Proc Tenth National Conference on
Artificial Intelligence (AAAI '92),
San Jose, California, July 1992

Using Joint Responsibility to Coordinate Collaborative Problem Solving in Dynamic Environments†

N.R.Jennings and E.H.Mamdani
Dept. Electronic Engineering,
Queen Mary & Westfield College,
Mile End Road, London E1 4NS, UK
nickj@qmw.ac.uk and amamdani@qmw.ac.uk

Abstract

Joint responsibility is a new meta-level description of how cooperating agents should behave when engaged in collaborative problem solving. It is independent of any specific planning or consensus forming mechanism, but can be mapped down to such a level. An application of the framework to the real world problem of electricity transportation management is given and its implementation is discussed. A comparative analysis of responsibility and two other group organisational structures, selfish problem solvers and communities in which collaborative behaviour emerges from interactions, is undertaken. The aim being to evaluate their relative performance characteristics in dynamic and unpredictable environments in which decisions are taken using partial, imprecise views of the system.

Introduction

As computing systems are being applied to ever more demanding and complex domains, so the infeasibility of constructing a single monolithic problem solver becomes more apparent. To combat this complexity barrier, system engineers are starting to investigate the possibility of using multiple, cooperating problem solvers in which both control and data is distributed. Each *agent* has its own problem solving competence; however it needs to interact with others in order to solve problems which lie outside its domain of expertise, to avoid conflicts and to enhance its problem solving.

To date, two types of multi-agent system have been built: those which solve particular problems (eg air traffic control (Cammarata, McArthur & Steeb 1983), vehicle monitoring (Lesser & Corkill 1983) and acting as a pilot's aid (Smith & Broadwell 1988)) and those which are general (eg MACE (Gasser, Braganza & Herman 1988) and ABE (Hayes-Roth et al. 1988)). However, as yet, there have been few serious attempts at applying general-purpose systems to real size, industrial problems (Jennings & Wittig 1992). One of the major stumbling blocks to this advancement has been the lack of a clear, implementable theory describing how groups of agents should interact during collaborative problem solving (Bond & Gasser 1988; Gasser & Huhns 1989). Such a theory becomes especially important in complex domains in which events occur at unpredictable times, in which decisions are based on incomplete and imprecise information, in which agents posses multiple areas of problem solving competence and when social interactions are complex (i.e. involve several iterations over a prolonged period of time). In these harsh environments it is difficult to ensure that a group's behaviour remains coordinated, because initial assumptions and deductions may be incorrect or inappropriate; therefore a comprehensive theory must provide a grounded basis from which robust problem solving communities can be constructed.

Many authors have recognised that intentions, a commitment to present and future plans (Bratman 1990) are essential in guiding the actions of an individual (Cohen & Levesque 1990; Werner 1989). However in order to describe the actions of a group of agents working collaboratively the notion of joint intentions, a joint commitment to perform a collective action while in a certain shared mental state (Cohen & Levesque 1991) is needed to bind the actions of team members together. Most accounts concentrate exclusively on what it means for a joint intention to exist (Rao & Georgeff 1991; Searle 1990; Tuomela & Miller 1988); this description being in terms of nested structures of belief and mutual belief about the goals and intentions of other agents within the community. In contrast, the notion of joint responsibility (Jennings 1991a) outlined in this paper stresses the role of intentions as "conduct controllers" (Bratman 1990) - specifying how agents should behave whilst engaged in collaborative problem solving. This behavioural specification offers a clearer path from theory to implementation; providing functional guidelines for architecture design, criteria against which the monitoring component can evaluate ongoing problem solving and a

† The work described in this paper has been partially supported by the ESPRIT II project P2256 (ARCHON)

prescription of how to act when collaborative problem solving becomes untenable. Responsibility subsumes the work on joint persistent goals (Levesque, Cohen & Nunes 1990), defining a finer structure for joint commitment which involves plan states as well as goal states.

The responsibility framework has been implemented in GRATE* (Jennings 1992) and demonstrated on the exemplar domain of monitoring electricity transportation networks. The problems faced in this domain are typical of many industrial applications - especially the need to respond to the dynamics of the process being controlled/ monitored and taking decisions using partial, imprecise views of the system. An introduction to electricity transport management is given and a joint action involving three agents is described. The responsibility framework is outlined and its implementation in GRATE* is discussed. Finally some experimental results are given: offering an empirical evaluation of the characteristics of the proposed framework in dynamic, unpredictable environments.

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 illustration of three such systems, working together to produce a list of faults, is given below:

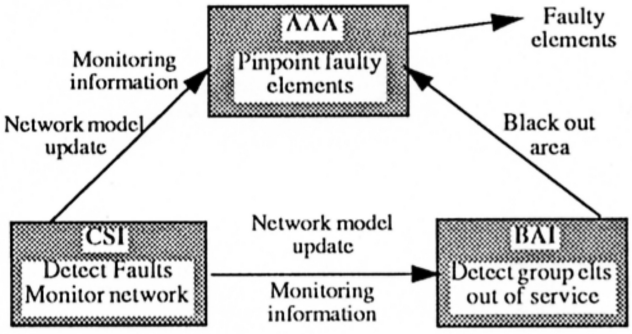

Figure 1: Cooperating Agents

The CSI is responsible for receiving messages from the network and analyzing them to determine whether they represent a fault. The AAA can pinpoint the elements at fault and the BAI can indicate the group of elements out of service, both agents using information from the CSI. Several cooperative scenarios can be identified between this group of agents (Aarnts et al. 1991), however we concentrate on the one depicted above. The CSI is continuously receiving 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. 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 group of elements out of service (the black out area), which when calculated is passed onto the AAA. In order to be consistent, the elements identified by the AAA should also be in the black out area produced by the BAI - a fact taken into account by the AAA while carrying out 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, which will invalidate any diagnoses being made, or indicate whether the fault was only transient. Once a fault has been detected, each agent has a role to play and by combining their expertise, problem solving is enhanced. Overall system robustness and performance can be improved by intelligently sharing information which is available in 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 the fault is transient, meaning the other two are attempting to diagnose a nonexistent fault. Secondly if further faults occur, the network topology may be so radically altered that the diagnosis is predicated on invalid assumptions.

Joint Responsibility

The formal account of joint responsibility uses modal, temporal logics to define preconditions which must be satisfied before joint problem solving can commence and to prescribe how individual team members should behave once it has (Jennings 1991a). Both facets are essential ingredients for a full definition of joint intentionality.

Joint Problem Solving Pre-Conditions

Once the need for joint action has been established, three conditions need to be met before it can actually begin. Firstly, a group of agents who wish to solve a common problem must be identified. In our example, willing participants are those which have or can be persuaded to have the goal of participating in the detection of faulty network elements. Secondly, participants must agree that they will work together to achieve their common objective - in particular they must acknowledge the *principle* that a

common solution is essential. Without acknowledging this, there can be no *intentional* joint action, only unintentional (accidental) interaction (Bratman 1990). The actual solution will only begin to be developed once all prerequisites have been satisfied. Finally agents must agree that they will obey a "code of conduct" to guide their actions and interactions whilst performing the joint activity. This code specified below ensures that the group operates in a coordinated and efficient manner and that it is robust in the face of changing circumstances.

Prescription of Behaviour

A comprehensive description of how individuals should behave in social interactions needs to address the duality of roles which they play - describing how to carry out local problem solving and how to act towards others

The notion of *commitment* is central to the definition of joint responsibility and means that once agents agree they will perform an action they will endeavour to carry it out. Therefore once the common solution has been agreed, all participants should ensure that they reserve sufficient resources to carry out the actions in which they are involved. However because of the unpredictability and dynamics of the environment - events may occur which affect this commitment. For example new information may become available which invalidates previous assumptions or unexpected events may require urgent attention. In such circumstances, it would be irrational for an agent to remain committed to the previously agreed actions; so conditions for reneging need to be enumerated. There are two levels at which lack of commitment can occur: to the common objective (eg there is no longer a need to diagnose faults) or to the common solution. The following reasons for dropping commitment to the common objective have been given (Levesque, Cohen & Nunes 1990):

- the objective already holds

  eg another agent has computed the faulty elements

- the motivation for the objective is no longer present

  eg CSI realises that the group of alarms do not correspond to a fault

- the objective will never be attained

  eg AAA realises that it is not being supplied with sufficient alarm messages to make a diagnosis

However conditions under which agents can drop commitment to the common solution also need to be defined (Jennings 1991a). Separate conditions relating to plan states are necessary because dropping commitment to a plan typically involves developing a new solution for the same problem rather than dropping the goal completely (i.e. it has a different functional role) and also that it provides a more detailed specification for the system implementor. Reasons include:

- following the agreed plan does not lead to the desired outcome

  eg CSI detects 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

- one (or more) of the actions cannot be executed

  eg CSI is no longer receiving information about the network and so is unable to monitor its status

- one of the agreed actions has not been performed correctly

  eg the BAI has been distracted by an unplanned task and cannot produce the black out area at the agreed time. Meaning the AAA cannot compare its initial hypotheses with the black out area to ensure consistency before undertaking the detailed analysis.

When an individual becomes uncommitted (to either the objective or the means of attaining it) it cannot simply stop its own activity and disregard other team members. Rather it must endeavour to inform all team members of this fact and also of the reason for the change. This ensures team members can monitor the progress of events which affect their joint work and, in the case of failure, the amount of wasted resource can be minimised. Combining the local and social facets, leads to the following prescription of behaviour for each team member:

while committed to joint action **do**
  perform agreed activities at correct times
  monitor situation to ensure commitment is still rational
if no longer jointly committed **then**
  suspend local actions associated with joint act
  determine if local remedial action available
  inform others of lack of commitment, reason and proposed remedy if exists

The remedy will depend on the reason for dropping commitment; varying from rescheduling actions if the plan was not executed correctly, to drawing up a new solution if the plan no longer leads to the desired objective, to abandoning the joint action if the objective is unattainable or the motivation no longer valid.

Implementing Responsibility

Joint responsibility is a meta-level prescription of agent behaviour during collaborative problem solving which is independent of the mechanisms used to obtain agreements and carry out planning. It is, therefore, equally applicable in communities where one agent carries out all the planning for other agents and in those in which the planning is carried out as a collaborative activity. It makes explicit much of the reasoning present in such planning systems, thus facilitating deeper reasoning about the process of collaboration. There are an infinite number of possible realizations of the framework (of which GRATE* is but one); each with its own *protocol* for obtaining agreements and defining the common solution. GRATE* agents have the architecture shown below - thicker arrows represent data flow and the thinner ones control.

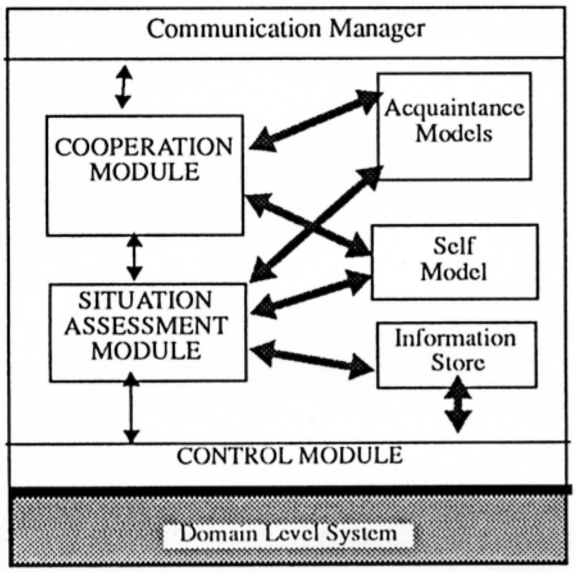

Figure 2: GRATE* Agent Architecture

An agent is divided into two parts: the domain level system (DLS) in which the agent's problem solving competence is located and the cooperation and control layer (CCL) which ensures that domain level actions are coordinated with those of others. The CCL has three main problem solving components - each implemented as an independent production system communicating with the others via messages. The *situation assessment* module provides an overview and evaluation of the local and global situation, as perceived by that agent. It monitors local actions to ensure commitments are honoured, detects commitment failures and proposes remedial solutions if they exist. The *cooperation module* has to establish social interactions once the situation assessment module detects the need (eg enact the GRATE* responsibility protocol), maintain established social interactions and provide feedback on social action initiations from other agents. The *control module* is the interface to the DLS and is responsible for managing all interactions with it. The information store provides a repository for all domain information received by the CCL (emanating from either the DLS or as a result of interaction with other agents). The acquaintance and self models are representations of other agents and of the local domain level system respectively (Jennings 1991b).

In the GRATE* protocol, each team has one leader - the agent which detects the need for joint action. This agent then contacts other community members to establish whether they are interested in participating in the group activity. Interested community members create a joint intention representation within their self model (see below) and return a message indicating their willingness to participate.

Name: (DIAGNOSE-FAULT)
Motivation: ((DIAGNOSE-NETWORK-FAULT))
Chosen Recipe: (((*START* (IDENTIFY-INITIAL-BOA)) (*START* (GENERATE-TENTATIVE-DIAGNOSIS)) (*START* (MONITOR-DISTURBANCE))) ((*START* (PERFORM-FINAL-DIAGNOSIS))))
Start Time: 8 **Maximum End Time**: 82
Duration: 74 **Priority**: 20
Status: EXECUTING-JOINT-ACTION
Outcome: (VALIDATED-FAULT-HYPOTHESES)
Participants: ((*SELF* PROPOSER EXECUTING-JOINT-ACTION) (*CSI* TEAM-MEMBER EXECUTING-JOINT-ACTION) (*BAI* TEAM-MEMBER EXECUTING-JOINT-ACTION))
Bindings: ((*BAI* (IDENTIFY-INITIAL-BOA) 19) (*SELF* (GENERATE-TENTATIVE-DIAGNOSIS) 19) (*CSI* (MONITOR-DISTURBANCE) 19) (*SELF* (PERFORM-FINAL-HYPOTHESIS-DIAGNOSIS) 35))
Contribution: ((*SELF* ((GENERATE-TENTATIVE-DIAGNOSIS) (YES SELECTED)) ((PERFORM-FINAL-DIAGNOSIS) (YES SELECTED))) (*BAI* ((IDENTIFY-INITIAL-BOA) (YES SELECTED))) (*CSI* ((MONITOR-DISTURBANCE) (YES SELECTED))))

Most of the structure is self evident, however some slots require explanation. The chosen recipe (Pollack 1990) for joint intentions is a series of actions together with some temporal ordering constraints that will produce the desired outcome and for individual intentions (see figure 3) it is the name of a local recipe. For joint intentions the status refers to the current phase of the protocol - forming-group, developing-solution or executing-joint-action; whereas for individual intentions it is simply executing or pending. The participants slot indicates the organisational structure of the group - in this example there is one organiser (AAA) and two other team members (BAI & CSI). The

bindings indicate the agents who were chosen to participate, the actions they are to perform and the time at which these actions should be carried out. The contribution slot records those agents who expressed an interest in participating in the joint action, the actions they could potentially contribute, an indication of whether they were willing to make this contribution in the context of the joint action and whether or not they were ultimately chosen to participate.

When all the potential team members have replied indicating their willingness (or not) to participate, the leader decides upon a recipe for realising the desired outcome. It then starts the detailed planning of the recipe's action timings using the following algorithm:

Forall actions in recipe **do**

select agent A to carry out action α

(criteria: minimize number group members)

calculate time (t_α) for α to be performed based on temporal orderings

send (α, t_α) proposal to A

A evaluates proposal against existing commitments (C's):

if no-conflicts (α, t_α) **then** create commitment C_α for A to (α, t_α)

if conflicts((α, t_α), C) $\wedge$ priority(α) > priority(C) **then** create commitment C_α for A to (α, t_α) and reschedule C

if conflicts((α, t_α), C) $\wedge$ priority(α) < priority(C) **then** find free time ($t_\alpha + \Delta t_\alpha$), note commitment C_α and return updated time to leader

if time proposal modified, update remaining action times by Δt_α

Making a commitment (above) involves creating an individual intention to perform an action:

Name: (ACHIEVE (IDENTIFY-INITIAL-BOA))
Motivation: (SATISFY-JOINT-ACTION (DIAGNOSE-FAULT)))
Chosen Recipe: (IDENTIFY-INITIAL-BOA)
Start Time: 19 Maximum End Time: 34
Duration: 15 Priority: 20
Status: PENDING Outcome: (Black-Out-Area)

Figure 3: Individual Intention Representation

Experimental Results

To verify the claim that the responsibility framework is

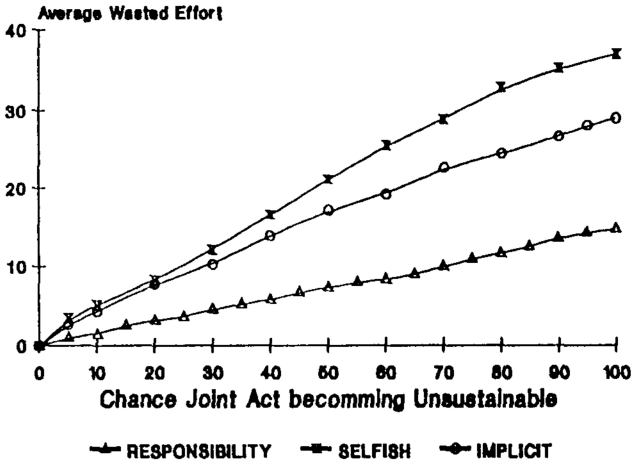

Figure 4: Varying Chance of Unsustainability

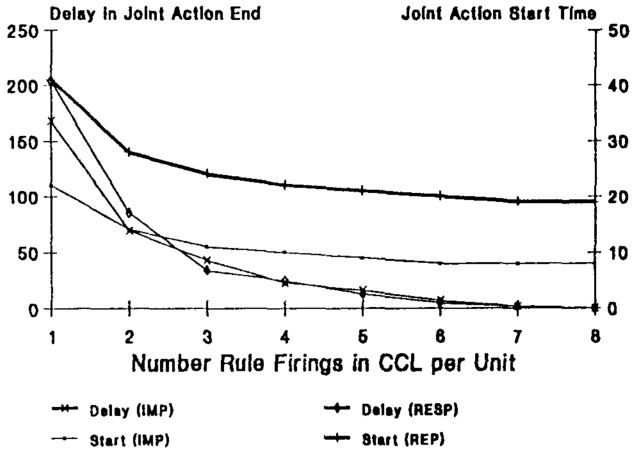

Figure 5a: Varying Amount of CCL Processing Power

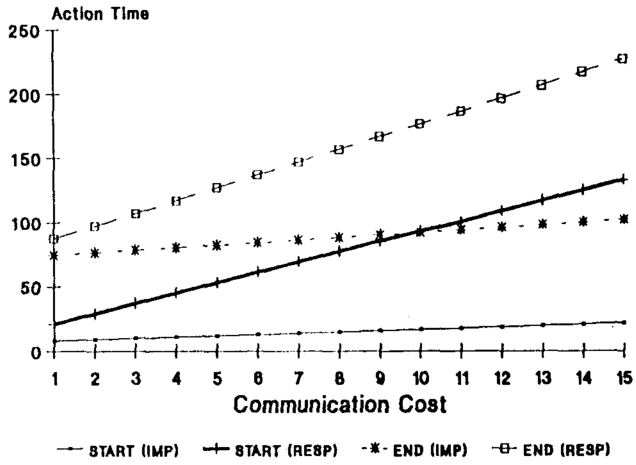

Figure5b: Varying Message Delay

capable of ensuring robust joint problem solving in dynamic and unpredictable environments, a series of comparative experiments were undertaken using the cooperative scenario outlined earlier. Three organisational structures were compared: *responsibility*, *implicit* group formation and *selfish* problem solvers. With the implicit organisational structure the agents do not explicitly form groups (there is no joint intention), rather the group structure "emerges" as a result of interaction (Steels 1991). In the selfish organisation, groups and common solutions are formed as in the GRATE* responsibility protocol. However if an agent comes to believe that the joint action has become unsustainable, it stops the associated local processing, but does *not* inform others of its lack of commitment. This is deemed selfish because the agent who detects a problem and realises that the joint action is doomed to failure does not expend additional resources informing others, since doing so brings it no direct benefit.

Figure 4 shows the performance of the three groups, in the face of varying chances of the joint action becoming unsustainable. An increased chance of unsustainability corresponds to a more dynamic environment or an environment in which decisions are based on less stable views. The distribution of the reason for unsustainability is uniform over the conditions described earlier, as is the time during the joint action at which the problem occurs. Wasted effort is defined to be the number of processing cycles from the reason for commitment failure occurring, to the time when an agent stops performing its associated actions. The average is taken over all the agents in the team (3 in this case) over 100 runs.

As this figure shows, the average wasted effort for the responsibility framework is significantly less than the other two organisational forms, confirming our hypothesis that it leads to robust behaviour in complex environments. The implicit group performs better than the selfish one because agents exchange information (based on known interests stored in the acquaintance models (Jennings 1991b)) which can lead to the recipient realising that some of its intended actions are no longer appropriate and hence should be abandoned. This informal interchange means that in the case of unsustainability, an agent which cannot detect a problem with the joint action itself may be supplied with the necessary information by an agent who can. In contrast, in the selfish group structure such informal communication paths were deliberately not used since calculating agents interested in a piece of information requires computational resource - meaning agents which were unable to detect a problem were left to complete all their actions. Therefore when claiming that self interest is the basis for cooperation (Durfee, Lesser & Corkill 1988; Axelrod 1984), it is important to note that it should *not* be used as a criteria for defining agent behaviour once the social action has started. Participation in group problem solving requires some element of compromise, meaning self interest needs to be tempered with consideration for the group as a whole.

Other studies were also carried out to examine the behaviour of the responsible (RESP) and implicit (IMP) groups when processing power is limited and communication delays varied. Figure 5a shows the affect of limiting the CCL's processing power - in terms of the total number of rules it can fire per cycle. It shows that the difference in start times between the two organisational forms remains virtually constant except when the amount of processing per time unit becomes very small. This is somewhat surprising since it was envisaged that the responsibility protocol would require greater processing power and hence be more adversely affected. The responsibility framework always takes longer to start, because it constructs groups and common solutions afresh each time a joint action is required. In practice it is unlikely that such activities would need to be undertaken every time, because common patterns would begin to emerge and hence reasoning from first principles would not be necessary. The figure also shows that except in cases where processing is severely limited, the delay (compared with infinite processing power) in the time taken to finish the joint action is approximately the same for both organisational forms. Figure 5b shows the affect of varying the time taken for a message to be delivered. By showing a sharper rise in start and finish times, it highlights the greater communication overhead present in the responsibility protocol - a result consistent with theory and practice of organisational science

Conclusions

The responsibility framework provides a new meta-level description of how agents should behave when engaged in collaborative problem solving. It has been implemented in the GRATE* system and applied to the real-world problem of electricity transportation management. An analysis of its performance characteristics has been undertaken: comparing it with emergent and selfish group organisational structures. These experiments highlight the benefits, in terms of decreased resource wastage, of using responsibility as a means for prescribing agent behaviour in dynamic and unpredictable environments. They also indicate that, in most cases, the GRATE* responsibility protocol requires no more processing power than the implicit group structure. One potential drawback, that of a large communication overhead, has been identified - therefore for less complex forms of social interaction or

time critical environments it may be appropriate to devise a more efficient protocol, whilst retaining the behavioural specification for robust and coherent behaviour.

Acknowledgments

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 discussion with, and comments from, Erick Gaussens, Thies Wittig, Juan Perez, Inaki Laresgoiti and Jose Corera have been particularly useful.

References

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

Axelrod, R. 1984. The Evolution of Cooperation. Basic Books Inc.

Bond, A. and & Gasser, L. eds. 1988. Readings in Distributed Artificial Intelligence, Morgan Kaufmann.

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

Cammarata, S., McArthur, D. and Steeb, R. 1983. Strategies of Cooperation in Distributed Problem Solving. Proc. IJCAI 767-770.

Cohen, P. R. and Levesque, H. J. 1991. Teamwork. SRI Technical Note 504.

Cohen, P. R. and Levesque, H. J. 1990. Intention is Choice with Commitment. Artificial Intelligence, 42, 213-261.

Durfee, E. H., Lesser, V. R. and Corkill, D. D. 1988. Cooperation through Communication in a Distributed Problem Solving Network, in Distributed Artificial Intelligence (Ed M.Huhns), 29-58

Gasser, L. and Huhns, M. eds 1989. Distributed Artificial Intelligence Vol II, Pitman Publishing.

Gasser, L., Braganza, C., and Herman, N. 1988. MACE: A Flexible Testbed for Distributed AI Research, in Distributed Artificial Intelligence (Ed M.Huhns), 119-153.

Hayes-Roth, F. A., Erman, L., Fouse, S., Lark, J., and Davidson, J. 1988. ABE: A Cooperative Operating System and Development Environment. AI Tools & Techniques, (Ed M.Richer), Ablex.

Jennings, N. R. and Wittig, T. 1992. ARCHON: Theory and Practice in Distributed Artificial Intelligence: Theory and Praxis (eds L.Gasser & N.Avouris), Kluwer Academic Press (forthcoming)

Jennings, N. R. 1992. GRATE*: A Cooperation Knowledge Level System For Group Problem Solving. Technical Report, Dept. Electronic Engineering, Queen Mary & Westfield College.

Jennings, N. R. 1991a. On Being Responsible. in MAAMAW, Kaiserslautern, Germany.

Jennings, N. R. 1991b. Cooperation in Industrial Systems. ESPRIT Conference, 253-263, Brussels.

Lesser, V. R. and Corkill, D. 1983. The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks. AI Magazine, 15-33.

Levesque, H. J., Cohen, P. R. and Nunes, J. H. 1990. On Acting Together", AAAI, 94-99.

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. and Georgeff, M. P. 1991. Social Plans: A Preliminary Report. MAAMAW, Kaiserslautern, Germany.

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

Smith, D. and Broadwell, M. 1988. Pilot's Associate: An Overview. SAE Aerotech Conference

Steels, L. 1991. Towards a Theory of Emergent Functionality, in From Animals to Animats, ed J.Meyer and S.Wilson, 451-461, Bradford Books.

Tuomela, R. and 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 L.Gasser & M.Huhns), 3-36.