

Using Adaptation and Organisational Knowledge to Coordinate Mobile Agents

Steven Willmott and Boi Faltings

Laboratoire d'Intelligence Artificielle, Department Informatique,
Swiss Federal Institute of Technology, IN (Ecublens), CH-1015 Lausanne, Switzerland.
{willmott,faltings}@lia.di.epfl.ch

Abstract. Quality of Service (QoS) routing generally requires fast reaction times, tight coupling of interactions between routing systems and mechanisms for ensuring that actions taken throughout the network are coherent. [18] showed how an agent based QoS routing approach can benefit significantly from making controller agents mobile and allowing them adapt the information and control distribution in the network over time. This paper discusses how giving mobile agents organisational models can bridge the gap between the need for tight, fast coordination and freedom to move around the network. Furthermore coordination is achieved without imposing any globally or external controls on the mobile agents in the system.

1 Introduction

[18] describes how groups of agents performing Quality of Service routing can be made to adapt their working relationships over time with respect to the state of their environment. The resulting system is based on mobile controller agents moving around the network as resource availability patterns change. This paper describes how giving mobile agents models of their relationships to other agents in the system:

- Builds up an organisational structure
- Speeds up coordination for solving routing problems
- Controls agent mobility to ensure that the routing service is provided efficiently throughout the network

Section 2 of the paper gives an overview of an agent based Quality of Service routing approach, Section 3 describes the use of organisational models. Section 4 briefly covers preliminary results and Section 5 discusses the importance of the organisational approach. Section 6 concludes the paper.

2 Hierarchies of Agents for Quality of Service Routing

On-line, state-based Quality of Service (QoS)¹ routing presents challenging problems in the distribution of information and control. As the network state changes,

¹ See [5] for a good overview of existing techniques, issues and discussion of the need for on-line state based approaches.

information for routing decisions needs to be propagated and routing decisions made which respect current constraints. Hierarchical models (such as the ATM forum's PNNI architecture [2]) are often proposed to help deal with the complexity of the problem:

- The network is divided up into disjoint *regions* $r \in L_1$, each region grouping together one or more network nodes
- Subsets of these first tier regions are then grouped together at the next level up (L_2 into *meta-regions*)

This process continues recursively until a complete hierarchy is formed. Aggregation techniques can be used to generate summarised models of the network state at higher levels in the hierarchy. This aggregation abstracts away the detailed link state information to produce more manageable information models.

Going one step beyond the standard hierarchical models, the regions and meta-regions can be controlled by *controller agents* $c_{i/j}$ responsible for resource allocation decisions in each region $r_{i/j} \in L_j$. Agents controlling regions $r \in L_1$ are responsible for actual physical resources, agents controlling regions at levels L_2 and above have an aggregated view of the network state and are responsible for coordinating the actions of agents visible to them in the level below.

2.1 Static Hierarchy

To solve routing problems which traverse more than one region, controller agents need to interact with one another. The hierarchy created by the regions $r \in L_1$, $r \in L_2$ etc. can be used to define relationships between controllers which determine how information is propagated and how agents work together to solve individual routing problems. These temporally stable relationships are often called *organisational* relationships [7] whereas interactions carried out by agents to jointly solve individual routing tasks are classified as *coordination* (see [19] for example).

A basic agent based routing scheme for a hierarchy is shown in Figure 1. (Note that standard hierarchical routing schemes such as [17] and [2] for example often have non-hierarchical control flow.)

2.2 Adapting the Hierarchy to the Network State

The division of regions between agents and controllers represents a static distribution of information (local controller $c_{i/j}$ keeps track of the state in its part of the network $r_{i/j}$) and control ($c_{i/j}$ is responsible for all resource allocations in $r_{i/j}$). As has often been pointed out however, no distribution is optimal for every situation [15] - this is especially true in an environment as challenging and dynamic as a communications network. To address this problem, [18] describes an agent based approach which is able to change distribution of information and control according to the remaining bandwidth available on the links in the network. The division of the network into regions is based on a *blocking islands*

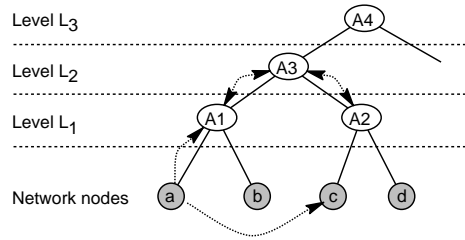


Fig. 1. Agents A1 and A2 are responsible for regions containing nodes a and b and c and d respectively. When A1 receives a demand for a route from node a to node c it is not able to allocate resource immediately since c is not in its own region. A3 is then requested to help, which it does by allocating resources between A2 (the owner of node c) and A1 before requesting both A1 and A2 to complete the remainder of the routes in their own regions.

clustering described in [6]. The clustering scheme defines a unique structure $AH(R)$ for any given configuration R of bandwidth availability on the network links. Briefly:

- Regions in the network are formed by grouping together nodes which are *mutually reachable* at a specified bandwidth level β Mbit/s
- Clustering is repeated at predetermined levels of the hierarchy with β decreasing at successively higher levels. The lowest level (L_1) therefore contains the smallest regions, clustering nodes connected by very high available bandwidth. Regions at higher levels are connected at progressively lower available bandwidths

The decomposition at any given moment is therefore dependent on R and changes over time as resources are allocated and de-allocated in the network. The abstraction proved very successful for off-line resource planning (see [6]).

The need for mobility: As described in [18], the blocking islands approach can also be applied to the on-line, state-based routing problem, extending the idea of a static hierarchy of routing agents (Section 2.1) and making the hierarchy adaptive to the network state. The added dimension of change over time adds the requirement that the previously stationary controller agents become mobile. Each region controller stays resident on one of the nodes in its domain to:

- Reduce the resources needed to propagate state data
- Minimise reaction time for routing decisions
- Make the system more robust by ensuring that the controller is co-located with its resources

These reasons make agent mobility an essential part of the approach, allowing the agent based routing service to adapt its configuration to the network state over time.

Coordination Problems: Unlike previous mobile agent approaches to routing which targeted telephone [1, 14], IP style [16] and ad-hoc [10] networks the approach described above requires a high degree of coordination between the agents providing the routing service. Previous approaches were able to minimise contact between agents by leaving markers in the environment to build up information about the network state (which is then used by an on-line packet forwarding or call allocation using routing tables). In none of the systems were single agents themselves responsible for making routing decisions. On-line, state-based QoS routing however, presents a rather different set of problems: 1) each demand must be routed individually as it arrives in the network (on-line), 2) Routing decisions are made on the basis of an up to date model of the network state (state-based) and 3) Resource allocation along the entire route must be coordinated (since each link in the chosen path must support the allocation and end-to-end constraints such as delay and cell loss must be checked).

These problems boil down to finding an effective distribution of information and control in the network. Dividing up the routing task to be performed as a service provided by a set of mobile agents therefore raises a number of serious coordination issues which must be addressed:

- Agents need to be free to move and adapt the information and control distribution to follow $(AH(R))$ in this case)
- Routing decisions need to be made very quickly, with a minimum of time spent locating agents with relevant state information and resources on offer
- Problem solving requires tight synchronisation between agents (to access all relevant information, ensure that the correct resources are reserved and check end-to-end constraints)
- Ensure that the routing service is available throughout the network at all times (an agent should not be able to renage responsibility for routing in a region $r_{i/j}$ without finding a replacement controller for the nodes concerned)

The remainder of the paper goes on to discuss how this level of coordination can be achieved without imposing external control structures on the agents in the system.

3 Introducing Organisational Knowledge

Dealing with the coordination problems identified in Section 2.2 is non-trivial. A further constraint is to keep any mechanisms added “agent centric”, that is to allow agents complete and local control over the coordination mechanism rather than imposing an *external* structure. This is vital for robustness and scalability. The approach introduced in [18] achieves this by providing each agent $c_{j/i}$ with three models:

- Information model $c_{j/i}(I)$
- Control model $c_{j/i}(C)$
- Organisational model $c_{j/i}(O)$

The information model represents the current network state of the $r_{j/i}$ and the control model represents the types of action $c_{j/i}$ is able to take during resource allocation tasks. These alone are sufficient for the agent to complete routing tasks autonomously in its own region $r_{j/i} \in L_i$. The organisation model $c_{j/i}(O)$ shown in Figure 2 represents $c_{j/i}$'s relationships to other controllers. The ensemble of the information contained in all the organisational models of all the agents at any one instant in time describes the state of the whole organisational structure.

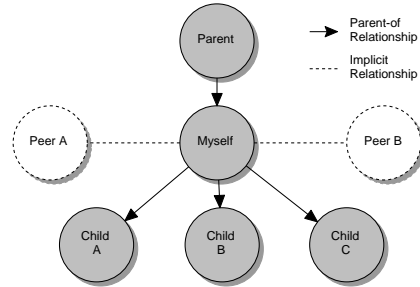


Fig. 2. In its organisational model, the agent maintains explicit relationships to its current child agents (the agents controlling regions in the level below its own) and its parent (the agent responsible for it in the level above). There are also implicit relationships to peer agents in its own layer, these are not maintained in the model - when contact is needed, the location and identity of the peer agent in question is provided by the parent agent.

Each $c_{j/i}$'s three I , C and O models are linked as follows:

- I determines the resources the C applies to
- Local changes in the I prompt the agent to change its relationships in O to its fellow agents (driving adaptation in the organisation)
- C determines which actions can be taken when allocating traffic (which influences I)
- C determines the actions which can be taken when cooperating with other agents in the I to solve routing problems
- O determines which agents are relevant when solving routing problems outside the scope of C (supporting coordination)
- Changes in O update the scope of I (managing coverage of the network with the routing service)

The organisational model is also the key to mobility in the agent system and adapting the routing mechanism to best match the network resources. As the environment (bandwidth state) changes, the agents have the option of locally updating the organisational structure in two ways:

- **Split:** an agent may decide to split the region $r_{j/i}$ into several new regions in L_i

- **Merge:** an agent may decide to merge its region $r_{j/i}$ with the region of an adjacent controller $c_{k/i}$

These two actions are all that is needed for agents themselves to gradually change the organisation to make it more like the uniquely defined structure $AH(R)$ for any given R .

3.1 Splitting Into Several Regions

As resources inside a region are allocated to traffic demands, the region may become disconnected at the bandwidth level the controller is working at (the β mentioned in Section 2.2). This change prompts the controller to consider dividing the region into two or more new regions which themselves connected at bandwidth β .

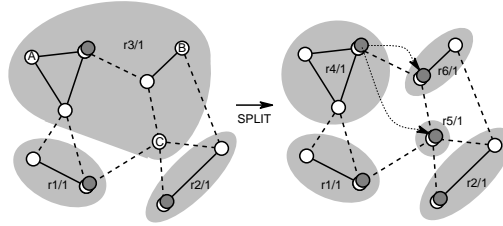


Fig. 3. The figure shows network with two types of link: links with less than β bandwidth remaining (dashed) and links with β or more bandwidth remaining (solid). The dark circles represent controller agents and the dotted arrows agent migration. During the split action, one large region is divided up between three new controllers.

Figure 3 shows the transition involved in a split. In this case, region $r_{3/1}$ is no longer connected at bandwidth level β (no traffic demand with a bandwidth requirement of β or more could be allocated between any of nodes A, B and C). This prompts the controller of region $r_{3/1}$ to consider splitting the region into three pieces. The splitting process involves the following steps:

- The agent determines the new regions $r_{4/1}$, $r_{5/1}$ and $r_{6/1}$ (this is uniquely defined by the bandwidth available on the links - known from $c_{3/1}(I)$)
- Creates a new controller agent for each region (these new agents will subsequently be peers of the current region controller)
- The new agents $c_{5/1}$ and $c_{6/1}$ each migrate to node which is in their new region, $c_{3/1}$ updates its I, C and O models and becomes $c_{4/1}$

The newly created regions are now once more internally connected at bandwidth level β (the organisational structure is equal to $AH(R)$ for bandwidth level β). Each new region controller has its own new I, C and O models which represent their place in the network and organisation.

3.2 Merging with a Neighbouring Region

As resources between two regions are freed up, these become candidates for a merger (since they could form a larger connected cluster). The parent agent supervising the two regions is able to detect this in its information model (which consists of nodes made up of the regions at the lower level and the links between them). The parent agent then sends a suggestion to one of the regions concerned to merge with the other.

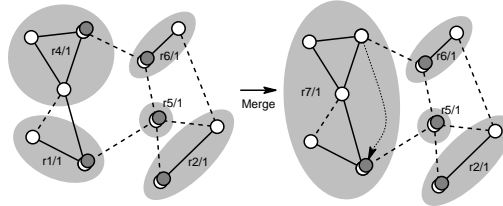


Fig. 4. Executing the merge action, the controller of $r_{4/1}$ moves to join up with the controller of $r_{1/1}$ to form region $r_{7/1}$

On receiving a suggestion to merge, the agent considers its current state and workload before deciding what to. If the agent decides to merge it:

- Contacts the other controller involved ($r_{1/1}$ in Figure 4) and requests a merger
- If the merge can go ahead, $c_{4/1}$ moves to the same node as $c_{1/1}$ and takes over its role and information model (merging $c_{4/1}(I, C, O)$ with $c_{1/1}(I, C, O)$)
- Thus the two mobile agents themselves merge - leaving only one controller with an enlarged region ($r_{7/1}$)

Figure 4 shows the transition involved in such a merge. The social interaction between $c_{4/1}$ and $c_{1/1}$ is very important to ensure that both parties are prepared for the merger to happen, $c_{1/1}$ might refuse a merger for several reasons including mergers or splits of its own or a large number of open routing tasks.

4 Implementation and Experimental Results

All testing was carried out in a distributed Java based test environment with the following core components:

- A purpose built agent platform supporting agent local/remote communication, management and configuration
- Agents running on each platform to model the roles in scenarios to be tested. Services modelled as agents include: network resources (implementing the ATM PNNI protocol suite (see [12]) for connection setup and tear down), traffic generation, standard QoS routing mechanisms, data collection, monitoring and the resource allocation organisations themselves

Mobility was implemented as simply as possible by caching the code for mobile agents at each node and transmitting only memory state between platforms. The benefits which would be achieved by true code mobility (flexibility, robustness and simplified code updates) are orthogonal to the results described in this paper. A final important feature of the platform (and one of the primary reasons for custom building the testbed and not using an existing framework such as [3]) is that all messages between platforms (network nodes) incur send delays equal to those they would incur if they took the shortest path in the real network (calculated using the known network link delays). Thus *all* network actions carried out by agents, including controller mobility, experience the appropriate network communication conditions.

Preliminary tests were carried on a model of the Compuserve USA backbone which has 11 nodes, 14 (45Mbit/s) ATM links and approximate links delays varying between 5 and 60ms.² Figure 5 compares results for the static hierarchy (SH), the adaptive hierarchy (AH) for a simple traffic scenario using increasing random traffic over a 10 hour simulation period. The graph summarises rejection percentages³ for both approaches averaged over 16 test runs (see [18] for preliminary comparisons with other techniques).

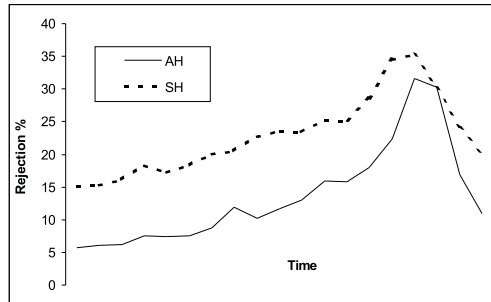


Fig. 5. AH rejects significantly less demands than SH (13% average v's 22% average over the whole period). Standard deviations (for variance between experiment runs) vary between 3% and 9% (larger variances at increased traffic load).

The results show that even though both hierarchies (SH and AH) had access to the same state information and worked under the same network conditions the mobility enabled AH performs significantly better than SH. The reasons behind the difference between SH and AH are discussed in detail in [18]. Briefly however, the differences are primarily due to the fact that SH agents often do not have access to pertinent network state information. For AH, bandwidth information is captured implicitly in the organisational structure of the agents

² Source: <http://www.caida.org/Tools/Mapnet/>.

³ Comparisons using the percentage of offered bandwidth accepted give comparable results.

and agents are able to focus on the other QoS parameters (in this case delay only). AH's organisation supports coordination more effectively than SH by maintaining links between agents which reflect their available network resources (and hence whether they might be part of a team solving a particular routing problem).

5 Achieving Globally Coherent Behaviour

The split and merge actions are purely local. That is, based on one or more agents' local models of their regions. Despite the local nature of the changes however, splitting and merging throughout the network leads to globally coherent behaviour. This is because the unique configuration $AH(R)$ is not simply a global network property but can be applied to any subset of nodes in the network. A local controller is therefore able to apply simple rules to its information model to check if the local organisational model requires an update. There are only two possible types deviations from the local part of $AH(R)$ which corresponds to $c_{i/j}$'s region:

1. The region is no longer connected at the specified bandwidth level β for the current level L_j . This can be solved by splitting the region as described in Section 3.1
2. The region contains subregions which are connected at a bandwidth level equal to or higher than the β for the level below (L_{j-1}). This can be addressed by suggesting the subregions to merge as describe in Section 3.2

Both of these types of deviation from $AH(R)$ can be detected simultaneously and occur multiple times in an agent's information model. In this case choosing any of the possible corrective actions eliminates one or more of the deviations and moves the local structure closer to $AH(R)$. Thus for any fixed R' , actions considered or taken by the agents in the network will always move the global structure closer to $AH(R')$. Agents are free to move around inside its own region but required to agree any other movement which might imply organisational change with the agents in adjoining agents. This coordination ensures that:

- No agent moves without handing over control of the nodes it is responsible for to the control of another agent. Information and control models are maintained continually cover the whole network
- The ensemble of the organisational models always forms a complete consistent hierarchy. Routing tasks herefore can always be carried out because links to agents relevant to solving the routing problem are always valid

Clearly however, as the agents make local changes to their organisational models to move towards $AH(R')$, R' may be changing. Organisational updates also incur communication costs and are therefore not instantaneous. Thus, in a highly dynamic network, $AH(R)$ may never be reached. However, as the results in Section 4 show, the agents get close enough to $AH(R)$ to perform significantly

better than a static hierarchy SH (see [18] for more details on the speed of change in the network scenario tested). Although the target $AH(R)$ is a moving one, it provides a focus for the organisational changes considered by the agents throughout the network.

5.1 The Importance of Organisation

Work on coordination in Distributed Artificial Intelligence can be divided into three broad categories:

- **Coordination Media and Languages:** these approaches support communication about coordinated action in various ways. Mobile agent related work includes Linda-like tuple spaces [11, 13], simple blackboards [4] or using the environment to support communication [1]
- **Coordination Mechanisms:** this includes protocols and conventions which agents can use to coordinate their actions for a single task or task episode. Whilst [9] gives an overview of multi-agent systems approaches to this problem, most of the mobile agent related work in this area is confined to simple examples describing the use of the coordination media above
- **Organisational Approaches:** organisational techniques are defined as building relationships between agents which last for more than one problem solving episode [8]. Very little work has been done in applying these techniques to mobile agent systems. One reason is perhaps that organisational structures are generally too restrictive to make good use of the mobile agent paradigm

The organisational approach presented here is vital to the QoS routing task because:

- It provides a way to maintain relationships between moving agents which can be used to tightly couple interactions between subsets of agents for brief periods of time
- The only medium that is required to support this is a shared point to point messaging system (provided by the agent platforms), there is no need for a blackboard (as in [4]) or message broadcasts (as in several multi-agent system coordination mechanisms)
- the organisation constrains the mobility of the agents to ensure that the information and control they hold in the network is always effectively distributed to cover the whole network

This goes beyond what can be achieved using coordination mechanisms based on single tasks since these require significant initial overhead for each task before beginning work (agents would need to continually keep re-finding each other). It also has certain advantages over proposed mobile agent coordination media since these provide mainly asynchronous communication (information is posted in a common space such a tuple space or on a blackboard) for other agents to find. This incurs a significant speed penalty over explicitly addressing individual agents, and in the case of blackboards has strong centralising effect on the system.

5.2 On-going Work

Current work on the approach is focused on the following main areas:

- Extensive testing using larger networks, more realistic traffic scenarios and comparing the routing techniques to other standard routing approaches
- Adapting the ATM routing techniques for use in IP networks
- Investigating the dynamics of the organisation

An important weakness in the current approach is that organisational change can only occur *explicitly* through the exchange of messages. This means that if a message or an agent is lost, the whole organisation may be out of step and never recover. An significant area of work is therefore adapting the agents to detect organisational relationships by themselves and allow them to recover from such failures.

6 Conclusion

This paper describes how giving mobile agents organisational models of their relationships to fellow agents enables them to perform tasks as complex as on-line, state-based QoS routing. The relationships maintained by the agents:

- Tightly couple interactions for individual tasks at given instants in time
- Enable fast coordination
- Ensure that the global behaviour of all agents is coherent

This coordination is achieved without imposing an external framework on the agents. As they move around the network, the agents are themselves responsible for keeping track of their relationships and all organisation, information and control knowledge is encapsulated in the agents themselves.

Acknowledgements

This work is partly supported by the Swiss National Science Foundation under project number 21-59081.99.

References

1. S. Appleby and S. Steward. Mobile Software Agents for Control in Telecommunications Networks. *British Telecom Technology Journal*, 12(2), 1994.
2. ATM-FORUM. P-NNI V1.0 - ATM Forum approved specification, af-pnni-0055.000. *ATM FORUM*, 1996.
3. M. Breugst and T. Magedanz. On the Usage of Standard Mobile Agent Platforms in Telecommunication Environments. In S. Trigila et al., editor, *Proceedings of 5th Int. Conference on Intelligence in Services and Networks (IS&N)*, Lecture Notes of Computer Sciences 1430, Intelligence in Services and Networks: Technologies for Ubiquitous Telecom Services, pages 275–286, Antwerp, Belgium, May 1998. Springer Verlag.

4. L. Cardelli and D. Gordon. Mobile Ambients. In M. Nivat, editor, *Foundations of Science and Computational Structures*, pages 140–155. Springer Verlag, LNCS 1378, 1998.
5. Shigang Chen and Klara Nahrstedt. An Overview of Quality of Service Routing for Next-Generation High-Speed Networks: Problems and Solutions. *IEEE Network*, pages 64–79, November/December 1998.
6. Christian Frei and Boi Faltings. Abstraction and Constraint Satisfaction Techniques for Planning Bandwidth Allocation. In *IEEE INFOCOM'2000*, Tel-Aviv, Israel, March 2000. In print.
7. L. Gasser. Social Concepts of Knowledge and Action. *Artificial Intelligence*, 47:107–138, 1991.
8. L. Gasser. DAI Approaches to Coordination. In N. M. Avouris and L. Gasser, editors, *Distributed Artificial Intelligence: Theory and Praxis*, pages 31–51. Kluwer, 1992.
9. N. R. Jennings. Coordination Techniques for Distributed Artificial Intelligence. In F. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 187–210. John Wiley & Sons, 1996.
10. N. Minar, K. H. Kramer, and P. Maes. Cooperating mobile agents for dynamic network routing. In A. L. G. Hayzelden, editor, *Software Agents for Future Communications Systems*, chapter 12. Springer-Verlag, Berlin Germany, 1999.
11. A. Omicini and F. Zambonelli. Coordination of mobile information agents in tucson. *Journal of Internet Research*, 8(5), 1998.
12. R. O. Onvural and R. Cherukuri. *Signalling in ATM Networks*. Artech House, 1997.
13. Gian Pietro Picco, Amy L. Murphy, and Gruiia-Catalin Roman. LIME: Linda meets mobility. In *Proceedings of the 21st International Conference on Software Engineering*, pages 368–377. ACM Press, May 1999.
14. Ruud Schoonderwoerd, Owen Holland, and Janet Bruten. Ant-like agents for load balancing in telecommunications networks. In W. Lewis Johnson and Barbara Hayes-Roth, editors, *Proceedings of the 1st International Conference on Autonomous Agents*, pages 209–216, New York, February 5–8 1997. ACM Press.
15. Y-P. So and E. H. Durfee. Designing Organisations for Computational Agents. In M. J. Prietula, K. M. Carley, and L. Gasser, editors, *Simulating Organisations*, pages 47–64. AAAI Press, 1998.
16. D. Subramanian, P. Druschel, and J. Chen. Ants and Reinforcement Learning: A Case Study in Routing in Dynamic Networks. . In *Proc. of IJCAI'97*, pages 832–838. 1997.
17. W. T. Tsai, C. V. Ramamoorthy, W. K. Tsai, and O. Nishiguchi. An Adaptive Hierarchical Routing Protocol. *IEEE Transactions on Computers*, 38(8):1059–1075, August 1989.
18. S. Willmott and B. Faltings. The Benefits of Environment Adaptive Organisations for Agent Coordination and Network Routing Problems. In *Proceedings of the Fourth International Conference on Multi Agent Systems (ICMAS-2000)*, page tba. IEEE Press (in print), 2000.
19. N. R. Wooldridge, M. J. amd Jennings. Formalizing the Cooperative Problem Solving Process. In *Proceedings of the 13th International Workshop on Distributed Artificial Intelligence (IWDAI'94)*, pages 403–417. 1994.