will execute them. This permits the action planning mechanisms to be independent of task and resource allocation considerations. Once the action sequence has been defined, the agents who will actually perform it need to be decided upon; actions and action sequences must be *instantiated*:

Primitive Action Instantiation: $<\alpha, a>$: agent $\alpha \in A$ is involved in primitive action $a \in P$

Social Action Instantiation: $<\{\alpha_1,..,\alpha_n\}, \sigma>$: agents $\alpha_1,..,\alpha_n \subseteq A$ $(n > 1)$ are involved in social action $\sigma \in S$

Action Sequence Instantiation: A sequence of primitive and social action *instantiations*. It specifies the actions and the agents who will perform them. If Σ_σ is an action sequence, its instantiation is denoted by Σ'_σ

Other predicates associated with actions ($a \in P$, $\sigma \in S$) include; EXECUTE(α, a) and EXECUTED(α, a) meaning that α will execute action a next and has just executed a respectively. MOTIVE($<\{\alpha_1,..,\alpha_n\}, \sigma>$) gives the reason why $\alpha_1,..,\alpha_n$ wish to achieve σ. This will typically represent a goal-subgoal hierarchy with the root node giving the reasoning for carrying out the joint action. RELATION-OK indicates that the relationship between two actions $\sigma_i, \sigma_j \in \Sigma'_\sigma$ is satisfied. If the two actions are unrelated then this returns true:

RELATION-OK($<\{\alpha_w... \alpha_x\}, \sigma_i>, <\{\alpha_y... \alpha_z\}, \sigma_j>, \Sigma'_\sigma$) ≡
$?\Re_{\sigma i,\sigma j}$ v $(\sim\exists\, \Re_{\sigma i,\sigma j} \in \Sigma'_\sigma)$

2.2.2 Performing Actions

For an agent ($\alpha \in A$) to execute primitive action ($a \in P$) within the context of action sequence Σ'_σ; all relationships involving a in Σ'_σ must be satisfied:

PERFORM($<\alpha, a>, \Sigma'_\sigma$) ≡
($\forall <\{\alpha_w...\alpha_x\}, \sigma_i> \in \Sigma'_\sigma$)
RELATION-OK($<\alpha, a>, <\{\alpha_w...\alpha_x\}, \sigma_i>, \Sigma'_\sigma$)?;
EXECUTE(α, a)

Before a group of agents ($\{\alpha_1...\alpha_n\} \subseteq A$, $n \geq 2$) can execute a social action ($\sigma_i \in S$) within the context of action sequence Σ'_σ; any relationships involving σ_i must be satisfied:

PERFORM($<\{\alpha_1,..,\alpha_n\}, \sigma_i >, \Sigma'_\sigma$) ≡
($\forall <\{\alpha_w... \alpha_x\}, \sigma_j> \in \Sigma'_\sigma$)
RELATION-OK($<\{\alpha_1..\alpha_n\}, \sigma_i >, <\{\alpha_w...\alpha_x\}, \sigma_j>, \Sigma'_\sigma$)?;
($\exists \Sigma'_{\sigma i}$ PERFORM($<\{\alpha_1,..,\alpha_n\}, \sigma_i >, \Sigma'_{\sigma i}$))

where $\Sigma'_{\sigma i}$ is a solution developed by $\{\alpha_1,..,\alpha_n\}$ for solving σ_i. PERFORMED indicates whether a joint action has been carried out and uses EXECUTED instead of EXECUTE.

2.2.3 Defining Solution Commitment

All participants must firstly acknowledge the *principle* that a common solution is needed to tackle the joint act:

NEED-COMMON-SOLUTION($<\{\alpha_1,..,\alpha_n\}, \sigma>$) ≡
($\Diamond\, \exists \Sigma'_\sigma$ PERFORM($<\{\alpha_1,..,\alpha_n\}, \sigma>, \Sigma'_\sigma$))

Commitment to the common solution is also not irrevocable, especially when agents are situated in dynamic environments. To aid the execution tracking process, circumstances in which it is rational to drop commitment to the agreed solution need to be enumerated. In all subsequent formulations it is assumed that α is a member of the group $\{\alpha_1..\alpha_n\}$.

• the motivation for carrying out one of the actions is not present (eg the objective already holds or the actions have already been performed).

LACKING-MOTIVE($<\{\alpha_1...\alpha_n\}, \sigma>, \Sigma'_\sigma$) ≡
$\{\alpha_w,..,\alpha_x\} \subseteq \{\alpha_1,..,\alpha_n\}$
($\exists <\{\alpha_w..\alpha_x\}, \sigma_i> \in \Sigma'_\sigma$) ~MOTIVE($<\{\alpha_w..\alpha_x\}, \sigma_i>$)?

• the agreed sequence does not achieve the desired outcome

INVALID ($<\{\alpha_1..\alpha_n\}, \sigma>, \Sigma'_\sigma$) ≡
PERFORMED($<\{\alpha_1,..,\alpha_n\}, \sigma >, \Sigma'_\sigma$); ~$\sigma$?

• one of the specified actions cannot be carried out

UNATTAINABLE ($<\{\alpha_1..\alpha_n\}, \sigma>, \Sigma'_\sigma$) ≡
$\{\alpha_w,..,\alpha_x\} \subseteq \{\alpha_1,..,\alpha_n\}$
($\exists <\{\alpha_w,..,\alpha_x\}, \sigma_i> \in \Sigma'_\sigma$)
❑~PERFORM($<\{\alpha_w,..,\alpha_x\}, \sigma_i> \Sigma'_\sigma$)

• one of the agreed actions was not carried out

VIOLATED ($<\{\alpha_1..\alpha_n\}, \sigma>, \Sigma'_\sigma$) ≡
~PERFORMED($<\{\alpha_1,..,\alpha_n\}, \sigma>, \Sigma'_\sigma$)

These represent situations in which an individual team member can detect, for itself, that the common solution is no longer sustainable. In such circumstances it needs to

reassess its commitment to the agreed solution:

LOCAL-PROBLEM(α, <{$\alpha_1..\alpha_n$}, σ>, Σ'_σ) ≡
  BEL(α, LACKING-MOTIVE(<{$\alpha_1..\alpha_n$}, σ>, Σ'_σ) v
        INVALID(<{$\alpha_1..\alpha_n$}, σ>, Σ'_σ) v
        UNATTAINABLE(<{$\alpha_1..\alpha_n$}, σ>, Σ'_σ)) v
        VIOLATED(<{$\alpha_1..\alpha_n$}, σ>, Σ'_σ))

Because of the very nature of group problem solving, if one member stops contributing the whole initiative may be jeopardised. Therefore if an agent realises that one of its fellow team members has dropped commitment to the solution, it needs to reassess its position to take this information into account. In contrast with the previous reasons, the actual problem has not been detected locally by the agent:

NON-LOCAL-PROBLEM(α, <{$\alpha_1..\alpha_n$}, σ>,Σ'_σ)≡ $\alpha_i \neq \alpha$
  BEL(α, ($\exists\alpha_i \in \{\alpha_1,..,\alpha_n\}$
        LOCAL-PROBLEM(α_i, <{$\alpha_1..\alpha_n$}, σ>, Σ'_σ)))

It is now possible to state the situations under which agent α can drop commitment to an agreed common solution Σ'_σ for group action <{$\alpha_1..\alpha_n$}, σ>:

DROP-SOL-COMMIT(α, <{$\alpha_1..\alpha_n$}, σ>, Σ'_σ) ≡
  LOCAL-PROBLEM(α, <{$\alpha_1..\alpha_n$}, σ>, Σ'_σ) v
  NON-LOCAL-PROBLEM(α, <{$\alpha_1..\alpha_n$}, σ>, Σ'_σ)

It is not sufficient for an agent to simply disregard a joint action once it is no longer committed to the agreed solution. The reason for this being that just because one team member (α) has detected a problem it cannot be assumed that all its accomplices have been able to so. Therefore to ensure such information is disseminated as widely as possible within the group, α must endeavour to inform all other team members of the fact that it is no longer committed and also the reason why. This enables them to reassess the actions involving α and the agreed solution itself - meaning that if the common solution needs to be abandoned or refined, then the amount of wasted resource is minimised because futile activities are stopped at the earliest opportunity. *Individual solution commitment* (ISC) represents a high level description of how each team member should behave in its own problem solving and towards others with regard to the agreed solution:

ISC(α, <{$\alpha_1..\alpha_n$}, σ>, Σ'_σ) ≡
  WHILE ~DROP-SOL-COMMIT(α,<{$\alpha_1..\alpha_n$},σ>,Σ'_σ) DO[1]
    ($\forall$ <{α, $\alpha_w..\alpha_x$}, σ_i> $\in \Sigma'_\sigma$) {α,$\alpha_w..\alpha_x$}$\subseteq${$\alpha_1..\alpha_n$}
        BEL(α, $\Diamond$PERFORM(<{α, $\alpha_w..\alpha_x$}, σ_i>, $\Sigma'_{\sigma i}$)) ^
        $\Diamond$PERFORM(<{α, $\alpha_w..\alpha_x$}, σ_i>, $\Sigma'_{\sigma i}$))
  WHEN GOAL(α, MB({$\alpha_1..\alpha_n$},
        DROP-SOL-COMMIT(α, <{$\alpha_1..\alpha_n$}, σ>, Σ'_σ)))

Therefore for each action that α is involved in, it should believe that it is going to perform that action and also that it will actually perform the action at the appropriate time. This mental state continues until α has good cause not to follow the agreed solution; whereupon it aims to disseminate its lack of commitment to all the others. Combining the results of this section, there are two facets concerned with performing actions in a social group: agreeing to the principle of a common solution and defining how individuals should behave once such a solution has been chosen:

SOL-COMMITMENT(<{α_1,..,α_n}, σ>) ≡
  MB({α_1,..,α_n},
      NEED-COMMON-SOLUTION(<{α_1,.., α_n}, σ>)) ^
  MB({α_1,..,α_n}, ($\forall$ $\alpha_i \in$ {$\alpha_1..\alpha_n$}
        ISC(α_i, <{$\alpha_1..\alpha_n$}, σ>, Σ'_σ)))

2.3 Full Joint Responsibility

We can now define the mental state of joint responsibility which a group of agents {α_1,..,α_n} must adopt if they are to jointly solve common problem σ**:**

JOINT-RESPONSIBILITY (<{α_1,..,α_n}, σ>) ≡
  MB ({α_1,..,α_n}, JPG(<{α_1,..,α_n}, σ>)) ^
  MB ({α_1,..,α_n}, SOL-COMMITMENT (<{α_1.. α_n}, σ>))

3. Responsibility in Transport Management

Electricity transportation is concerned with the process of taking electrical energy from where it is produced to where it is consumed. It requires sophisticated monitoring and any problems need to be identified at the earliest opportunity [11]. The CSI (Control System Interface) receives messages from the network and analyses them to determine whether they represent a fault. The AAA (Alarm Analysis Agent) pinpoints elements at fault and the BAI (Blackout Area Identifier) indicates groups of elements out of service (BOA). In the cooperative scenario depicted by fig 1, the CSI receives an indication that a fault has occurred and informs the other two, also providing them with information for updating their network topology models on which their diagnosis is based. The AAA starts to identify the specific network elements at fault - initially producing a quick, approximate answer which it subsequently refines using a more accurate procedure. In parallel, the BAI starts determining the BOA, which when calculated is passed onto the AAA. In order to be consistent, the elements identified by the AAA should also be in the BOA produced by the BAI - a fact taken into account by the AAA during its detailed diagnosis. While the AAA and BAI are working on diagnosis, the CSI continues to monitor the network in order to detect significant changes in status or indicate whether the fault was only transient. Once the 3 agents have been informed and agreed to participate, a joint goal exists: σ = <{AAA, BAI, CSI},DIAGNOSE-FAULT>. Each has a role to play and by combining their expertise, problem solving is enhanced. Robustness is attained by sharing information which is available within the system, but not readily avail-

[1.] WHILE p DO q WHEN r: while p is true, q will remain true. When p becomes false, q will be false and r will become true

able to all the agents. The role of joint responsibility is to provide the basis for determining which information should be shared and how agents should act when they receive it.

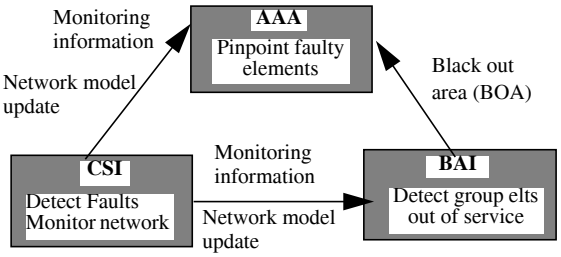

Figure 1: Cooperating Agents in transport management

When the necessary preconditions for joint action have been met, the actual solution can be developed. The responsibility framework is independent of any particular planning paradigm; so it may be derived by one or more agents. The outcome of this process will be an action sequence instantiation $\Sigma'_{diagnose}$ for σ: {***PAR*** (<{CSI}, MONITOR-NETWORK>, <{BAI},PRODUCE-BOA>, <{AAA}, INITIAL-DIAGNOSIS>), ***AFTER***(<{BAI}, PRODUCE-BOA>, <{AAA}, FINAL-DIAGNOSIS>), ***AFTER*** (<{AAA}, INITIAL-DIAGNOSIS>, <{AAA}, FINAL-DIAGNOSIS>)}

Having established the common solution, responsibility requires each agent to carry out its agreed part whilst commitment is rational. If everything goes smoothly, the objective will be satisfied and the joint goal will be terminated according to the rules specified for joint persistent goals. However because of the environmental dynamics and inherent uncertainty, several events may disrupt this activity. Related to the goal of diagnosing faults, the CSI may come to realise that the group of alarms only represented a transient fault (motivation for σ no longer present) or the AAA may realise that it is not being supplied with sufficient alarms with which to make a diagnosis (σ will never be attained). Problems may also arise with the agreed solution: the CSI may detect a substantial change in the network, meaning that the models being used by the AAA and BAI are so inaccurate that any ensuing diagnosis will be incorrect (plan invalid) or that it is no longer receiving information about the network and so is unable to monitor its status (plan unattainable). The BAI may be distracted by an unplanned task and be unable to produce the BOA at the agreed time (plan violation), meaning the AAA cannot compare its initial hypotheses with the black out area to ensure consistency before undertaking the detailed analysis.

This collaborative activity is fraught with opportunities for inconsistencies and when it does run into problems, it is usually detected by only one team member. Without a prescription of how to behave or criteria against which to evaluate joint activity, the team may perform in an uncoordinated manner. For example if after having detected the fault is transient, the CSI failed to inform the others they would continue to expend resources on diagnosing a nonexistent fault. In this case, responsibility ensures the CSI tries to inform the others that the motive is no longer present.

4. Conclusions

We have proposed a high-level model of collaborative problem solving as a contribution towards the development of a cooperation knowledge level. Responsibility describes the conditions which need to be satisfied before joint problem solving can commence and prescribes how individuals should behave once it has begun. Any theory of DAI ought to account for how aggregates of agents can achieve joint actions that are robust and continuable despite intermediate foul-ups and inconsistency [12]. Responsibility offers a step towards this; providing mechanisms for controlling activity in dynamic and unpredictable environments, whilst retaining a degree of generality and predictability. Empirical evidence to substantiate this claim has been obtained [13]. Compared with groups of selfish problem solvers and communities in which social interactions just emerge, agents organised using the responsibility model performed over twice as well as the other two if there was a greater than 10% chance of the problem solving running into difficulty.

References

[1] L.Steels (1990), "*Components of Expertise*" AI Mag., 28-48

[2] A.Newell, (1982), "*The Knowledge Level*" Artificial. Intelligence 18, 87-127.

[3] J.R.Searle, (1969), "*Speech Acts: An Essay in the Philosophy of Language*", Cambridge University Press.

[4] M.Klein & A.Baskin, (1990), "*A Computational Model for Conflict Resolution in Cooperative Design*", in Cooperating Knowledge Based System, 201-222, Springer Verlag.

[5] C.Castelfranchi, (1990), "*Social Power: A Point Missed in Multi-Agent, DAI and HCI*", MAAMAW, Cambridge, UK.

[6] G.Kiss & H.Reichgelt (1991) "*Towards A Semantics of Desires*" MAAMAW, Kaiserslautern, Germany.

[7] P.Cohen & H.Levesque, (1991), "*Teamwork*" SRI Technical Report 504.

[8] A.S.Rao & M.P.Georgeff, (1991), "*Social Plans: A Preliminary Report*", MAAMAW, Kaiserslautern, Germany.

[9] K.E.Lochbaum, B.Grosz & C.L.Sidner, (1990), "*Models of Plans to Support Communication*", AAAI, 485-490.

[10] J.Hendler, A.Tate & M.Drummond (1990) "*AI Planning: Systems and Techniques*", AI Mag., 61-77

[11] N.Jennings, A.Mamdani, I.Laresgoiti, J.Perez, & J.Corera, (1992), "*GRATE: A General Framework for Cooperative Problem Solving*" Journal of Intelligent Systems Engineering, 1.

[12] L.Gasser (1991), "*Social Conceptions of Knowledge and Action*", Artificial Intelligence 47, 107-138

[13] N.Jennings & A.Mamdani, (1992), "*Using Joint Responsibility to Coordinate Collaborative Problem Solving in Dynamic Environments*", AAAI, San Jose. (in press)