PLANNING AND BEHAVIOURS - A HYBRID ARCHITECTURE FOR MOBILE ROBOTS

R.S.Aylett,[1] A.Coddington,[1] Dr D.P.Barnes.[2] R.A.Ghanea-Hercock[2], Prof. J.O.Gray[2]

[1]IT Institute, University of Salford, [2]EEE Department, University of Salford

Abstract: The paper discusses a hybrid predicitve-behavioural architecture being developed by the Mobile Robotics Research Group at Salford University for cooperating mobile robots. The test bed used for experiments is described, the multi-agent framework for the hybrid architecture is explained and the behavioural and predictive components are presented. Issues of division of responsibility, communication nd the relationship between the two styles of architecture are described.

Keywords: Mobile Robots, Hybrid Vehicles, Intelligent Machines, Tasks, Co-operation, Artificial Intelligence

1. INTRODUCTION

There are two main architectural approaches to building advanced mobile robots for the carrying out of complex tasks. The more traditional architecture, going back to SHAKEY (Fikes et al 1972), is essentially hierarchical and model-based, with action deriving from predictive planning. This style of planning projects actions into the future in which they will be executed by using a symbolic model of the world.

The disadvantages of this approach have been well-documented in the last five years or so: high resource demand, fragility and inflexibility. Planning can never be better than the world model on which it is based, and the creation and updating of such models is computationally expensive and fraught with inaccuracy and ambiguity.

As a result of these criticisms, a number of alternative approaches were produced, with behavioural architectures proving the most influential in robotics. Of these Brooks' (1986) subsumption architecture is the best known, though it is far from the only one. Behavioural architectures form a strong contrast to the earlier approach.

Advocates of such architectures point out that their horizontal layers of behaviour, in each of which close coupling of sensors and actuators occurs, allow each to work without a large processing overhead and to react very quickly to incoming data. The reactivity of systems using this approach makes them far more robust and adaptive than was possible using a more traditional architecture. A world model is not required in order to act, so that the enormous processing overhead disappears, the problems of dealing with sensor accuracy and dynamic updating are much reduced as are the engineering problems of producing mobile robots which actually work.

Work by David Barnes and colleagues at Salford over the past few years (Barnes and Gray 1991) has resulted in the development of the Behaviour Synthesis Architecture (BSA) which has been applied to the problem of producing cooperation between mobile robots as they jointly transport an object from one location to another. The BSA has demonstrated that the behavioural approach can indeed bring the advantages claimed for it and that cooperation need not be based on internal models and symbolic processing.

However, behavioural architectures in their turn have their weaknesses. The purely local decision making employed may result in grossly inefficient actions or in the worst case in total failure to solve a problem. In addition, the problem of planning complexity is simply removed to the human designer of the behaviour repertoire who must hardwire in the correct reflexes. Finally, and most seriously for practical applications in industrial contexts, behavioural systems provide a completely opaque interface to a human who might be responsible for the goals they are to achieve.

In a sense behavioural agents may be *too* autonomous in many situations: if they fit human purposes then it is only because they are designed to do so *a priori*, not because there is any easy way of communicating such purposes to them. Yet even simple robots represent a substantial investment, so that task flexibility is likely to be frequently required. Such flexibility can only be achieved if there is a way of communicating tasks.

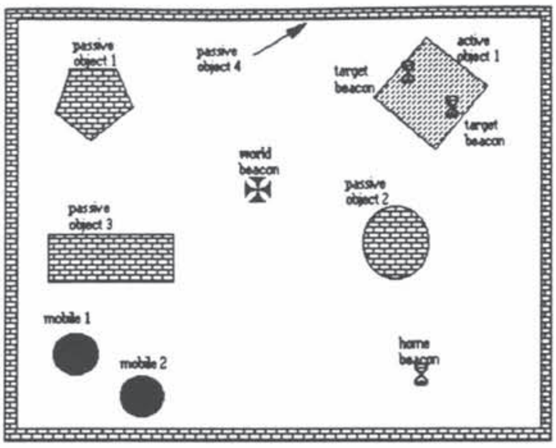

Figure 1. plan view of the robot test bed

It is for these reasons that a hybrid architecture, in which both predictive and behavioural components are combined, becomes an attractive proposition and is being investigated by a number of groups (Bonasso 1993, Gat 1992, Zelinsky et al 1994). This paper describes work currently being carried out in Salford to combine the already developed BSA with a predictive planning system.

2. THE ROBOT TEST BED

The Salford Mobile Robots Research Group has focused upon the development of multiple autonomous devices for complex task achievement with direct industrial applications. The particular task explored to date is the relocation of objects in semi-structured environments (Eustace et al 1993).

Two B12 robots have been used for this work, and Figure 1 shows a plan view of the environment in which they work. The passive objects in this diagram represent obstacles which must be avoided, the active object represents a loading station from which the two robots are to fetch an object (equivalent to a pallet in an industrial context) which is to be relocated at the home beacon. Figure 2 shows a 3D version of the environment in which the robots are relocating the object.

The task imposes the following requirements on the robots:

- independent travel through the environment
- obstacle avoidance
- beacon/obstacle location
- object acquisition by both robots in coordinated fashion
- cooperative relocation while avoiding obstacles

Object acquisition is managed by two robots docking with the loading station. When the station senses that both robots are present, it will lower the object onto the top of the robots. This top surface is equipped with spring-loaded capture heads which allow the robots to support the object.

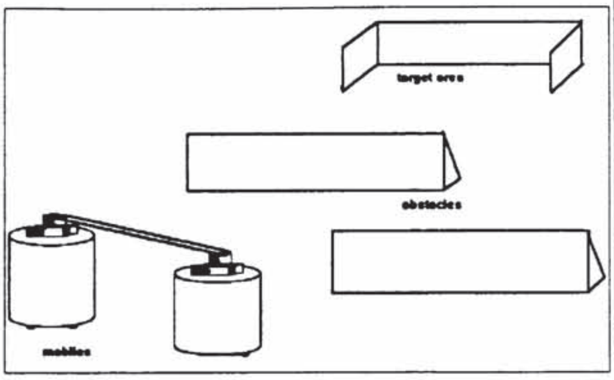

Figure 2. 3D view of the relocation task

3. A MULTI-AGENT FRAMEWORK

In a hybrid architecture there are a number of key problems: the extent to which the separate components are integrated, how they communicate, and the relationship between predictive and reactive phases over time as tasks are carried out.

The hybrid architecture which has been developed (Aylett et al 1993) makes use of the AI multi-agent paradigm but takes a different approach in applying this idea from most purely software based multi-agent systems. Here, agents have an homogenous architecture, equivalent capabilities and the ability to communicate with each other at any time. In Salford it was decided to specialise agents into a system containing one reflective agent, endowed with the predictive planning capability of the system, and a number of behavioural agents running the BSA. The reflective agent can communicate with the behavioural agents only intermittently.

This gives a semi-hierarchical structure (Ephrati and Rosenschein 1992): hierarchical in that the reflective agent acts as an interface for a human operator organising robot missions, but semi- because behavioural agents have a good deal of local autonomy in executing such missions. This supervisory relationship maps well onto widely-used forms of industrial organisation.

In principle the division of responsibility, which is a key research-issue in this type of hybrid system, is related to an abstraction hierarchy for the domain. Aspects of the domain which are relatively stable and large-scale (for example doors and walls in an indoor environment) can be modelled in the classic manner, while the more dynamic and uncertain features of the environment - transient obstacles, other agents, humans - are not modelled at all but left to the behavioural capacities of the mobile robots themselves to deal with. This produces a task hierarchy since only those tasks relating to the more abstract elements of the domain can be planned. The basic competencies of the behavioural agents define the primitive level of the planner.

For example, the two robots in the test bed described above maintain no spatial model, but

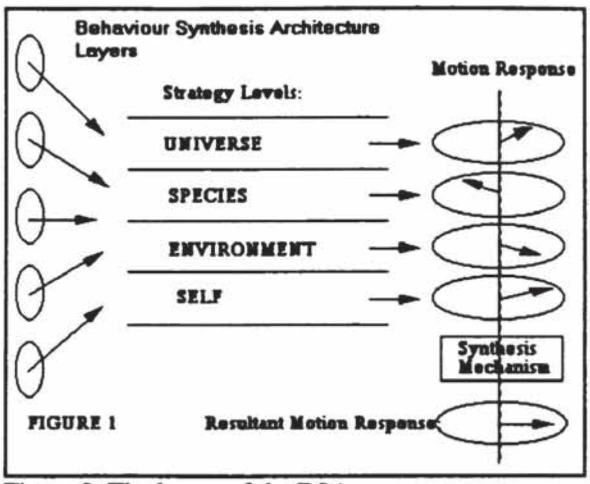

Figure 3. The layers of the BSA

navigate with reference to beacons specified in their behaviour scripts (see 4.2 below). For these robots therefore, high-level route planning is necessary within the planner, so that the sequence of beacons to look for may be specified. On the other hand, moving between beacons is a concern of the BSA and not a problem for the planner at all. Likewise cooperation between the robots is incorporated at an abstract level in the planning of the reflective agent through planning decisions about how many robots to allocate to a task, but the behavioural agents actually cooperate without needing to know that this is what they are doing.

4. INTER-AGENT COMMUNICATION

Of course dividing responsibilities between agents is no use unless communication can be established between them. On the one hand, there must be enough communication between behavioural agents to allow them to act cooperatively. How this is achieved will be discussed in 4.1 following. On the other hand, the reflective agent must be able to communicate with behavioural agents if the user's task is to be carried out, and the behavioural agents must also communicate with the reflective one at the very least to indicate success or failure in that task. The mechanism supporting this communication, the Behaviour Script, will be discussed in 4.2.

4.1 The Behaviour Synthesis Architecture (BSA)

Work at Salford has resulted in the development of the Behavioural Synthesis Architecture (BSA). This is described in more detail elsewhere (Barnes and Gray 1991), but we summarise it here as follows. A number of behaviour patterns are active concurrently in a robot, each mapping an input stimulus onto an output motion response. Unlike the Brooks approach, in which only one behaviour pattern has control of the actuators at any one moment, in the BSA, as the name suggests, the different motion responses are synthesised together to produce a resultant motion, as shown in Figure 3.

However, apart from the stimulus-response mapping, each behaviour pattern also contains a stimulus-utility mapping, which for a given stimulus measures the importance of that behaviour pattern's response. For example, as a robot gets close to an obstacle, the importance of the slow-down response being generated by the obstacle-avoidance behaviour rises (Figure 4). The resultant response will therefore be increasingly affected by the output from this behaviour pattern compared to others active at the same time.

This may be expressed as follows:

$$bp_t = \{r_t = f_r(s)\}\ \{u_t = f_u(s)\} \qquad (1)$$

where r_t is the motion response at time t and this is a given function, f_r, of a given sensory stimulus, s. Associated with it is its utility, u_t, which is a function, f_u, of the same sensory stimulus. The values of r_t and u_t together form a vector known as a *utilitor*.

As Figure 3 shows, behaviours may be categorised into different levels. An example of a *self* level behaviour would be battery recharging, of *environment* level behaviour obstacle avoidance, of *species* level behaviour cooperation with another robot, and of *universe* level behaviour navigating to a predetermined destination.

An example of cooperative behaviour has been demonstrated at Salford in the task described in section 2 above. Each B12 robot has a spring-loaded capture head on top with sensors able to measure both the displacement of the capture head relative to its central position, and the velocity with which it is moving. If an object is placed on top of two robots' capture heads, behaviour patterns using these sensors are able to affect the resultant motion of the two robots such that they travel as a pair (or *dyad*). Disparity in velocity translates into capture head displacement which feeds back into a velocity correction.

This is a good example of communication between behavioural agents in the hybrid architecture. Such communication need not take any symbolic form or contain explicit information, but can be treated in the same way as any other sensor input. The behavioural agents communicate without knowing

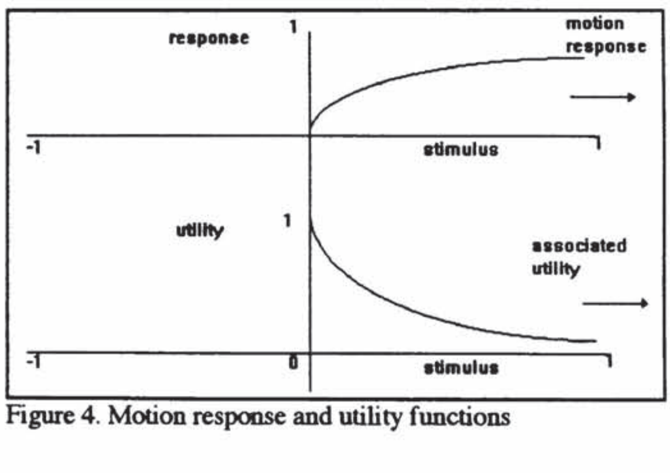

Figure 4. Motion response and utility functions

that is what they are doing and cooperate without knowing of each others' existence.

4.2 Behaviour Scripts

Different issues are involved in allowing a reflective agent to communicate with behavioural agents. A symbolic-numeric translation must take place, since the reflective agent is carrying out symbolic modelling and planning, while the behavioural agents are reacting to digitised sensor data. The task structure developed by the reflective agent planner must also be communicated. Both these issues are addressed by a component of the BSA known as a behaviour script.

It was realised early in the development of the BSA that major problems attached to allowing all behaviour patterns to be active all the time. To give the most obvious example: if obstacle avoidance behaviour was always active, a robot could never dock with a loading station as described in section 2 above. Obstacle avoidance would prevent it ever getting that close.

Associated therefore with every behaviour pattern within a robot is an 'active' flag which enables or disables a particular pattern. Thus obstacle avoidance, for example, can be turned off when required. A behaviour script is a way of organisaing the activation of behaviour patterns appropriate to the particular sub-task being carried out, and the deactivation of behaviour patterns not appropriate for this subtask.

For example, the following behaviour patterns have been developed for the task discussed in section 2 above:

bp1 ensures a steady translate acceleration.

bp2 decelerates the mobile when an object is detected to be within a given range.

bp3 causes the mobile to rotate away from an object when detected to be within a given range.

bp4 ensures the mobile translates to an external beacon destination in the shortest possible time, i.e. travels at a maximum given velocity.

bp5 orientates the mobile so that it is facing towards an external beacon destination point.

bp6 maintains a zero capture head velocity.

bp7 ensures that the position of the capture head remains central to its location on top of a mobile.

In order to achieve proximity to the loading station, **bp1, bp2, bp3, bp4** and **bp5** are required. However the robot could not dock with **bp2** and **bp3** active as these would lead it to treat the loading station as if it were an obstacle. The behaviour script consists of a set of triplets: {sensor precondition(s), behaviour packet, sensor post condition(s)}, where each behaviour packet lists the behaviour patterns to be flagged as active. Thus the behaviour script triplet for docking treats 'distance < tolerance' as a sensor precondition, lists **bp4** and **bp5** in its behaviour packet, and 'object loaded onto capture head' as its sensor post-condition. As each segment of the behaviour script is carried out, the pre-condition for the next is encountered so that the whole script is finally executed. Figure 5 shows the overall process of executing a behaviour script.

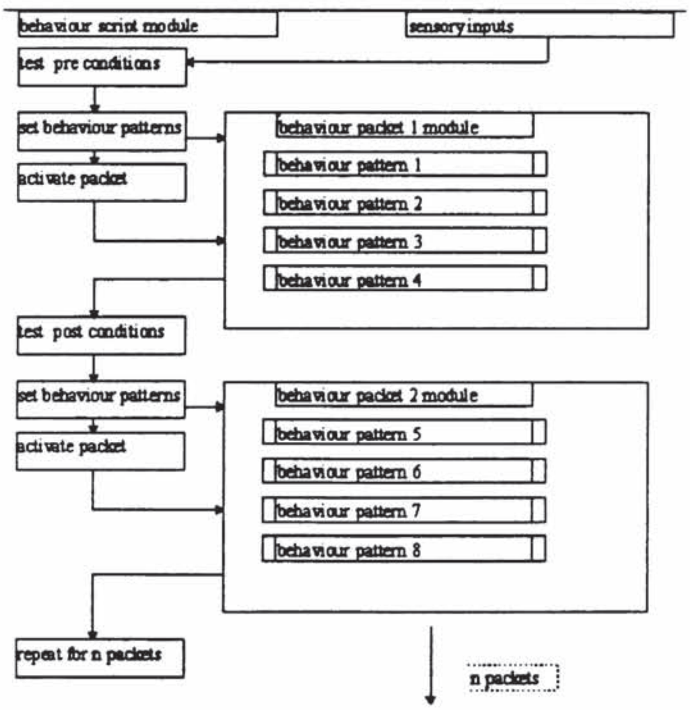

Figure 5. Behaviour script execution

One of the significant aspects of behaviour scripts is that they sequence behaviour patterns into subtask-achieving packets. A behaviour script therefore reflects the subtask structure of an overall task and may therefore act as a communication mechanism between a reflective agent determining this subtask structure and the behavioural agents which maust carry them out.

5. THE REFLECTIVE AGENT

The tasks to be performed by the reflective agent should now be clear. It is required to take a user goal, decompose this into subtasks, allocate them to the appropriate number of mobile robots, and translate the subtasks into a behaviour script for each such mobile. The collection of behaviour scripts relating to a particular user goal is known as a *mission*.

Figure 6 shows the internal architecture of the reflective agent. The planner and its knowledge base are much as would be found in any predictive planning system, however the behaviour library and the agents knowledge base are particular to the hybrid architecture. The former contains a list of all behaviour packets known to the system, together with *symbolic* versions of the sesor pre- andpost-conditons discussed in 4.2 above. This library is used to convert the network of primitive planning actions generated by the planner into behaviour script triplets.

The agent knowledge base contains a descritpion of all behavioural agents known to the system. Each agent entry references the behaviour packets that agent is capable of carrying out (not all agents may have identical abilities) as well as the communications routines used to contact the agent. This last information is required because the reflective agent may organise more than one mission concurrently, and this is being explored in the current project by driving both the two B12 robots in the Robot Test Bed and simulated robots in virtual worlds. Physically, communication with the two B12 robots takes place via an RF link. However, communication with the project's simulator is via C function calls.

The mission organiser component is responsible for transforming the sub task netwrok from the planner into behaviour scripts and in communicating those to appropriate robots. It will also spawn a mission monitor for each mission which will await a success/fail message from the participating robots.

In conventional predictive planning systems, such a monitor checks that the preconditions of each planned action are true before it is executed and then that the expected effects have in fact occurred. The reflective agent however, cannot assume that it is possible to communicate with behavioural agents with that kind of frequency and will usually only hear at the end of a mission that all the actions in it have been successfully completed.

Likewise, if there is a problem, this may occur some way thorough a mission and a number of issues are involved in finding out the status of other robots involved in the same mission and attempting to replan. This problem is closely linked of course with giving the behavioural agents the ability to detect failure, and is a major thread of the research.

Because the behavioural robots have local initiative, and because the reflective agent's world model is only correct at some abstract level, it is not possible for the reflective agent to make a very accurate assessment of how long a mission should take to complete. However, its mission monitor component is given a time allocation for the mission adjusted upwards for uncertainty, and this is used as a time-out mechanism to trigger attempts to contact the behavioural agents in a mission if no message has been received within this time.

6. DEALING WITH FAILURE

With the BSA, and similar behaviour based architectures (Arkin 92,Brooks 86), the growth in interactions between behavioural components or patterns may rapidly exceed the scope of manual design when producing a system for a complex task (Harvey 93) This problem is manifest in the BSA where the effect of conflicting utility functions becomes increasingly difficult to predict, (particularly if the functions are non-linear). This leads to the requirement for on-line modification of the parameters, such as relative utility and response, for each behaviour. One possible solution to this is via learning mechanisms which may be based on neural net architectures (Maes 92). The direction of this work however was to create a flexible but easily controlled architecture with limited computer processing resources on the robot, as this can be transferred onto small desktop mobiles suitable for researching large group dynamics, such as flocking. In addition the result of having multiple co-operating robots makes the task of assigning a reward function to reinforce learning considerably harder than for single agent systems. Thus the guiding principles have been described by Maes (1994)as:

- Looking at complete systems changes the problems often in a favourable way.
- Interaction dynamics can lead to emergent complexity.

6.1 Coupled Behaviours

The approach which is being currently researched is based on the principle that in most higher organisms their behaviours are an integrated set of stimulus-response functions. Behaviours are run in parallel and have specific sensory inputs, however the overall response of the organism may require the modification of each behaviours importance with respect to that of the other behaviours, i.e. the total response to a given set of sensory inputs is a complex set of interactions (Steeles 94).

It is therefore necessary to dynamically couple the separate behaviours within the BSA so that they act in a coordinated way. This has been implemented by allowing limited coupling between related behaviours, where the relationship may be similar actuation output or behaviours connected by level of coordination, i.e. individual or group activity. In order to study this process a simulator system, which was created for the original project (Eustace 1993) has been used to determine which behaviours may be coupled and at what point the system becomes unstable. Clearly as the degree of coupling increases, stability and predictability may be difficult to control. The limits of the method are part of current research and these issues will be presented in future literature. Preliminary results indicate that for a limited degree of coupling the pair of co-operating robots exhibit a significantly improved response as they negotiate a difficult environment.

7. CONCLUSIONS

This paper has presented a hybrid architecture in which a multi-agent framework is used to combine a reflective agent using classical predictive planning with cooperating mobile robots using the Salford BSA. The user is presented with a clear interface and the user goal is automatically decomposed and allocated to the mobiles. The flexibility and resilience of the behavioural architecture allows the mobiles to operate in a semi-structured environment which may differ from the abstract model used by the planner. Thus some of the benefits of each approach have been combined.

REFERENCES

Arkin, R.C. (1992) Co-operation Without Communication: Multiagent Schema Based Robot Navigation, *Journal of Robotic Systems* v9 n3, pp 351-364

Aylett, R.S. & Eustace, D. (1993) Multiple Cooperating Robots - Combining Planning and Behaviours. Proceedings, CKBS-SIG, Ed: S.M.Deen, DAKE Centre, Keele, 1993, pp3-12

Barnes, D.P. and Gray, J.O.(1991) "Behaviour Synthesis for Co-operant Mobile Robot Control", Proc. IEE Int. Conference on Control 91, Vol. 2, pp 1135 - 1140.

Bonasso, R.D. (1993) Integrating Reaction Plans and Layered Competences Through Synchronous Control *Robotica*, pp1225-1231

Brooks, R.A., 1986, " A Robust Layered Control system for a Mobile Robot", IEEE Journal of Robotics and Automation. RA-2, No.1, pp14-23.

Ephrati, E. & Rosenschein, J.S. (1992) Constrained Intelligent Action: Planning Under the Influence of a Master Agent. Proceedings, 10th National Conference on Artificial Intelligence, AAAI 92, pp263-268.

Eustace, D., Barnes, D.P., Gray, J.O., 1993, "Co-operant Mobile Robots for Industrial Applications", 19th Conf. IEEE Industrial Electronics Society, Hawaii

Fikes, R.E; Hart, P.E. & Nilsson, N.J. (1972) Learning and Executing Generalised Robot Plans*Artificial Intelligence*, 3.

Gat, E. (1992) Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture for Controlling Real-World Mobile Robots. Proceedings, 10th National Conference on Artificial Intelligence, AAAI-92, pp809-815

Harvey, I., Husbands, P. and Cliff, D., 1993, "Issues in Evolutionary Robotics", Proc. 2nd Int Conf. on Simulatiom and Adaptive Behaviour, Meyer, J., Roitblat, H. and Wilson, S. (Eds), MIT Press. pp 364-373.

Maes, P. 1992, "Learning behaviour networks from experience", Towards a practice of autonomous stsrems, Proceedings of the First European Conference on Artificial Life., F. J. Varela & P. Bourgine (Eds.), Cambridge, MA, MIT Press.

Maes, P., 1994, "Modeling Adaptive Autonomous Agents", Artificial Life Vol 1. MIT Press, Cambridge, pp 135-162.

Steeles, L., 1994, "The Artificial Life Roots of Artificial Intelligence", Artificail Life Vol. 1 MIT Press, Cambridge, pp 75-110.

Zelinsky, A; Kuniyoshi, Y. & Tsunkune, H. (1994) Monitoring and Coordinating Behaviours for Purposive Robot Navigation. Procs, IROS '94

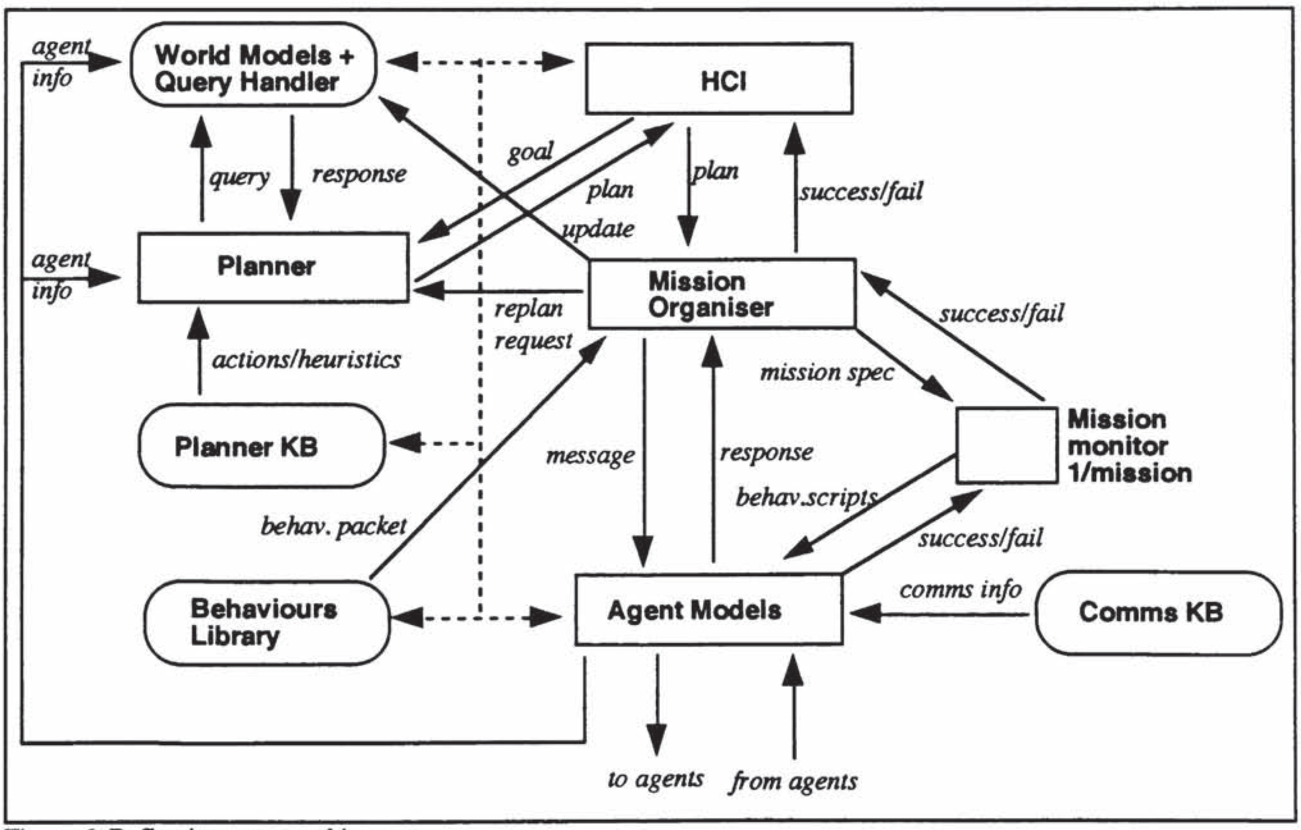

Figure 6. Reflective agent architecture.