Proc. AAAI Workshop on Cooperation among Heterogeneous Intelligent Systems,
Anaheim, USA

The Impact of Heterogeneity on Cooperating Agents

C. Roda, N.R. Jennings & E.H. Mamdani

Dept Electronic Engineering,
Queen Mary and Westfield College,
University of London,
Mile End Road,
London E1 4NS.
UK

email: c_roda/n_jennings/a_mamdani@eurokom.ie

1. Introduction

The majority of previous work in Distributed AI (DAI) has concentrated on cooperation between similar agents. Such agents typically have homogeneous structures (e.g. BEINGS [Lenat75] or DVMT [Lesser&80]), are capable of carrying out identical tasks (eg Air Traffic Control [Cammarata&83]) or are assumed to have identical domains of discourse. However such cosy assumptions and the way they simplify the problems of coordination and cooperation, are often not applicable in real-world situations - real problems involve heterogeneity!

Within the ARCHON[1] project we are developing a cooperation environment for industrial systems. There are two main characteristics of this domain which impact upon the design of our system, both of which need to be tackled if DAI is ever to leave the laboratory phase and progress into the "real" world:

- there is a substantial amount of existing software
- industrial systems are complex and require many diverse activities to be performed

Typically within the domain of industrial systems, software has been developed in an ad hoc fashion when the company has perceived that certain functions could profitably be automated given the available technology. The result is that such companies posses a large number of stand-alone systems, developed at different times, by different groups of people and utilising different problem solving techniques. These systems all operate within the same physical environment (i.e. take input from and produce output in the same process) and could benefit from interaction (sharing of information and conclusions) with other such stand-alone systems. However since they have not been constructed with an eye to future interaction, they employ techniques and representations which are best suited to the particular problem(s) they were designed for - ensuring that data/knowledge formalisms and the associated reasoning techniques are specific to that particular implementation. Faced with this situation, there are several courses of action open to the system designer who wants to exploit the benefits of coordinating activities of the pre-existing systems:

- rebuild all the existing software in a common environment, so that they all share common

1. ARCHON is an ESPRIT II project (P2256) in which the partners are Krupp Atlas Elektronik, JRC Ispra, Framentec, QMW, Iridia, Iberduero, Labein, Electricity Research & Development Centre, Amber, Technical University of Athens, University of Amsterdam, Volmac, Cern & University of Porto.

representations, reasoning mechanisms and knowledge semantics

- construct a framework into which these existing systems can be incorporated (with minimal modifications) which allows them to communicate and cooperate with each other.

  Typical examples of the modifications which have to be made include allowing the cooperation to affect the control instance of the system and altering the system's presentation of information to the user (to incorporate the fact that the system is now a team member, not merely an individual).

We chose the latter option and within this paper we relate some of our experiences with the aim of highlighting the major pitfalls and problems which need to be addressed[2]. A further design force which follows from the desire to incorporate pre-existing systems is that they will be capable of significant processing in their own right and that they will be able to solve a large percentage of their problems alone (i.e. they are (semi-) autonomous problem solvers)[3].

In addition to the problems of heterogeneity associated with pre-existing systems, we argue that even when designing a cooperating systems anew problems associated with heterogeneity cannot be avoided if the domain is sufficiently complex. Consider, for example, the domain of industrial process control; examination of a typical scenario reveals that at least the following generic functions can be identified [Gaussens90]:

Diagnosis:	delivering an understanding of a world state given some information about this world
Planning:	sequence a set of possible actions
Control:	particular case of planning where actions are executable and low-level
Supervision:	reflecting decision link between diagnosis of a dynamical system and the alternative actions needed for exceptional situation handling

The wide range of problem solving techniques necessary for each of these sub-areas means that a common technique and knowledge/data representation would be infeasible. To obtain the necessary complexity, different techniques will be needed for different tasks and even within a single generic task more than one technique may be employed (eg in diagnosis there may be a heuristic-based component and a model based component). This wide spectrum of problem solving paradigms required, clearly indicates that any solution based on a cooperating systems metaphor will encounter, and have to deal with, many forms of heterogeneity.

The remainder of this paper is divided into three sections: firstly a brief overview of the ARCHON approach to multi-agent systems, secondly a classification of the types of heterogeneity which may be present in a multi-agent system and finally a discussion of how such heterogeneity affects the knowledge and reasoning requirements of cooperating systems.

---

2. Huhns et al. describe a communication aid which can be attached to existing expert systems enabling them to interact with other expert systems. However this is a lower level description of the possible problems associated with communication (such as deadlock) and it does not address the types of issues presented in this paper [Huhns&90].

3. By limiting the scope to sophisticated problem solvers we rule out one possible dimension of heterogeneity, that of the complexity of problem solvers within the community; see [Steiner&90] for an illustration of communities composed of agents composed of vastly differing levels of complexity.

2. The ARCHON Approach

In order to cooperate in a purposeful manner an entity needs to have knowledge (be aware) of problem solving activity at both a local and a community level. To be successful, it must know how its own activity impacts on that of others, what are their intentions and current activities and what is the community actually trying to achieve. It is generally the case that the accuracy of such knowledge is directly proportional to the degree of coherent and coordinated behaviour which can be observed in such a system. In practice there is a trade-off between maintaining such information (i.e. modelling other agents) and actually performing useful problem solving, the two dominant characteristics in this trade-off being the dynamicity of each agent and the completeness of the model that agents maintain. If agents have a complete model of each acquaintance: skills, reasoning methods, payoff of actions and so on, and this model is completely accurate at all the times, entirely coordinated behaviour can be obtained at the cost of the computation necessary to evaluate this information (see [Genesereth&86, Rosenschein&89] for example). Unfortunately, in real world applications is not normally possible to have a complete and updated model of all acquaintances at all times:

- each model would almost be a complete replica of the modelled acquaintance
- the evaluation of this model would involve expensive computations
- some characteristics of acquaintances cannot be exactly modelled at all times because of their dynamicity (e.g. workload, computational resources, etc.)

Normally only a partial model can be maintained and this has to be updated whenever new/ more recent information is required. In order to update the acquaintance model communication must be established among the agents, therefore the achievement of coordinated behaviour requires the burden of communicating information in addition to that of keeping updated acquaintance models and evaluating this information. The level of heterogeneity of systems in the community has a strong impact on the ability of each agent to evaluate the activities of its acquaintances and also on the structure of the model that each agent should maintain of others. This subject is discussed more extensively in section four.

It is possible that what looks like “coordinated” behaviour (from an external perspective) may occur if such knowledge is completely lacking. An example of such a situation is when two art enthusiasts independently arrive at an art gallery with the aim of destroying a particularly shocking picture. One of them is intercepted by the guard, but because the guard is distracted, the other succeeds [Power84]. However by examining the mental state of the participants it is clear that they are not cooperating in the true sense of the word [Galliers89]. Master-slave behaviour can also be interpreted as a type of cooperation requiring minimal coordination knowledge. The master need only know that the slave has certain capabilities and it will endeavour to supply all the services requested, the slave that all the master’s requests should be obeyed (if possible). It is apparent that different organizations require different levels of knowledge about the community [Fox81], but we will not dwell on this subject here. Rather we will concentrate on the knowledge required by a community of semi-autonomous agents in order to cooperate in the presence of several types of heterogeneity.

As described previously, we have to deal with pre-existing systems which, in general, have no knowledge about cooperation. In order to give such systems a social competence (the ability to participate in group activity) we can:

- radically modify their control structure
- introduce the necessary knowledge into purpose built “coordination agents”
- introduce a structure (layer) dedicated to social activities “on top” of each problem solver

The latter solution was adopted because it maintains decentralised control over the community (greater reliability and graceful performance degradation) and introduces minimal modifications to the possibly pre-existing problem solver. In essence, therefore, the cooperation (ARCHON) layer facilitates planning and operation in a cooperative environment - this process involves making use of, and maintaining, information about other agents as well as about the self and the ability to interact in various styles. We indicate here the types of knowledge which may be maintained at the AL and in following sections highlight how different types of heterogeneities impact upon the requirements for this knowledge. A formal basis for describing the knowledge required to support cooperative activity is one of the major open issues in contemporary DAI research and therefore we propose this classification based upon experimentation only (like many others before us, e.g. [Avouris&89, Brandau&89, Gasser&88, Werner89]):

- State Knowledge

  Indicates the activities which are currently being carried out, how far they have progressed and when they are likely to be completed.

  e.g. each agent maintains a description of the tasks it is carrying out and their status (executing, finished, waiting for information, etc.)

- Capability Knowledge

  Knowledge about actions which can be performed, how they are combined to achieve particular results, the information they require, the results which can be expected.

- Intentional Knowledge

  Gives a high level description of the targets an agent is working towards or will be working towards in the near future (they represent the systems longer term objectives[4]). Complex agents are likely to have several intentions active at any one time and an agent’s intentions may conflict with those of others within the community.

- Evaluative Knowledge

  When faced with several alternatives for achieving the same objective, evaluative knowledge provides a means of distinguishing between them.

  e.g. tasks T1 and T2 may both produce result R, however one may produce the result

4. Intentions have been used to describe many different concepts (eg [Bratman90, Cohen&90, Werner89]); however within this context they refer to a desired state without reference for how that state can be reached. The distinction between plans and intentions is well defined in [Pollack90]: plans correspond to recipes for performing particular actions or for achieving particular goal states and are “known” by an agent. Intentions are adopted and are used to guide an agent’s problem solving activity

in a faster way than the other. All things being equal, an agent would usually use this information to select the task which can produce the result fastest.

• Domain Knowledge

Facts and relationships which hold true of the environment in which the agent is operating.

e.g. never do task T1 and T2 in parallel, task T1 is more important than T2 in most circumstances and so on.

3. Heterogeneity: A Classification

In the ARCHON environment, each agent is composed of a sub-system which acts at the domain level and performs the actual problem solving (DLS: Domain Level System) and of an ARCHON Layer (AL) which takes charge of that agent's social activities[5]. The ALs are homogeneous[6], meaning heterogeneity is confined to the DLS. This heterogeneity can be observed at various levels:

Type	Example
PURPOSE & SEMANTIC	*Designed to perform different tasks* *Assign different semantics to same syntactic item*
ARCHITECTURE	*eg Expert Systems: Frames, Blackboard, etc.* *eg Databases: Relational, Hierarchical, etc.*
SYSTEM TYPE	*Classical control, database, expert system*
PROGRAMMING LANGUAGE	*written using different programming languages*
OPERATING SYSTEM	*implemented on different operating systems*

We postulate that only the top two heterogeneity levels have a significant impact on cooperative activity. Lower level heterogeneities are normally hidden (e.g. it is possible to use a programming language in several operating system environments because the compiler hides any difference), but may occasionally be exploited (e.g. a task may be assigned to a certain agent because it is implemented on superior hardware). However these issues are only of marginal

5. Obviously these two activities are not disjoint - what an agent does locally effects its position within the community and its success in cooperative interactions effects its local activities.
6. The Archon Layer can be regarded as a shell which needs to be instantiated with domain dependent information. Therefore the cooperation layer structure is homogeneous, but the agents are heterogeneous.

interest for the discussion in this paper and will not be treated further.

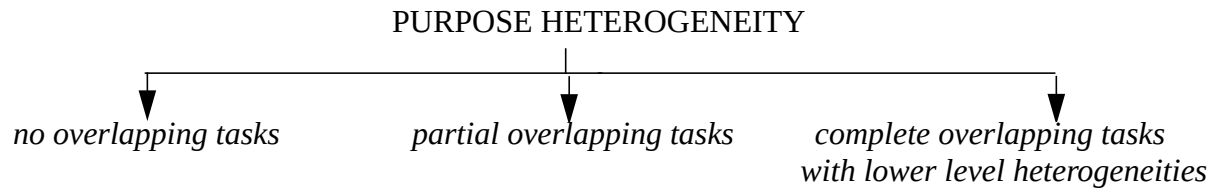
4. Heterogeneity and Cooperation

There are two major types of cooperative interaction which can be identified in a multi-agent system: the first concerns which agents performs which tasks (the task allocation problem) and the second concerns the sharing of information (both results and observations on the outside world) between agents. Purpose heterogeneity is primarily concerned with the former type and semantic heterogeneity with the latter. This section investigates the impact of both types of heterogeneity on the cooperation process in terms of the information and reasoning requirements of the ARCHON layer.

By stating that purpose and semantic heterogeneities are relevant at the cooperation level, we mean they must be transparent to the AL if it is to coordinate the DLS's activities. Note that semantic heterogeneity impacts upon purpose heterogeneity at least at the level of tasks and "object of tasks" representation. Consider the following example: two robots A and B are able to move objects between rooms. Robot A has the primitive (move-blk agt blk roomX roomY) whilst robot B has the two primitives (move-light-obj agt obj roomX roomY) and (move-heavy-obj agt obj roomX roomY). In order for A to believe that there is some overlapping between his capabilities and those of B, A has to realise that his move-blk is equivalent to the two tasks move-light-obj and move-heavy-obj in B. This also implies that the concept of weight is present in robot B but not in robot A; also robot A may think in terms of "blocks" and robot B in terms of "heavy objects" and "light objects". Things become even more complex if fuzzy concepts such as heavy and light are defined differently in different agents. For the time being we assume semantic heterogeneity at the level of tasks (purposes) descriptions. As far as the description of the "objects of tasks" (e.g. block, heavy object, light object) is concerned, there must be a common understanding of such concepts amongst the cooperating agents (this is exactly the problem tackled in the semantic heterogeneity section).

4.1 Purpose Heterogeneity

The notion of overlapping tasks is central in this dimension: two tasks overlap iff they produce the same effect at the level at which they are observed (a task is intuitively defined as a unit of work). This means two tasks overlap even if the effect is obtained by different means - e.g. the task PRINT FILE defined as "the content of FILE is output on paper" may be realised in one instance via a LASER-PRINT and in another via INC-PRINT[7].



7. Note that the definition of PRINT FILE is given in such a way that hardware heterogeneity is transparent at this level.

Purpose heterogeneity is a measure of the replication (distribution) of tasks throughout the community - the three categories shown above represent all possibilities. We consider each alternative with the aim of demonstrating that the type of knowledge which needs to be represented at the ARCHON layer and the reasoning associated with this knowledge differs with each of the three options; thereby illustrating that this classification is a relevant consideration when designing multiple agent systems.

In situations in which there is no overlap in the tasks that can be performed, deciding which agent should be asked to complete the task is trivial as each task can only be performed by one agent. In such circumstances the important consideration is that of ordering agent's tasks with the aim of minimising the impact of dependencies between agents' activities[8]. To complete this process, the following knowledge is needed at the AL:

- which agent can potentially undertake a certain task (*capability knowledge*)
- what task-dependencies exist (*domain knowledge*)
  (i.e. execution of which tasks preclude/enable execution of others)
- what tasks other agents intend to undertake (*intentional knowledge*)

Knowing the first piece of information enables the unique agent which can perform the particular task to be identified, the other two pieces are needed to schedule activities so that potential conflicts can be dealt with and the most urgent services can be identified and given highest priority.

At the other extreme, task allocation in environments in which there is a complete overlap of tasks requires more sophisticated reasoning (eg which agents can perform the task "better", faster, with less consumable resource, etc.), but the dependency problem is less crucial (an agent can usually rely on others to perform critical tasks). Furthermore, all agents are able to reason about premises and consequences of task execution (we assume an agent which is able to perform a certain task also has this information), implying that task-dependencies are common knowledge. Such common knowledge enables agents to make assumptions about each others behaviour and needs. Under these circumstances, the knowledge required (over and above that detailed for non-overlapping tasks) in order to decide which agent should be requested to perform the task is as follows:

- quality of the task performed by each agent (*evaluative knowledge*)
  (e.g. agent1 performs task1 with the fastest algorithm, agent2 performs it most accurately, etc.)
- agents's status (workload, current activity, etc.) (*state knowledge*)

This overlap also means that mechanisms must be provided for determining whether a task should be performed locally or whether an acquaintance should be asked, what cooperation protocol (eg client/server, contract net) should be used to establish the cooperation, how many agents should be run that task (more agents implies greater reliability and speed but greater

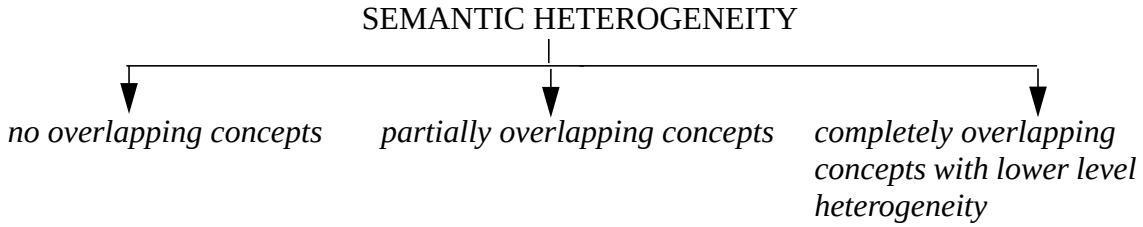
8. Once a task has been allocated (through whatever mechanism), the recipient may decide to execute the task immediately or with some delay. This decision should be based both on the local activity (are other important tasks active at the moment?) and on the knowledge of the urgency of that task for the overall community's activity (is some agent waiting for the execution of that task? can a solution be achieved if the execution of the task is delayed?). We deal with these issues more thoroughly in [Roda&91].

overhead in ensuring consistency of results). None of these criteria are pertinent if a task can only be performed at one place within the community

If the domain is partitioned into partially overlapping tasks, sophisticated reasoning structures are required both for the task allocation problem and for intelligent ordering of tasks to minimise dependencies. Each agent may have to decide if it should perform tasks locally or pass them to other agents, but it may not posses all the knowledge to reason about the task (this missing knowledge should be supplied by the AL). As an example[9] of this latter case, assume a DLS is able to diagnose faults on a low voltage electricity network (LV-DLS), but it does not know that the low voltage network is fed from a high voltage network. Given a description of the network's symptoms the LV-DLS can produce a diagnosis. The AL of the LV-DLS is aware of another DLS which performs high voltage diagnosis (HV-DLS). Once the LV-DLS's diagnosis has been received, its AL will either check this against the HV-DLS diagnosis or ask the HV-DLS to supply one. In such circumstances, it is obvious that the AL needs significantly more information about the problem domain and about the activity of the DLSs than that needed in the case of homogeneous systems.

4.2 Semantic Heterogeneity

As the notion of a task was central to purpose heterogeneity, so "concept" is central to semantic heterogeneity. A concept can informally be defined as an entity which the agent represents in its encapsulation of the world. So in the electricity distribution example: transformers, substations, voltages, etc. are all examples of concepts. In what follows we are concerned solely with domain level concepts, although it is clear that cooperation level concepts do exist (eg notions of agent workload, current activity and areas of interest). As was the case with purpose heterogeneity, it is possible to provide exhaustive coverage using three categories:



Notions of semantic heterogeneity become important when agents want to exchange domain - level information. Such interchange may be requested (eg the outcome of a task) or may occur spontaneously if the generator believes the information will be beneficial to the receiver.

If the agents are distributed in a way in which they have no overlapping concepts, then this type of interchange is pointless, the recipient will simply not understand (nor have any use for) any information it receives! It would not even be possible to transfer concepts between agents because there is nothing upon which such concepts could be described. Therefore if any degree of meaningful interaction is desired, there must be at least partially overlapping concepts (objects which can be understood, at some level, by two or more agents).

9. This is a real world problem taken from an ARCHON application

If there is at least partial overlap, then it is possible for agents to interact at the level of exchanging values for such concepts. Such exchanges may serve to provide a value if one is not already known (saving computational effort and making it immediately available) or verify an existing value. However there is also the possibility of one agent “teaching” another new concepts, based on existing shared definitions - eg if two agents both understand the concepts of voltage and resistance and one of them is aware of Ohm’s Law then it can teach the other about current. Such interchange requires the teacher to know what concepts the pupil is familiar with as well as their relationship to the new concept and it must be able to convey such ideas in a manner which is meaningful to the recipient.

With complete overlap of concepts between two agents, the only problems which arise are concerned with syntax. So somewhere in the interaction between American and English agents there would need to be translation from concepts such as candy, hood and sidewalk into sweets, bonnet and pavement respectively. This purely syntactic translation can be achieved by means of a dictionary (look-up table) which converts between the two languages; transformations may take place at the sender, the receiver or both if a common interchange language is used.

Merely knowing which concepts are understood by others is of little use unless meta-information is available to describe when and why the information is relevant. Without this additional information, exchange would either have to be random, total (all information exchanged) or non-existent. This meta-knowledge can therefore be regarded as the basis for intelligent information interchange.

5. Conclusions

Within the DAI community there have been little concerted effort spent on analysing the effects of heterogeneity on the design and operation of cooperating multi-agent systems. However whilst attempting to develop real-size industrial applications for process control we were faced with the inescapable fact that such problems are inherently heterogeneous. This paper aims to bridge this gap, offering an abstract analysis, based upon practical experience. We have proposed a classification hierarchy and have given illustrations of the two forms of heterogeneity which have greatest impact on the cooperation process - these two being the way in which task capabilities are distributed between community members and also the degree of overlap of understanding of domain level concepts. We assume that, in real world applications, agents in a multi-agent system base their reasoning about interaction on some knowledge about their acquaintances. Whilst we believe that this knowledge is necessary we stress that it cannot be complete and consistent at all times during computation.

Components of this knowledge becomes particularly important in presence of certain types of heterogeneity. The amount and type of knowledge which agents need to maintain and update varies with the type of heterogeneity in the system. Furthermore, semantic heterogeneity has a strong impact on the amount of communication required for the agents to achieve agreements on the activity to perform and on the results obtained. In our implementations we have mainly dealt with purpose heterogeneity rather than semantic heterogeneity meaning that in our experiments the agents would understand what others asked them to do, or what data another agent required. Obviously agents cannot coordinate their activity if they have not a common understanding of each others activity (tasks). However, the level of detail to which this common understanding should be achieved is strongly dependant on the type of reasoning that is performed at the task

level. For this reason we believe that a study on heterogeneous systems should firstly define the requirements for cooperation in the case of (no/partial) overlapping tasks and then define how to achieve a common understanding at the semantic level.

6. References

[Avouris&89] Avouris,N.M., Liedekerke,M.H.V., & Sommaruga,L., (1989), "Evaluating the CooperA Experiment", in proc. of 9th DAI Workshop, pp 351-366.

[Brandau&89] Brandau,R. & Weihmayer,R., (1989), "Heterogeneous Multi-Agent Cooperative Problem Solving in a Telecommunication Network Management Domain", in proc. DAI workshop, pp 41-57.

[Bratman90] Bratman,M.E., (1990), "What is Intention?", in Intentions in Communication (eds P.R.Cohen, J.Morgan & M.E.Pollack), pp 15-31.

[Cammarata&83] S.Cammarata, D.McArthur & R.Steeb, (1983), "*Strategies of Cooperation in Distributed Problem Solving*", Proc IJCAI, pp 767-770.

[Cohen&90] Cohen,P.R. & Levesque,H.J., (1990), "Persistence, Intention and Commitment" in Intentions in Communication (eds P.R.Cohen, J.Morgan & M.E.Pollack), pp 33-71.

[Fox81] M.S.Fox, (1981), "*An Organisational View of Distributed Systems*", IEEE Trans. on SMC, 11, 1, pp 70-80.

[Galliers89] J.R.Galliers, (1989), "The Positive Role of Conflict in Cooperative Multi-Agent Systems", European Workshop on Modelling an Autonomous Agent in a Multi-Agent World.

[Gasser&88] Gasser,L., Braganza,C. & Herman,N., (1988), "MACE: A Flexible Testbed for Distributed AI Research" in Distributed Artificial Intelligence (ed M.N.Huhns), pp 119-153.

[Gaussens90] E.Gaussens, (1990), "*Needs and Opportunities for Expert Systems in the Process Control Field*" Vacation School for Process Control, University of Strathclyde, Scotland.

[Genesereth&86] Genesereth,M.R., Ginsberg,M. & Rosenschein,J.S., (1986), "Cooperation without communication". Proc. 5th AAAI, Philadelphia, PA Aug. 1986, pp.51-57

[Huhns&90] Huhns,M.N, Bridgeland,D.M. & Arni,N.V., (1990), "A DAI Communication Aide" MCC Technical Report, ACT-RA-317-90.

[Lenat75] D.B.Lenat, (1975), "*BEINGs: Knowledge as Interacting Experts*", Proc IJCAI, pp 126-133

[Lesser&80] V.R.Lesser & L.D.Erman, (1980), "*Distributed Interpretation: A Model and Experiment*", IEEE Trans. on Computers, 29, 12, pp 1144-1163.

[Pollack90] Pollack,M.E., (1990), "Plans as Complex Mental Attitudes", in Intentions in Communication (eds P.R.Cohen, J.Morgan & M.E.Pollack), pp 77-105.

[Power84] R.Power, (1984), "*Mutual Intention*" Journal for the Theory of Social Behaviour, 14, pp 85-101.

[Roda&91] C.Roda, N.R.Jennings & E.H.Mamdani, (1991), "*ARCHON: A Cooperation Framework for Industrial Process Control*", International Working Conference on Cooperating Knowledge Based Systems, University of Keele, England, Springer-Verlag (to appear).

[Rosenschein&89] Rosenschein,J.S. & Breese,J.S., (1989), "Communication-free interaction among rational agents: a probabilistic approach" in L.Gasser and M.N. Huhns eds. "Distributed artificial intelligence. Vol. II". Pitman London, Morgan Kaufmann, San Mateo CA. 1989.

[Steiner&90] Steiner,D.D., Mahling,D.E. & Haugeneder,H., (1990), "Human Computer Cooperative Work", in proc. of 10th International Workshop on Distributed Artificial Intelligence, Texas.

[Werner89] Werner,E., (1989), "Cooperating Agents: A Unified Theory of Communication and Social Structure" in Distributed Artificial Intelligence Vol II, (eds L. Gasser & M.Huhns), pp3-37.