PERGAMON

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Engineering Applications of Artificial Intelligence 16 (2003) 191–201

Engineering Applications of
ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

Shifting matrix management: a model for multi-agent cooperation

G. Li[a], A.A. Hopgood[b,*], M.J. Weller[c]

[a] Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes MK7 6AA, UK
[b] School of Computing and Mathematics, The Nottingham Trent University, Burton Street, Nottingham NG1 4BU, UK
[c] Institute of Educational Technology, The Open University, Walton Hall, Milton Keynes MK7 6AA, UK

Abstract

Shifting matrix management (SMM) is a model of agent coordination inspired by Mintzberg's model of organizational structures. Mintzberg's model permits many temporary lines of authority, reflecting the multiple and shifting functions of a flexible workforce. In order to apply these ideas to agent cooperation, a six-stage framework has been devised. The resulting model has been compared with two standard models: Contract Nets and Cooperative Problem-Solving. All three models have been implemented by means of an in-house blackboard system, Algorithmic and Rule-based Blackboard System (ARBS). 'Disembodied' agents have been constructed whose components are spread between system modules, known as knowledge sources, and private partitions of the blackboard. Tests have been carried out in which the three models have been applied to a set of tasks involving the control of two robots. Within the narrow context of these tests, the SMM model out-performs the other two approaches in terms of its task completion rate, number of tasks completed, and avoidance of wasted efforts. It is argued that although the SMM model expends more time reasoning about its actions, this is likely to be more than offset by the resultant efficient use of resources.
© 2003 Elsevier Ltd. All rights reserved.

Keywords: ARBS; Agent; Blackboard system; Contract Nets; Cooperative Problem-Solving (CPS); Disembodied agents; Distributed artificial intelligence; Shifting Matrix Management (SMM)

1. Introduction

Research into the coordination and cooperation of multiple robots is widely recognized as a step towards high performance and flexibility in industrial automation (Cardarelli et al., 1994; Koivo and Bekey, 1988; Paljug and Yun, 1993). Firstly, cooperating robots can perform tasks that are either difficult or impossible for a single robot, e.g. where a large object needs to be gripped from two ends. Secondly, distributed task performance amongst multiple robots can result in reliable performance and graceful degradation. Thirdly, well-organized multiple robots working in parallel and cooperatively can achieve high efficiency.

Two contrasting approaches to coordination and control of multiple robots are through either a centralized computer system or a cooperative distributed problem-solving (CDPS) approach (Durfee et al., 1989). This paper focuses on the latter approach, in which the tasks of information gathering, planning, and synchronizing robot actions are devolved to intelligent autonomous agents. We have proposed a new model for coordinating the behaviour of the agents, namely shifting matrix management (SMM) (Li et al., 1997). Here, we build on this work by describing a practical implementation and a comparison with two standard models, namely Contract Nets and Cooperative Problem-Solving (CPS). All three models have been implemented by means of a blackboard system, described in Section 2.

The SMM model of agent coordination has been inspired by Mintzberg's SMM model of organizational structures (Mintzberg, 1979), illustrated in Fig. 1. Unlike the traditional hierarchy, conventional matrix

*Corresponding author. Tel.: +44-115-848-6482; fax: +44-115-848-6518.
E-mail addresses: g.li@open.ac.uk (G. Li), adrian.hopgood@ntu.ac.uk (A.A. Hopgood), m.j.weller@open.ac.uk (M.J. Weller).

0952-1976/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0952-1976(03)00052-6

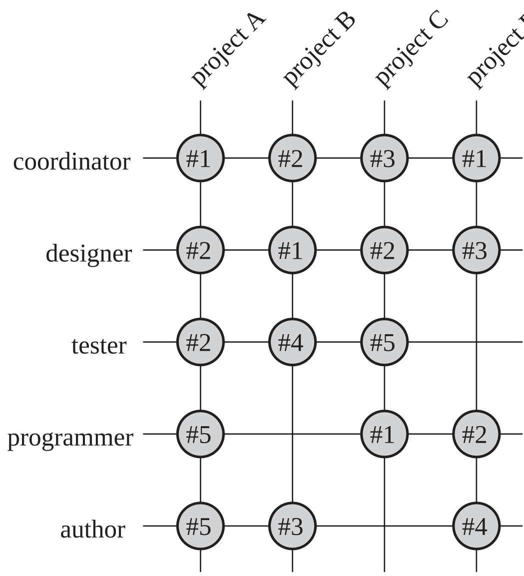

Fig. 1. An example of SMM in an organization, where the nodes represent individuals (Mintzberg, 1979).

management allows multiple lines of authority, reflecting the multiple functions expected of a flexible workforce. SMM takes this idea a stage further by regarding the lines of authority as temporary, typically changing as different projects start and finish.

2. Algorithmic and Rule-based Blackboard System (ARBS)

ARBS is an in-house blackboard system that has been progressively developed and successfully used in diverse applications (Hopgood, 1994; Hopgood et al., 1998, 1993). Although a distributed version of ARBS has recently been developed (Wong et al., 2001), the current work uses the standard serial architecture. In blackboard systems, a model of the problem and its solution (or part solution) evolve in a centralized memory area, viz. the blackboard (Nii, 1986). Domain knowledge is divided into modules called knowledge sources (KSs) which, in principle, can use any form of knowledge representation. Thus, each aspect of the domain knowledge can be represented in its most natural form. Communication between KSs is always indirect, via the blackboard.

In ARBS, the KSs can be rule-based (with a choice of inference modes), case-based, fuzzy, procedural, genetic algorithm or neural network (Fig. 2). Rule-based KSs comprise the rules and an associated inference mode to apply them to the current state of the problem. The supported inference modes in ARBS are directed forward chaining with multiple instantiation of variables, directed forward chaining with single instantiation of variables, or 'hypothesize and test'. Details of these three inference modes are given elsewhere (Hopgood, 2001). The size and complexity of individual KSs can vary from simple structures which run a single procedure to large rule sets which access many procedures and functions (Fig. 3).

Only procedural and rule-based KSs are used in the current study. In rule-based KSs, the inference engine determines which rules to apply and when to apply them. In this application, directed forward chaining is used by all rule-based KSs (Hopgood, 2001, 1994). With this mechanism, a rule dependence graph is built automatically for each rule-based KS prior to running the system. For each rule in a KS, the rule dependence graph shows which other rules may enable it and which rules it may enable. The inference engine uses the dependence graph to minimize the time spent examining inapplicable rules, so the benefit is greatest for the largest KSs. In some of the smaller rule-based KSs there are no interdependencies between the rules, so the inference mode defaults to 'first come, first served'.

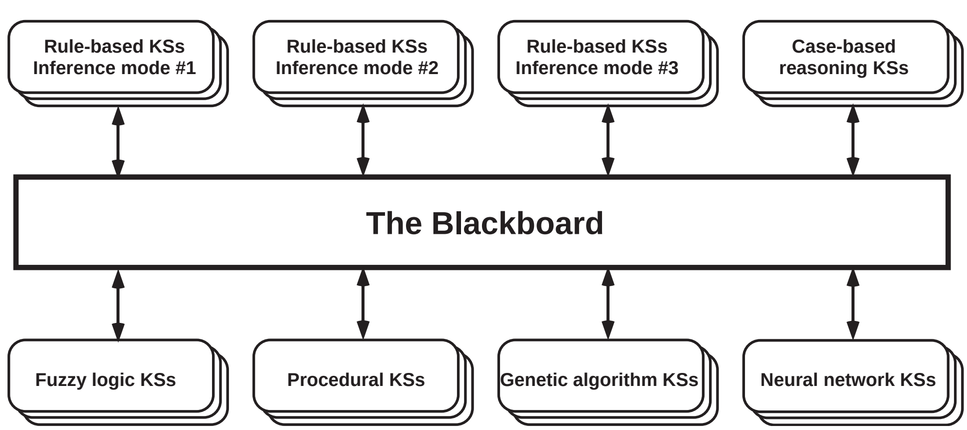

Fig. 2. The ARBS architecture.

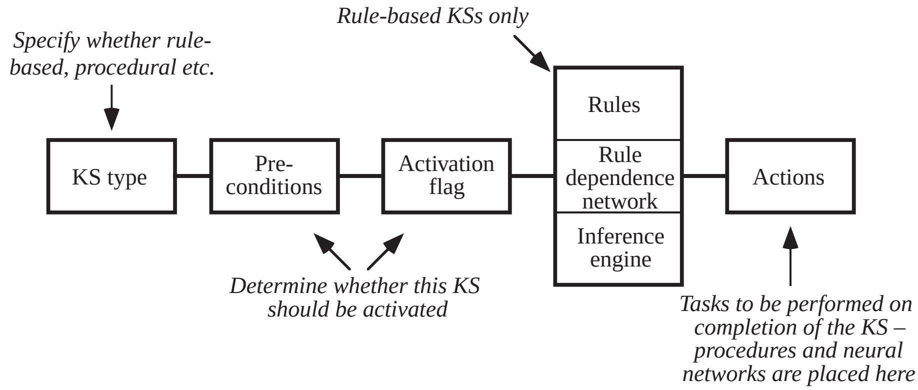

Fig. 3. A knowledge source in ARBS.

3. Agent and task representation in ARBS

Our intention was to build complex autonomous agents capable of both deliberative and reactive behaviours. A deliberative, or BDI (beliefs, desires and intentions), architecture involves planning and decision-making using a symbolic model. Coupled with this, the agents were also required to fulfil the role of 'actor', i.e. to display reactive behaviour in order to drive the motors of two or more physical robots (Fig. 4). Thus each agent requires the capabilities of an entire blackboard system, rather than those of a single KS. In order to achieve this aim without the overhead of implementing an entire blackboard system for each agent, *disembodied agents* were used, each of which comprises a share of rules from across the KSs and a share of the blackboard (Fig. 5). To accommodate local agent-specific information on the blackboard, the blackboard has been divided between public and private partitions. The private partitions are only accessible by the corresponding agent, which uses its partition to represent its developing world model. The public partitions correspond to the conventional use of a blackboard, i.e. as the means of communication between rules and KSs and, in this case, agents.

Although disembodied, each agent is encapsulated by the use of private blackboard partitions. As the system described here was implemented on a single serial computer, the agents cannot be truly autonomous. Nevertheless, some apparent parallelism is achieved since, when a given KS is active, it may be acting on behalf of several disembodied agents.

Each KS performs a step in the achievement of a task (Fig. 6). Therefore, a control scheme is needed in order to determine which KS should be activated at any given moment, given the current state of the blackboard. In a classical blackboard system, KSs are said to be opportunistic, i.e. they will be activated as soon as they can make a contribution to the overall process (Nii, 1986). In this way it is possible for each agent to maintain its own thread of control by the successive activation of its corresponding KS components. In ARBS, this ideal is approached by the attachment to KSs of preconditions, which determine which of the KSs are applied at any given time in accordance with the current information on the blackboard.

Reactive behaviour is achieved either through procedures attached to rules or by the inclusion of procedural KSs within the definition of the distributed agent. In the first case the reactive behaviour is dependent on a rule, in the second case it is dependent on the preconditions of the procedural KS.

The above representation of agents and task steps has the following advantages:

- *Agents and task steps are distinct*: Task steps, represented by KSs, can be added or deleted without modifying agents' roles in other tasks. Similarly, alteration of an agent's role in a task step will not alter other steps.
- *Efficiency of information exchange*: When a KS is active and a task step is underway, the exchange of information between agents, via the blackboard, is focused on that specific step.
- *Agent cooperation at each step*: An agent's contribution to the overall task is necessarily divided into contributions at different steps, increasing the scope for agent cooperation at each step.
- *Ease of alteration of behavioural strategy*: The behavioural strategy is implemented as procedural code which can be altered independent of the calling rules or KSs.

Although the blackboard architecture has been used elsewhere for the implementation of agent behaviour (e.g. Hayes-Roth et al., 1995), we believe the implementation described above, based on disembodied agents, to be novel. Three models of agent coordination, two standard and one novel, were implemented using this approach. They are described in the following sections.

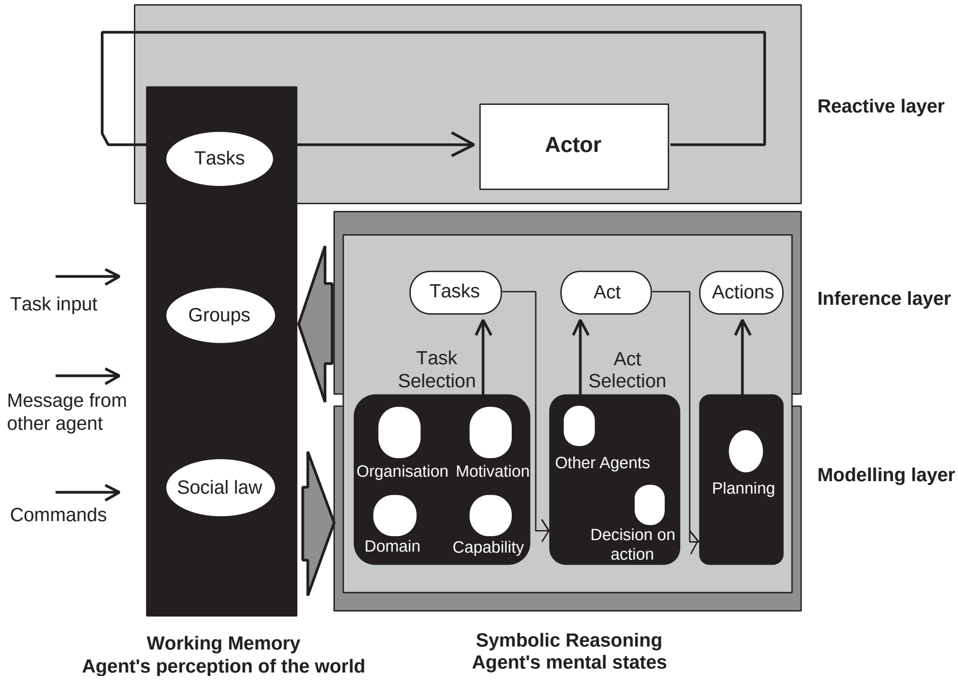

Fig. 4. A hybrid agent architecture.

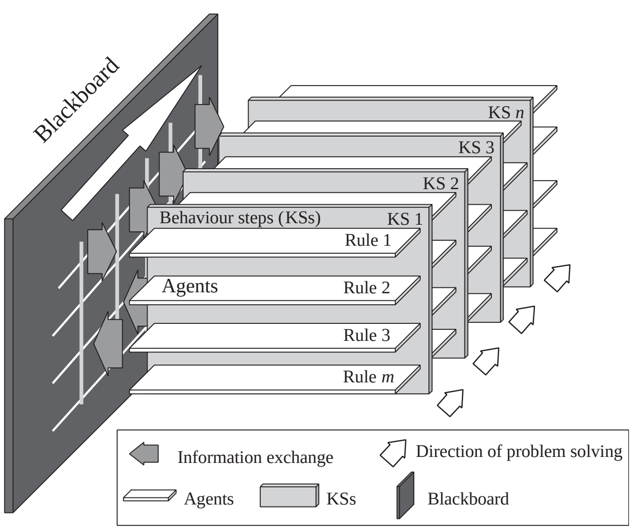

Fig. 5. Representation of an agent in ARBS.

4. Contract Nets

In the Contract Net model (Smith and Davis, 1981), a contract is an explicit agreement between an agent that generates the task (the manager) and an agent willing to execute the task (the contractor). The manager is responsible for monitoring the execution of the task and processing the results of the executions. The contractor is responsible for the actual execution of the task. The individual agents are not designated *a priori* as manager or contractor; these are only roles, and any agent can take on either role dynamically during problem-solving.

To establish a contract, the managers advertise the existence of the tasks to other agents. Agents that are potential contractors evaluate the task announcements and submit bids for those to which they are suited. An individual manager evaluates the bids and awards contracts for execution of the task to the agents it determines to be the most appropriate. The manager and contractor are thus linked by a contract and communicate privately while the contract is being executed. The managers supply mostly task information and the contractor reports progress and the eventual result of the task. The negotiation process may recur if a contractor subdivides its task and awards contracts to other agents, for which it is the manager.

In order to implement the Contract Net model within ARBS, the following alterations were made to the generic organizational structure shown in Fig. 6, resulting in the set of four KSs shown in Fig. 7.

- *Interface KS* becomes the *advertising KS*, which comprises a single agent that acts as the general manager seeking agents to perform tasks. Like the "task manager" in Smith and Davis' (1978) distributed sensing system, it has extensive processing capabilities but no task performance capabilities.
- *Goal selecting KS* and *act selecting KS* become the *bidding KS*. This KS comprises agents that watch task advertisements and submit bids to the task manager.
- *Planning KS* becomes the *awarding KS*.

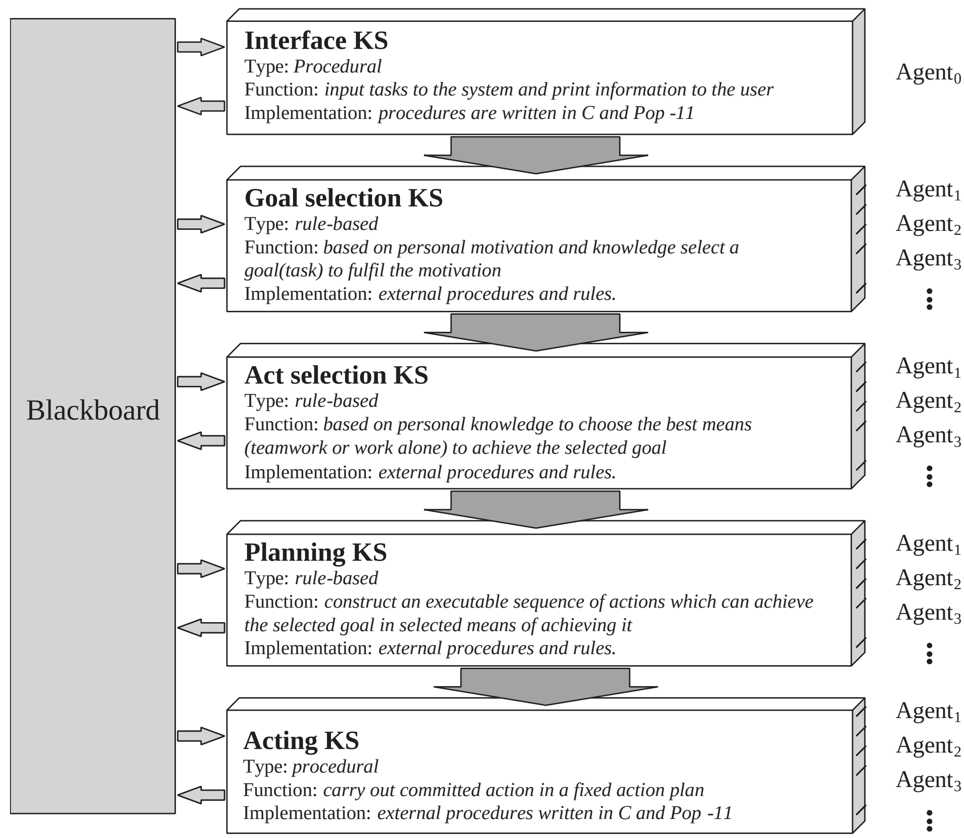

Fig. 6. Using knowledge sources to represent tasks.

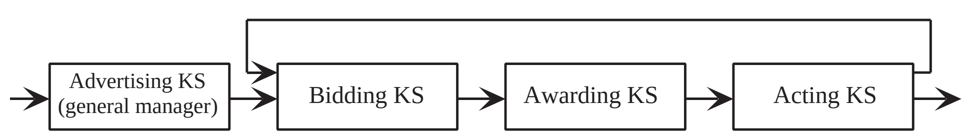

Fig. 7. KSs for implementing Contract Nets.

- The *acting KS* is retained, but the agents' rules are altered to enable the decomposition of combined tasks. The *action* field of the KS is used to activate the bidding KS so that the decomposed subtasks can be advertised.

5. Cooperative Problem-Solving

The CPS model comprises four stages which provide a top-level specification for agent coordination (Wooldridge and Jennings, 1994a, b). The model incorporates the benevolent assumption, i.e. the assumption that an agent is willing to perform all tasks requested of it and to volunteer its services to others, even though such benevolence arguably represents an erosion of autonomy. An agent's intentions determine its personal behaviour at any instant, while joint intentions control its social behaviour (Bratman, 1987). An agent's intentions are shaped by its *commitment*, and its joint intentions by its social *convention*. The four stages of the model are:

1. *Recognition*: Here some agents recognize the potential for cooperation with another agent that is seeking assistance, possibly because it has a goal it cannot achieve in isolation.
2. *Team formation*: Here the agent that recognized the potential for cooperative action at Stage 1 solicits assistance. If successful, this stage ends with a group having a joint commitment to collective action.
3. *Plan formation*: Here the agents attempt to negotiate a joint plan that they believe will achieve the desired goal.
4. *Team action*: Here the newly agreed plan of joint action is executed. By adhering to an agreed social convention, the agents maintain a close-knit relationship throughout.

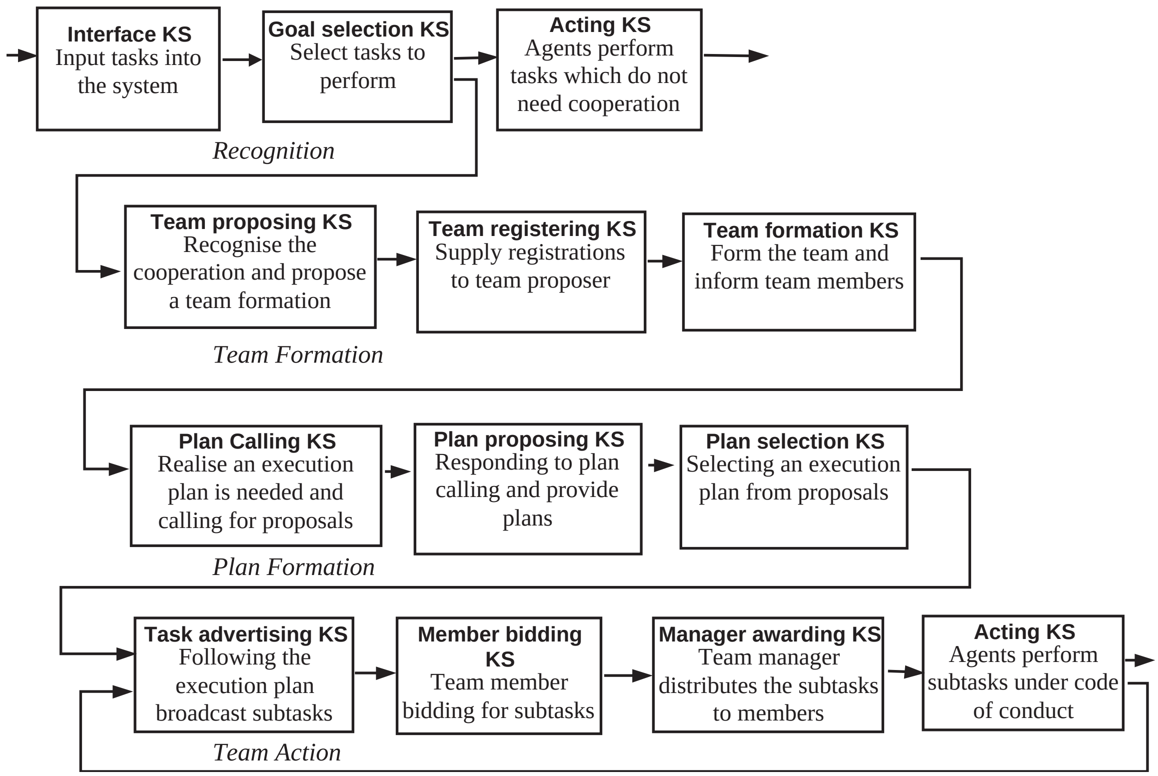

Fig. 8. KSs for implementing the CPS model.

Fig. 8 shows that, in order to implement the CPS model within ARBS, the following alterations were made to the generic organizational structure of Fig. 6:

- The blackboard contains two new public partitions: one for storing a team's commitments and conventions and another for its plan proposals and agreed plan.
- Each agent has a new private blackboard partition to store information about its current team.
- The act selection and planning KSs have been expanded.

6. Shifting matrix management

SMM (Li et al., 1997) is a model of agent coordination that has been inspired by Mintzberg's SMM model of organizational structures (Mintzberg, 1979), described in Section 1 and illustrated in Fig. 1.

The agents are distinguished by their different motives, functionality, and knowledge. These differences define the agents' variety of mental states with respect to goals, beliefs and intentions. The tasks need to be allocated weights describing their benefits and costs to individual agents and to the community of agents. The former are the self-benefit and self-cost; the latter are social benefit and social cost. Four different motivational stances towards tasks and subtasks have been implemented based on these weights:

1. *benevolent*—the agent is willing to perform all the tasks within its capability;
2. *selfish*—the agent requires that self-benefit weight > self-cost weight;
3. *social*—the agent requires that social benefit weight > social cost weight;
4. *socially responsible*—the agent requires that (social benefit weight + self-benefit weight) > (social cost weight + self-cost weight)

Additionally, each agent maintains a view of each other agent, represented as a probability of forming a team with that agent. These probabilities are initially set at 0.5, except for the non-selfish agents' view of the selfish agents, which was set to 0.4. This reflects an opinion at the outset that selfish agents are likely to be poor team-workers.

In order to apply these ideas to agent cooperation, a six-stage framework has been devised, outlined below.

1. *Goal selection*: The SMM model starts when a set of new tasks is generated. The first action performed by agents in a cooperation process is to select tasks to perform related to their initial mental states. At the end of this stage, agents should have settled goals.

2. *Act selection*: In this stage, agents select and adopt a way in which their settled goals can be achieved. Normally, there is more than one way of achieving a goal. For example, an agent that recognizes its intended goal is common to other agents would have to decide, on the basis of its rules, whether to bring about the goal in isolation or in collaboration with other agents. The decision is dependent on a number of factors, such as the agent's individual mental behaviour, the chances of an action taking place and being completed, and the agent's evaluation of the consequences of the action. At the end of this stage the agents have decided the way of achieving their intended goals.
3. *Team formation*: In this stage, the agents that decided to achieve their goals in a cooperative manner attempt to organize performance teams. The establishment of a team requires creation of a *common rule*, a *base of shared resources* within the team, a *common measure of performance* and *mutual adjustments* among team members.
4. *Plan formation*: One way of maximizing team resources and functionality is to have a joint plan for achieving the intended goal. The workload among team members can then be redistributed according to this joint plan so that work is performed by the most suitable agents. Plan formation is a process of negotiation in which team members jointly attempt to reach agreement on the joint plan.
5. *Team action*: In this stage, the newly agreed joint plan is executed by the team members under the code of conduct established during team formation.
6. *Shifting*: The last stage of the cooperation process, which marks the disbanding of the team, involves shifting agents' goals, positions, and roles. Mutual adjustments take place in which each agent updates its probability of team-working with other agents, depending on whether or not the completed team-working experience with that agent was successful. This updated knowledge is important as iteration through the six stages takes place until all the tasks are accomplished.

Fig. 9 shows that, in order to implement the SMM model within ARBS, the following alterations were made to the generic organizational structure of Fig. 6:

- Three new private blackboard partitions have been introduced for each agent to represent its motives, decision-making functions, and benefits achieved. The latter can be compared against the motives in order to measure success.
- As with the CPS model, the act selection and planning KSs have been expanded.
- A new KS, the shifting KS, has been implemented to carry out the sixth stage of the model.

7. Tests

7.1. Method

A series of tests were devised to demonstrate both the practicality of the SMM model and the novel use of a blackboard system, comprising disembodied agents, as a means for building agent-based systems. The three different models of agent cooperation were used to control two robots that were required to work together to complete a variety of tasks. This set-up was chosen because it is non-trivial, often requiring the breaking down of tasks into subtasks. The robots were model MA2000, produced by Tech-Quipment Ltd. Their physical functionality was identical, but they covered different regions of the working space. Although only two robots were involved, this did not constrain the number of agents. In fact, six agents were used in order to demonstrate multi-agent cooperation.

In order to test the systems, a total of 50 different tasks were generated, as shown in Table 1. Of these, 20 were *atomic tasks*, e.g. open or close the gripper, raise or lower the elbow, and roll the robotic hand. Either robot can perform these atomic tasks, and any of the more complex tasks are eventually performed by a combination of them. The time needed to complete an atomic task ranges between 1 and 4 s. A further 20 *combined tasks* could not be performed by any one robot in a single act. They either needed the robots to work together, or for one robot to perform a series of atomic tasks. The combined tasks comprised between two and six atomic tasks. Finally, 10 *complex tasks* were generated, which cannot be simply decomposed into a list of subtasks for a single robot and for which the task decomposition often has several alternative plans. The complex tasks comprised 2–16 atomic tasks. One example was the well-known Tower of Hanoi problem, which can be achieved in different ways depending on the problem-solving strategy adopted by the agents. Although there is an algorithm for the most efficient solution, this was unknown to the agents. Each of the 50 tasks was presented twice, creating a set of 100 tasks that were presented in random order.

For comparability, each system was set up with six agents of the same structure but different in-built knowledge. In addition to its six agents, the Contract Net model had a task manager agent which was not involved in the actual performance of any task. The CPS model had six benevolent agents, two of which had a stated preference for team-work as part of cooperative conditions specified in the *recognition* stage. These agents would always attempt to achieve a goal by team-work. In contrast, the other four agents only attempt team-work when they are unable to achieve the goal. Of the six agents in the SMM model, two were

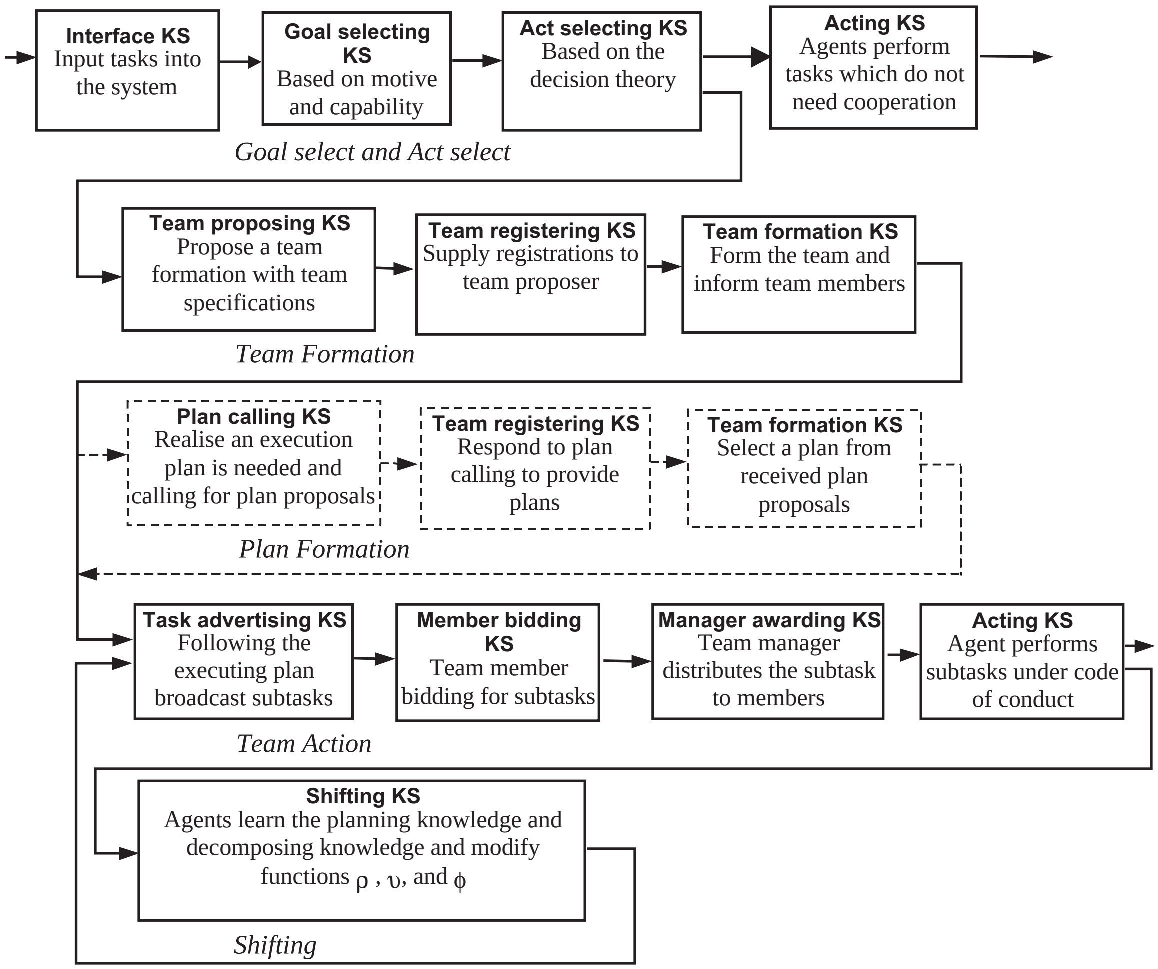

Fig. 9. KSs for implementing the SMM model.

Table 1
Tasks used for tests

Atomic tasks	Weight	Combined tasks	Weight	Complex tasks	Weight
Park1	10, 0, 50, 20	Hide	90, 30, 70, 40	Move(front, back)	140, 60, 80, 40
Bend_over	10, 0, 60, 20	Lay_on_left	100, 20, 50, 10	Move(left, right)	140, 60, 80, 40
Lay_down	20, 10, 60, 20	Lay_on_right	100, 20, 50, 10	Move(left, back)	140, 60, 80, 40
Sit_down	20, 10, 60, 20	Lay_on_front	100, 20, 50, 10	Move(front, right)	140, 60, 80, 40
Left	20, 15, 50, 10	Lay_on_back	100, 20, 50, 10	Move(position1, position4)	160, 60, 100, 20
Right	20, 15, 50, 10	Sit_left	100, 30, 60, 10	Transport(inter_front, inter_back)	160, 70, 100, 40
Front	20, 15, 50, 10	Sit_right	100, 30, 60, 10	Transport(inter_front, inter-right)	160, 70, 100, 40
Back	20, 15, 50, 10	Sit_front	100, 30, 60, 10	Transport(inter_left, inter_back)	160, 70, 100, 40
Grip	10, 0, 60, 10	Sit_back	100, 30, 60, 10	Transport(inter_left, inter_right)	160, 70, 100, 40
Release	10, 0, 60, 10	Left_grip	90, 20, 70, 25	Hanoi_tower(3, 5)	200, 40, 200, 60
Gripper_up_down	10, 0, 60, 10	Left_release	90, 20, 70, 25		
Gripper_sideway	10, 0, 60, 10	Right_grip	90, 20, 70, 30		
Inter_left	50, 20, 50, 10	Right_release	90, 20, 70, 30		
Inter_right	50, 20, 50, 10	Front_grip	90, 20, 70, 30		
Inter_front	50, 20, 50, 10	Front_release	90, 20, 70, 30		
Inter_back	50, 20, 50, 10	Back_grip	90, 20, 70, 30		
Position1	80, 30, 50, 30	Back_release	90, 20, 70, 30		
Position2	80, 30, 50, 30	Hand_over	120, 40, 60, 20		
Position3	80, 30, 50, 30	Hand_out	60, 20, 30, 10		
Position4	80, 30, 50, 30	Hand_in	60, 20, 30, 10		

benevolent, two were socially responsible, one was selfish and one selfless. In the case of the tests of the SMM model, each task needed to be allocated weights for social benefit, social cost, self-benefit and self-cost, as shown in Table 1. In general, these values are user-supplied to represent the nature of a task. For these tests, arbitrary values were allocated which were ignored by the other two models.

7.2. Results

Fig. 10 shows the number of tasks completed, from the set of 100, for the three models as a function of time. The data shown in Fig. 10 are an average over four rounds of tests, where the tasks were presented in a different random order for each round. For these tasks and conditions, the SMM model achieves an average task completion time of 5.0 s, compared with 6.03 s for the CPS model and 7.04 s for the Contract Net model. Not only is the task completion rate greatest for SMM, so too is the *scope*. All 100 tasks were completed under the SMM model, compared with 71 for contract nets and 84 for the CPS model.

A comparison was also undertaken between the CPS and SMM models to quantify the occurrence of 'wasted efforts', i.e. occasions when the initial selected goals are not achieved as the agents expected. The Contract Net model was not considered here as it does not include any uncertain attempts. Its mutual selection, based on two-way information transfer, can ensure that efforts are not wasted. In the SMM and CPS models some wasted efforts can be expected since both models include the notion of *attempt* in relation to the formation of a joint plan. The wasted efforts were measured by the number of processing cycles between an agent having an intended goal and returning the goal as an incomplete task. Fig. 11 shows that there is a decline in the accumulated wasted efforts across four rounds of tests

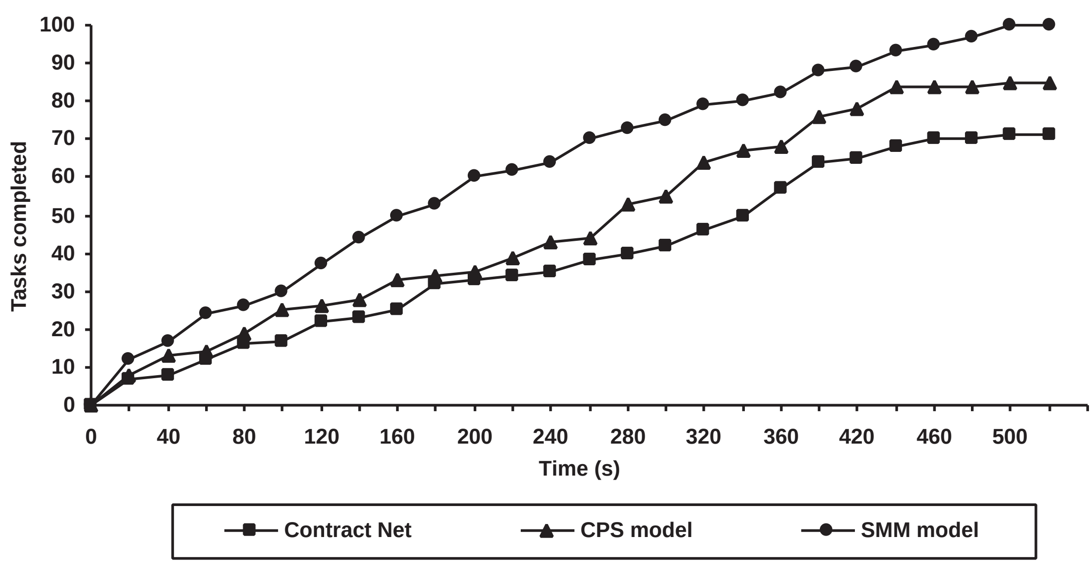

Fig. 10. Task completion, averaged over four test runs.

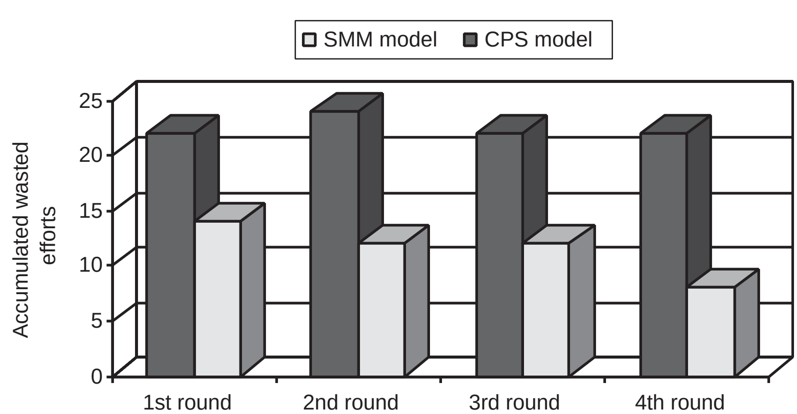

Fig. 11. Accumulated wasted efforts over four rounds of tests.

for the SMM. Two reasons for the decline can be identified. Firstly, mental adjustments are carried out by agents in the shifting stage of the SMM model, through which they learn from previous experience. Secondly, SMM cooperation takes place in a deliberate and predictable way because of the agents' decision-making behaviour.

8. Conclusions

ARBS has been shown to provide an effective means of implementing multi-agent systems. Disembodied agents have been used, whose components are distributed across knowledge sources and private partitions of the blackboard. Three models of multi-agent cooperation have been implemented and evaluated, namely Contract Nets, Cooperative Problem-Solving, and a new model, SMM. An objective comparison between the three models is difficult to achieve as they are each likely to be best suited to different types of problems and to be differently affected by the specifics of the ARBS implementation. Nevertheless, in a limited series of tests involving the control of two robot arms, the SMM model out-performed the other two in both its task completion rate and scope, i.e. the number of tasks it is capable of completing. Furthermore, where there is a high chance of a joint action running into difficulties, the SMM model wastes fewer resources than CPS. However, further investigation with more complex tasks and different implementations is needed before firm conclusions can be drawn about the relative performance of the three models.

As the SMM model requests many agents' mental deliberative behaviours rather than simple reactive behaviours, it might be argued that the SMM model is inappropriate in a time-critical environment. However, in the domain of cooperation between multiple robots, the agents' computation is much faster than the robots' physical movements. Typically, an atomic task takes a robot seconds to perform, while agent decision-making takes no more than a few milliseconds. Thus we propose that the SMM model is still appropriate because it enforces an agent's mental, rather than physical, behaviour. The SMM philosophy places the emphasis on reasoning before acting and, whenever a mistake is made, stopping actions sooner rather than later. Thus SMM provides an appropriate model for intelligent rather than reactive system behaviour.

Acknowledgements

Funding from the Open University Technology Faculty Research Board is gratefully acknowledged.

References

Bratman, M.E., 1987. Intentions, Plans, and Practical Reason. Harvard University Press, Cambridge, MA.

Cardarelli, G., Palumbo, M., Pelagagge, P.M., 1994. Critical aspects in multi-robot cooperation. Industrial Robot 21, 31–35.

Durfee, E.H., Lesser, V.R., Corkill, D.D., 1989. Trends in cooperative distributed problem solving. IEEE Transactions on Knowledge and Data Engineering 1, 63–83.

Hayes-Roth, B., Pfleger, K., Lalanda, P., Morignot, P., Balabanovic, M., 1995. A domain-specific software architecture for adaptive intelligent systems. IEEE Transactions on Software Engineering 21, 288–301.

Hopgood, A.A., 1994. Rule-based control of a telecommunications network using the blackboard model. Artificial Intelligence in Engineering 9, 29–38.

Hopgood, A.A., 2001. Intelligent Systems for Engineers and Scientists, 2nd Edition. CRC Press, Boca Raton, FL.

Hopgood, A.A., Woodcock, N., Hallam, N.J., Picton, P.D., 1993. Interpreting ultrasonic images using rules, algorithms and neural networks. European Journal of NDT 2, 135–149.

Hopgood, A.A., Phillips, H.J., Picton, P.D., Braithwaite, N.St.J., 1998. Fuzzy logic in a blackboard system for controlling plasma deposition processes. Artificial Intelligence in Engineering 12, 253–260.

Koivo, A.J., Bekey, G.A., 1988. Report of workshop on coordinated multiple robot manipulators: planning, control and application. IEEE Journal of Robotics and Automation 4, 91–93.

Li, G., Weller, M.J., Hopgood, A.A., 1997. Shifting matrix management—a framework for multi-agent cooperation. Proceedings of the Second International Conference on Practical Applications of Intelligent Agents and Multi-agents (PAAM97), London, pp. 353–364.

Mintzberg, H., 1979. The Structuring of Organizations. Prentice-Hall, Englewood Cliffs, NJ.

Nii, H.P., 1986. Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures. AI Magazine 7, 38–53.

Paljug, E., Yun, X., 1993. Experimental results of two robot arms manipulating large objects. IEEE International Conference on Robotics and Automation 1, 517–522.

Smith, R.G., Davis, R., 1978. Applications of the Contract Net framework: distributed sensing. Proceedings of the ARPA Distributed Sensor Net Symposium, Carnegie Mellon University, Pittsburgh, PA, pp. 12–20.

Smith, R.G., Davis, R., 1981. Frameworks for cooperation in distributed problem solving. IEEE Transactions on Systems, Man and Cybernetics 11, 61–70.

Wong, K.C.P., Nolle, L., Hopgood, A.A., 2001. Development of a distributed knowledge-based system. International Conference on Artificial Intelligence (IC-AI'01), Vol. III, Las Vegas, USA, pp. 1219–1224.

Wooldridge, M.J., Jennings, N.R., 1994a. Formalising the Cooperative Problem Solving process. Proceedings of the 13th International Workshop on Distributed Artificial Intelligence (IWDAI94), Lake Quinalt, WA, pp. 403–417.

Wooldridge, M.J., Jennings, N.R., 1994b. Towards a theory of Cooperative Problem Solving. Proceedings of the Sixth European Conference on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW94), Odense, Denmark, pp. 15–26.

Gangmin Li is a research fellow in the Knowledge Media Laboratory of the Open University. He was awarded a Ph.D. from the Open University in 1999 for his work on the design and implementation of a framework for agent cooperation.

Adrian Hopgood is Professor of Computing and Head of the School of Computing and Mathematics at the Nottingham Trent University. He has a Ph.D. from the University of Oxford and an MBA from the Open University. He is a visiting professor at the Open University, where he was a senior lecturer until December 2000.

Martin Weller is a senior lecturer in the Institute of Educational Technology at the Open University. His research interests include intelligent agents, e-commerce, web education and the implications of the Internet for universities. He has a Ph.D. in applied artificial intelligence from the University of Teesside.