AAAI Workshop on Cooperation among
Heterogeneous Intelligent Agents

A Knowledge Level Approach to Collaborative Problem Solving

Nick Jennings
Dept. of Electronic Engineering
Queen Mary & Westfield College
London E1 4NS, UK
nickj@qmw.ac.uk

Abstract. This paper proposes, characterizes and outlines the benefits of a new computer level specifically for multi-agent problem solvers. This level is called the *cooperation knowledge level* and involves describing and developing richer and more explicit models of common social phenomena. We then focus on one particular form of social interaction in which groups of agents decide they wish to work together, in a collaborative manner, to tackle a common problem. A domain independent model (called joint responsibility) is developed to describe how participants should behave during such problem solving. Particular emphasis is placed on the problem of ensuring coherent behaviour in the face of unpredictable and dynamic environments. The utility of this model is highlighted in the real-world environment of electricity transportation management. In this domain, agents have to make decisions using partial, imprecise views of the system and the inherent dynamics of the problem mean that team members have to continually evaluate the ongoing problem solving process. Joint responsibility provides the evaluation criteria and the causal link to behaviour upon which individual and social situation assessment is based

1. INTRODUCTION

Sophisticated problem solving is based upon knowledge. In advanced problem solving systems (typified by expert systems), this knowledge can be divided into two distinct categories: knowledge about the problem domain and knowledge about problem solving per se. First generation expert systems had many important drawbacks including brittleness, weak explanation and unclear boundaries during knowledge acquisition - characteristics attributed to their sole use of *surface knowledge* (Steels, 1984). To overcome these problems, second-generation systems use rich and explicit models of knowledge (deep knowledge (Steels, 1984)), explicit inference structures (eg (Clancy, 1985) within the field of heuristic classification), problem solving methods (McDermott, 1988) and generic tasks and task-specific architectures (Chandrasekaran, 1983). The success of such *knowledge-level* (Newell, 1982) approaches in single agent problem-solving systems, has yet to be transferred into Distributed AI (DAI) in which multiple agents work together to solve common problems. In multi-agent systems, a *cooperation knowledge level* would be concerned with those aspects of problem solving specifically related to interacting with other agents - offering rich and explicit models of various social phenomena (cooperation, conflicts, competition, etc.) and the reasoning processes which control them. In Newell's taxonomy of computer system levels, the cooperation level would be directly above the knowledge level. The cooperation knowledge level differs from the individual knowledge level in that it has groups of agents as the *system* rather than individuals. That is, descriptions at this level must be in terms of collectives, not just the individual components. Like the others, the cooperation level can be reduced to the level directly below it - ultimately being expressed in terms of single agents and individual goals, actions and knowledge states (knowledge level components).

Some recent trends in contemporary DAI can be viewed as moving towards cooperation knowledge level solutions; though there is still greater need for the recognition and definition of this new level. The most widespread use of deep models of social phenomena occur within the context of communication; in speech act theory communication primitives and their affects are explicitly represented and are reasoned about by the sender in order to try and bring about specific mental states in the hearer (Searle, 1969). Other

illustrations of deeper reasoning models occur in conflict resolution (Lander et al., 1990; Klein & Baskin, 1990) in which resolution strategies are categorized and selected according to the desired objective and prevailing circumstances; in persuasion/negotiation (Castelfranchi, 1990; Sycara, 1989) in which agents reason about how to induce greater cooperativeness in other community members and in the definition of hedonic states, likes, goals and values based on physical dynamics (Kiss & Reichgelt, 1991).

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 (such as industrial control (Jennings, 1991a)). 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 knowledge level can then be applied to new problems merely by substituting in the 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, musicians in an orchestra, driving in a convoy and playing cricket. A complete cooperation knowledge level description would need to cover the following aspects: the detection of when team problem solving is required/beneficial, what organization form the team will take (will there be a single controller, a controlling 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 with each member doing significant amounts of processing or larger teams with less active members?), how to recruit community members to the team (will individuals join merely 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. Fortunately, not all of these issues need to be addressed every time; in many situations, these issues will be constrained by the structures (organizational, common reference model etc.) already present in the environment.

Team problem solving is a sophisticated form of collaboration (cf requesting a piece of data); interactions may be protracted, involve several exchanges of information and opinions or require agents to modify their stances on certain issues 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 subject to change. Section two describes a typical industrial control scenario in which agents have to operate in such circumstances. In this environment 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 unpredictability, incompleteness and environmental dynamicity, it is important that the collaborators have a well specified description of how to evaluate their ongoing problem solving and a prescription of how to behave should the joint activity run into difficulties.

Groups of agents are the cooperation knowledge level system, therefore it is necessary to be able to describe the activities of groups, rather than just those of individuals. One potential group-level description is that of joint intentions - a commitment to perform collective action while in a certain shared mental state (Cohen & Levesque, 1991). Here we propose the notion of *joint responsibility* which defines shared commitment for team problem solving (Jennings,

1991b). Responsibility differs from other accounts of joint intentionality in that it stresses the role of a "conduct controller" (Bratman, 1990) - specifying how agents should behave whilst engaged in collaborative problem solving. Previous work in this area (Rao & Georgeff, 1991; Lochbaum et al., 1990; Searle, 1990) has concentrated on defining what it means for a joint intention to exist; this description being in terms of nested structures of belief and mutual belief about the goals and intentions of the individual and of other agents within the community. However these descriptions are of limited value in dynamic and unpredictable environments because they fail to describe what conditions may jeopardize collaborative activity and the concomitant prescription of how the agents should behave in such circumstances. Responsibility also incorporates and extends the work of Cohen and Levesque (1991); defining joint commitment for both plan and goal states.

2. MONITORING ELECTRICITY TRANSPORT NETWORKS

To be available at customers' sites, electricity has to be transported, sometimes over many hundreds of kilometres, from the power station where it is produced. During this process, there is significant scope for problems (eg power lines may become broken, substations damaged by lightning strikes, etc.). To ensure early detection of such problems, many distribution companies have installed sophisticated monitoring and diagnosis software. An example of three such systems, acting cooperatively to produce a list of faults, is shown below:

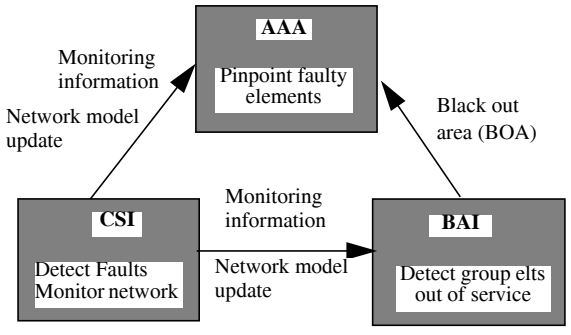

Figure 1: Cooperating Agents in transport management

The CSI (Control System Interface) is responsible for receiving messages from the network and analyzing them to determine whether they represent a fault. The AAA (Alarm Analysis Agent) has to pinpoint the elements at fault and the BAI (Blackout Area Identifier) indicates the group of elements out of service (the blackout area - BOA), both of which are based on the CSI's information. Several cooperative scenarios can be identified between this group of agents (Aarnts et al., 1991), however we will concentrate on the one depicted above.

The CSI continuously receives information about the state of the network, which it groups together and analyses. In most cases, this information will periodically be sent to the BAI and AAA so that they can update their network models. However when the information encodes a fault, the CSI immediately informs the other two. Whereupon the AAA starts its diagnostic process for identifying the specific network elements at fault - initially producing a quick, approximate answer which it subsequently refines using a more accurate procedure. At the same time, 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 the diagnosis, the CSI continues to monitor the network in order to detect significant changes in status or indicate whether the fault was only transient. Therefore once a fault has been detected, each agent has a role to play and by combining their expertise, problem solving is enhanced. The degree of system robustness can be improved by sharing information which is available within the system, but not readily available to all the agents. There are two main cases in which this can be seen: firstly, if the CSI detects that a fault is transient and the other two are working on diagnosing a nonexistent fault. Secondly if further faults occur, the network topology may be so radically altered that the existing diagnosis is predicated on invalid assumptions. Further details of the implementation of this scenario are given in

(Jennings et al., 1992).

3. THE RESPONSIBILITY FRAMEWORK

Joint responsibility defines the conditions which need to be satisfied before joint action can be initiated 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 from dynamic logic: ❑ (always) and ◊ (eventually) are also used. As in (Cohen & Levesque, 1990), 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".

3.1 Common and Joint Persistence of Goals

Before joint action can commence a group of two or more agents must realise they have a common objective (or *intention*[2]) that they wish to fulfill by collaborating with others. This recognition may occur through necessity (eg one agent cannot lift a heavy object alone) or through belief that a team approach is best (eg when searching for a lost object in a large area it is often better to do so as a team). Once a common objective has been agreed by all team members, a joint goal can be said to exist and individuals become committed to achieving it.

However individual commitment is not a sufficiently sturdy foundation upon which robust *joint* action can be based (Cohen & Levesque, 1991). To rectify these problems the notion of *joint persistent goals* (JPGs) is proposed, in which groups of agents become jointly committed to a common aim. There are two facets to JPGs: the conditions under which commitment to a joint goal can be dropped and a prescription of how an individual should behave in such circumstances. Commitment to a joint goal can only be dropped if one team member believes that: the goal has been achieved, the motivation for it is no longer present or it will never be attained. When any of these conditions are fulfilled, the team member who is no longer committed cannot simply disregard the remaining group members and start new work; rather it must endeavour to inform others of his lack of commitment. The rationale for this being that if one of the team members is no longer committed to the group objective, then there must be a good cause for this (since it was once committed) and hence other team members ought to be made aware so they do not waste effort unnecessarily.

3.2 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. There are many real world illustrations of the necessity of agreement of a solution before joint action can commence. At the cooperation knowledge level we are concerned with the underlying principles related to this common solution requirement, not implementation specific details. We are concerned with the fact that participants must agree to the *principle* of a common plan, of enumerating conditions under which commitment to the joint plan can be dropped and defining how team members should behave towards others in such circumstances.

3.2.1 Multi-Agent Planning Syntax

In common with most modern planning systems, the adopted representation formalism defines points in the search space as partially elaborated plans which are 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 (Hendler et al., 1990). There are two types of action: those which can be undertaken by individuals (*primitive actions*) and those in which groups of agents work together

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. Note MB({x}, p) ≡ BEL(x,p)

2. Intentions have been ascribed a variety of different meanings (eg (Bratman, 1990; Werner, 1989)); within this context they specify a desire or target *without* consideration of how it is to be attained.

(*social actions*)[3]. 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 the community 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*[4] 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 following relationship types 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, the tautology $\Re_{a,a}(a, a)$. Relationships between actions are as important as the actions themselves. For instance when moving an object in which all parties are required to lift at the same time, failing to satisfy the relationship SIMULTANEOUSLY means the object will not be moved.

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) can be written as follows: $\Sigma = \{p_1, s_1, \Re_{s2,s3}(s_2, s_3)\}$. For our purposes, it is assumed that primitive actions are solved by action sequences of length one (i.e. action $p \in P$ is solved by the sequence $\Sigma = \{p\}$[5]). 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 often useful to distinguish the actions which need to be performed from the agents who will actually perform them. This separation allows the mechanisms for deciding which actions need to be performed (determining the recipe (Pollack, 1990)) to be independent of task and resource allocation mechanisms. Once the action sequence has been defined, the agents who will actually execute them 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 agents who will perform each individual component as well as the actions which need to be performed. If Σ_σ is an action sequence, then its instantiation is represented by Σ'_σ

Other predicates associated with actions ($a \in P$, $\sigma \in S$) include: EXECUTE(α, a) and EXECUTED(α, a) meaning that agent α will execute action a next and has just executed a respectively. MOTIVE($<\{\alpha_1,..,\alpha_n\}, \sigma>$) gives the reason why agents $\alpha_1,..,\alpha_n$ 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. So in the transportation domain, the motivation for collaborative activity is that a fault has been detected and the motivation for carrying out specific actions (eg monitoring the network) is that they are a part of the joint act of diagnosing the fault. RELATION-OK indicates that the relationship between two actions $\sigma_i, \sigma_j \in \Sigma'_\sigma$ is

3. Social actions ultimately give rise to primitive actions as it is the individuals who have the ability to act.

4. Relationships may be temporal (Allen, 1984), plan transformations (Hendler et al., 1990) or express other types of constraint.

5. In reality primitive actions may be composed of several sub-actions, but because these are internal to the agent they need not be specified at this level.

satisfied. If the two actions are unrelated then this predicate returns true:

RELATION-OK(<{$\alpha_w..\alpha_x$}, σ_i>, <{$\alpha_y...\alpha_z$}, σ_j>, Σ'_σ) ≡

?$\Re_{\sigma i,\sigma j}$ v (~∃ $\Re_{\sigma i,\sigma j} \in \Sigma'_\sigma$)

3.2.2 Performing Actions

The simplest type of action whose execution can be described is a primitive act. For an agent ($\alpha \in$ A) to execute primitive action (a ∈ P) within the context of action sequence Σ'_σ; all relationships involving a in Σ'_σ must be satisfied:

PERFORM(<α, a>, Σ'_σ) ≡
  (∀ <{$\alpha_w...\alpha_x$}, σ_i> ∈ Σ'_σ)
    RELATION-OK(<α, a>, <{$\alpha_w...\alpha_x$}, σ_i>, Σ'_σ)?;
      EXECUTE(α, a)

Similarly before a group of agents ({α_1,..,α_n} ⊆ A, n ≥ 2) can execute a social action ($\sigma_i \in$S) within the context of action sequence Σ'_σ; any relationships in Σ'_σ involving σ_i must be satisfied:

PERFORM(<{α_1,..,α_n}, σ_i >, Σ'_σ) ≡
  (∀ <{α_w.. α_x}, σ_j> ∈ Σ'_σ)
    RELATION-OK(<{α_1,..α_n}, σ_i>, <{α_w..α_x}, σ_j>, Σ'_σ)?;
    (∃$\Sigma'_{\sigma i}$ PERFORM(<{α_1,..,α_n}, σ_i >, $\Sigma'_{\sigma i}$))

where $\Sigma'_{\sigma i}$ is a solution developed by {α_1,..,α_n} for solving σ_i. The predicate PERFORMED indicates whether a joint action has been carried out and uses EXECUTED instead of EXECUTE.

3.2.3 Defining Solution Commitment

Having given the syntax and semantics of the plan description language we proceed with the definition of solution commitment. The first aspect is that all team members acknowledge the *principle* that a common solution (action sequence) is needed if the objective is to be achieved:

NEED-COMMON-SOLUTION(<{α_1,..,α_n}, σ>) ≡
  (◊ ∃Σ'_σ PERFORM(<{α_1,..,α_n}, σ>, Σ'_σ)) v ❑~σ

A *prescription* of how team members should behave once such a solution has been developed is also required for a comprehensive model. To develop robust, collaborative problem solving systems for dynamic environments it is important that agents are able to continually evaluate their ongoing activities. To provide a well founded basis for this evaluation, circumstances in which commitment to an agreed solution can be dropped need to be enumerated:

• the motivation for carrying out one of the actions is not present.

LACKING-MOTIVE(<{α_1,..α_n}, σ>, Σ'_σ) ≡
  {α_w,..,α_x} ⊆ {α_1,..,α_n}
  (∃<{α_w..α_x}, σ_i> ∈ Σ'_σ) ~MOTIVE(<{α_w..α_x}, σ_i>)?

• following the agreed action sequence does not achieve the desired action.

INVALID (<{α_1,..α_n}, σ>, Σ'_σ) ≡
  PERFORMED(<{α_1,..,α_n}, σ >, Σ'_σ); ~σ?

• one of the specified actions cannot be carried out.

UNATTAINABLE (<{α_1,..α_n}, σ>, Σ'_σ) ≡
  {α_w,..,α_x} ⊆ {α_1,..,α_n}
  (∃ <{α_w,..,α_x}, σ_i> ∈ Σ'_σ)
    ❑~PERFORM(<{α_w,..,α_x}, σ_i> Σ'_σ)

• one of the agreed actions has not been carried out.

VIOLATED (<{α_1,..α_n}, σ>, Σ'_σ) ≡
  ~PERFORMED(<{α_1,..,α_n}, σ>, Σ'_σ)

The above situations are those in which an individual team member can locally detect that the agreed common solution is no longer sustainable. In such circumstances the agent needs to reassess its commitment to the agreed solution:

LOCAL-PROBLEM(α,<{α_1..α_n},σ>,Σ'_σ) ≡
  $\alpha \in$ {α_1,..,α_n}
  BEL(α, LACKING-MOTIVE(<{α_1..α_n},σ>,Σ'_σ))v
  BEL(α, INVALID(<{α_1,..α_n}, σ>, Σ'_σ)) v
  BEL(α, UNATTAINABLE(<{α_1,..α_n}, σ>, Σ'_σ)) v
  BEL(α, VIOLATED(<{α_1,..α_n}, σ>, Σ'_σ))

Because of the very nature of group problem solving, if one team member stops contributing because it has detected a problem then the whole joint action solution may be jeopardised. Therefore if an agent comes to realise that one of its fellow team members has dropped commitment to the solution, then it needs to reassess its position to take this information into

account. In contrast with the previous reasons, this time the problem has been detected or caused by the actions of another team member:

NON-LOCAL-PROBLEM(α, $<\{\alpha_1..\alpha_n\}, \sigma>, \Sigma'_\sigma$) $\equiv$
$\alpha_i \neq \alpha$, $\alpha \in \{\alpha_1,..,\alpha_n\}$
BEL(α, ($\exists\alpha_i \in \{\alpha_1,..,\alpha_n\}$
LOCAL-PROBLEM(α_i, $<\{\alpha_1..\alpha_n\}, \sigma>, \Sigma'_\sigma$)))

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 $<\{\alpha_1,..\alpha_n\}, \sigma>$:

DROP-SOLUTION-COMMIT(α, $<\{\alpha_1,..\alpha_n\}, \sigma>, \Sigma'_\sigma$)$\equiv$
$\alpha \in \{\alpha_1,..,\alpha_n\}$
LOCAL-PROBLEM(α, $<\{\alpha_1,..\alpha_n\}, \sigma>, \Sigma'_\sigma$) v
NON-LOCAL-PROBLEM(α, $<\{\alpha_1,..\alpha_n\}, \sigma>, \Sigma'_\sigma$)

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, $<\{\alpha_1,..\alpha_n\}, \sigma>, \Sigma'_\sigma$) $\equiv$ $\alpha \in \{\alpha_1,..,\alpha_n\}$
WHILE ~DROP-SOLUTION-COMMIT
(α, $<\{\alpha_1,..\alpha_n\}, \sigma>, \Sigma'_\sigma$) **DO**[6]
($\forall <\{\alpha, \alpha_w,..\alpha_x\}, \sigma_i> \in \Sigma'_\sigma$)
$\{\alpha, \alpha_w,..,\alpha_x\} \subseteq \{\alpha_1,..,\alpha_n\}$
BEL(α, $\Diamond$PERFORM($<\{\alpha, \alpha_w,..\alpha_x\}, \sigma_i>, \Sigma'_{\sigma i}$))^
$\Diamond$PERFORM($<\{\alpha, \alpha_w,..\alpha_x\}, \sigma_i>, \Sigma'_{\sigma i}$))
WHEN GOAL(α, MB($\{\alpha_1..\alpha_n\}$,
DROP-SOLUTION-COMMIT(α,$<\{\alpha_1..\alpha_n\}$,$\sigma>$,Σ'_σ)))

6. WHILE p DO q WHEN r: while p is true, q will remain true. When (if) p becomes false, q will be false and r will become true

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 given the correct circumstances. 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:

SOLUTION-COMMITMENT($<\{\alpha_1,..,\alpha_n\}, \sigma>$) $\equiv$
MB($\{\alpha_1,..,\alpha_n\}$,
NEED-COMMON-SOLUTION($<\{\alpha_1..\alpha_n\}, \sigma>$)) ^
MB($\{\alpha_1,..,\alpha_n\}$,
($\forall \alpha_i \in \{\alpha_1,..\alpha_n\}$ ISC(α_i, $<\{\alpha_1,..\alpha_n\}, \sigma>, \Sigma'_\sigma$)))

3.3 Full Joint Responsibility

We can now define the mental state of joint responsibility which a group of agents $\{\alpha_1,..,\alpha_n\}$ must adopt if they are to jointly solve common problem σ**:**

JOINT-RESPONSIBILITY ($<\{\alpha_1,..,\alpha_n\}, \sigma>$) $\equiv$
MB ($\{\alpha_1,..,\alpha_n\}$, JPG($<\{\alpha_1,..,\alpha_n\}, \sigma>$)) ^
MB ($\{\alpha_1,..,\alpha_n\}$, SOLUTION-COMMITMENT
($<\{\alpha_1,..,\alpha_n\}, \sigma>$))

Responsibility fulfils an important desiderata of joint intentions: “agents need general policies that govern the reconsideration of prior intentions and plans. This non-reconsideration should be treated as the default over-rideable by special kinds of problems” (Bratman, 1990). It defines the preconditions for joint problem solving as well as prescribing how individuals within the team should behave (both in default and exceptional circumstances) once such activity has started.

4. RESPONSIBILITY IN TRANSPORT MANAGEMENT

Once a fault has been detected and the three agents have been informed and agreed to participate in the collaborative activity, a joint goal exists:

σ = <{AAA, BAI, CSI}, DIAGNOSE-FAULT>

When the necessary preconditions 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 agent planning for all the others or from a collaborative planning exercise involving several agents. A potential action sequence $\Sigma'_{diagnose}$ instantiation for σ is:

{*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 committed to the joint action σ and to the solution $\Sigma'_{diagnose}$. 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 dynamics of the environment and the uncertainty inherent in the system, several events may disrupt this commitment. Related to the joint 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 (action unattainable). Also the BAI may be distracted by an unplanned task and be unable to produce the black out area 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.

As this scenario highlights, the collaborative activity of diagnosing a fault is fraught with opportunities for failures which may be caused by the actions of any of the team members. Also when the joint activity does run into problems, it is usually detected by only one team member, the others have to rely on being informed by this individual as they are unable to detect it themselves. Without a prescription of how to behave or criteria against which to evaluate joint problem solving activity, the team may perform in an inefficient or uncoordinated manner. For example, if after having detected the fault is transient, the CSI merely stopped its local activity without informing the others then they would continue to expend computational resource on diagnosing a fault which does not exist. Responsibility ensures that the CSI is tries to inform the BAI and the CSI that the motivation for the joint action is no longer present.

5. CONCLUSIONS

We have proposed a domain-independent, high-level model of the collaborative problem solving process as a contribution towards the development of the 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 such activity has begun. It offers an initial step towards the goal of providing a DAI theory which accounts for how aggregates of agents can achieve joint actions that are robust and continuable despite intermediate foul-ups and inconsistency (Gasser, 1991). It also provides a mechanism for controlling activity in dynamic and unpredictable environments, whilst still sustaining a degree of generality and predictability.

Empirical evaluation of the performance of agents situated in the dynamic and unpredictable environment of electricity transport management has been undertaken (Jennings & Mamdani, 1992). The results obtained highlight the importance of providing a theoretically grounded model for tracking the execution of joint actions and of prescribing how to behave when things do not go according to plan. Compared with groups of selfish problem solvers

and communities in which group problem solving emerges from agent interaction, agents organised according to the responsibility model perform over twice as well if there is a greater than ten percent chance of the problem solving running into difficulty.

Joint responsibility is also an important step forward in the field of distributed and multi-agent planning. At present these systems do not maintain or make use of a principled model of joint problem solving activity and their control knowledge is compiled-down and represented implicitly within the planning formalism. Joint responsibility offers an explicit meta-level representation which is independent of any particular planning paradigm but can easily be mapped down to this level. Hence the causal and motivational links, for many of the important actions and interactions within the planners can be explained by referring to the high level responsibility model.

Acknowledgments

The work described in this paper has been partially supported by the ESPRIT II Project ARCHON. We would like to acknowledge the help of all the ARCHON partners: Atlas Elektronik, JRC Ispra, Framentec, Labein, IRIDIA, Iberdrola, EA Technology, Amber, Technical University of Athens, University of Amsterdam, Volmac, CERN and University of Porto. In particular the help and interest of Erick Gaussens and Abe Mamdani has been greatly appreciated.

References

Aarnts,R., Corera,J., Perez,J., Gureghian,D., & Jennings,N.R., (1991), "*Examples of Cooperative Situations and their Implementation*", Journal of Software Research, 3, 4, pp 74-81.

Allen,J.F., (1984) "*Toward a General Theory of Time and Action*" Artificial Intelligence, 23, 123-154

Bratman,M.E., (1990), "*What is Intention?*" Intentions in Communication, (eds P.R.Cohen, J.Morgan & M.E.Pollack), 15-33, MIT Press

Castelfranchi,C., (1990), "*Social Power: A Point Missed in Multi-Agent, DAI and HCI*", in Decentralized AI, (Ed Y.Demazeau & J.P.Muller), 49-62, Elsevier.

Chandrasekaran,B., (1983), "*Towards A Taxonomy of Problem Solving Types*", AI Magazine 4, 9-17.

Clancy,W.J., (1985), "*Heuristic Classification*" Artificial Intelligence 27, 289-350.

Cohen,P.R., & Levesque,H.J., (1991), "*Teamwork*" SRI Technical Report 504.

Cohen,P.R., & Levesque,H.J., (1990), "*Intention is Choice with Commitment*" Artificial Intelligence, 42, 213-261.

Gasser,L., (1991), "*Social Conceptions of Knowledge and Action: DAI Foundations and Open System Semantics*", Artificial Intelligence 47, 107-138

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

Jennings,N.R., Mamdani,E.H., Laresgoiti,I., Perez,J., & Corera,J., (1992), "*GRATE: A General Framework for Cooperative Problem Solving*" IEE-BCS Journal of Intelligent Systems Engineering, Vol. 1.

Jennings,N.R. & Mamdani,E.H., (1992), "*Using Joint Responsibility to Coordinate Collaborative Problem Solving in Dynamic Environments*", AAAI, San Jose, California.

Jennings,N.R., (1991a), "Cooperation in Industrial Systems", ESPRIT Conference, Brussels, Belgium, pp 253-263.

Jennings,N.R., (1991b), "*On Being Responsible*" MAAMAW, Kaiserslautern, Germany.

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

Klein,M. & Baskin,A., (1990), "*A Computational Model for Conflict Resolution in Cooperative Design Systems*", in Cooperating Knowledge Based Systems (Ed S.M.Deen), 201-222, Springer Verlag.

Lander,S., Lesser,V.R., & Connell,M.E., (1990), "*Conflict Resolution Strategies for Cooperating Expert Agents*" in Cooperating Knowledge Based Systems (Ed S.M.Deen), 183-200, Springer Verlag.

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

McDermott,J., (1988), "*A Taxonomy of Problem Solving Methods*" Automating Knowledge Acquisition for Expert Systems, (Ed S.Marcus), 225-256, Kluwer.

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

Pollack,M.E., (1990), "*Plans as Complex Mental Attitudes*", Intentions in Communication, (eds P.R.Cohen, J.Morgan & M.E.Pollack), 77-103, MIT Press

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

Searle,J., (1990), "*Collective Intentions and Actions*", in Intentions in Communication, (eds P.R.Cohen, J.Morgan & M.E.Pollack), 401-416, MIT Press

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

Steels,L., (1990), "*Components of Expertise*" AI Magazine, Summer 1990, 28-48.

Steels,L., (1984), "*Second-Generation Expert Systems*", Future Generation Computer Systems 1, 213-237.

Sycara,K., (1989), "*Argumentation: Planning Other Agents' Plans*" IJCAI, 517-523

Tuomela,R., & Miller,K., (1988), "*We-Intentions*", Philosophical Studies 53, 367-389.

Werner, E., (1989), "*Cooperating Agents: A Unified Theory of Communication & Social Structure*", in Distributed Artificial Intelligence Vol II, (eds Gasser & Huhns), 3-36.