The SOMASS System: a Hybrid Symbolic and Behaviour-based System to Plan and Execute Assemblies by Robot

Chris Malcolm

Department of Artificial Intelligence, University of Edinburgh, 5 Forrest Hill, Edinburgh EH1 2QL, Scotland, UK

ABSTRACT

The SOMASS system consists of a classical symbolic hierarchical planner which plans assemblies of SOMA parts to be performed by a robot controlled by a behaviour-based plan execution agent. SOMA parts consist of simple bent and forked bricks which simplify the geometry while preserving the shape-dependent part-fitting and part-ordering characteristics of the assembly problem. The hybrid architecture both simplifies the planner and ensures reliable execution of the plans. The behaviour-based architecture of the assembly agent is based on the general principles underlying Brooks's subsumption architecture but is also specifically designed to support idealised symbolic planning. It is argued that this architecture securely grounds the symbolic terms in which the planner operates by tying them to the goals of the behavioural modules of the agent. In this way it avoids some of the criticisms addressed by Dreyfus and Searle to AI.

1 Introduction

The successful development of an experimental fully automated hybrid assembly system is described. The system is fully automated in the sense that the assembly is specified at the task level, a planning system plans the assembly, and an execution system executes the plan. It is hybrid because it is composed of a) a *symbolic* planner, written in PROLOG; and b) a *behaviour-based* plan execution agent. The system is experimental because it was developed to see if this hybrid architecture could simplify some of the problems of fully automated assembly systems. The development was successful because it did simplify them, and now forms the basis of the assembly robotics research programme at Edinburgh.

2 The Motivating Problems

Since the topic of this paper is the virtues of a particular hybrid architecture, and not assembly robotics, a simplified summary of assembly robotics will set the context.

Between about 1970 and 1986 Edinburgh University and other robot research laboratories were pursuing research programmes into the development of higher level assembly systems. In industrial mass production assembly is often done by dedicated machinery. When a new product arrives, new assembly machinery has to be built. If assembly were done by robots, then instead of having to build new assembly hardware, only reprogramming the robots would be needed. Since software is easier to change than hardware, this would save money.

It became clear that developing a robot program which would enable a robot to construct a typical industrial assembly reliably, despite typical uncertainties on part dimensions, locations, etc., was so difficult that many manufacturers decided that the old ways were better, and threw dust sheets over their robots. This could be seen as a programmer productivity problem, and the computer industry already had a successful paradigm for tackling programmer productivity problems: raise the level of the programming language. From the earliest days the advantages of raising the level of assembly robot programming had been recognised by robotics research laboratories. In mass-produced assemblies the shapes and dimensions of the parts, their tolerances, and how they were supposed to fit together, were all known. Would it be possible to deduce from this existing knowledge the movements with which a robot could accomplish the assembly?

Edinburgh's high-level RAPT robot programming language went some way towards this goal [25, 1, 26]. It permitted the movements of the robot to be specified in terms of relationships between geometric features of the parts to be fitted together. Knowing the size, shape, and location of the various parts, and having sophisticated geometric reasoning capabilities, the RAPT system parameterised the required movements in terms of the robot's spatial co-ordinate system.

While in a perfect world it would be possible to do assemblies reliably by pre-calculated moves without sensory feedback, in the real imperfect and uncertain world it is sometimes necessary to use sensing. Deciding how and when to use sensing is a complex technical matter potentially involving all that is known about the assembly. Solving this kind of problem involves search in a problem space, in this case by a planner. This is bad news, since the search space has definite combinatorially explosive tendencies and one has to be careful to avoid manifestations of the frame problem [29]. Planning itself is a peculiarly awkward kind of search, e.g., as Chapman has shown, many restricted planning problems are intractable [6].

Assembly planning is particularly awkward, because at almost every stage there can be a crucial dependence on a detailed geometric relationship which is complex to calculate and which can't be worked out until many other details have been settled. As a consequence there is no good place to start planning an assembly. Wherever one begins, tripping over some small detail in the final stages of planning may force a fundamental backtrack to the earliest decisions.

Edinburgh's attempts to automate some of these considerations of sensing and uncertainty foundered on the combinatorial explosion and consequent intractability [12]. Similar attempts in other research laboratories suffered similar fates. In particular, it was for this kind of reason that Brooks at MIT decided to part company with the assembly research programme on which he had been collaborating with Lozano-Perez [18]. Brooks declared that so many fundamental problems of robotics were being elided by the "classical" approach that it was necessary to retreat to investigating simple behavioural competences in simple mobile robots, and, via his new behaviour-based subsumption architecture, to try to recapitulate (reverse-engineer) the progress of biological evolution from the simplest mobile creatures to the eventual goal of intelligent behaviour [5].

Formal logic and the development of the general purpose computer are founded on the idea that human thinking is based on reasoning about propositionally represented knowledge. Brooks maintains that this is a rational reconstruction and not an ingredient in the implementation; in other words it is a behavioural description, not a generative mechanism. The avoidance of knowledge representation and reasoning of this kind is an important general principle behind his subsumption architecture.

Assembly robots are intended to perform a complex detailed pre-specified task, and to be easily commanded to perform new ones. It is hard to see how they can do this without a considerable amount of knowledge. At the very least they need a plan of some kind. Since a plan is exactly the sort of thing Brooks wanted to avoid, assembly robots were not (for Brooks) a fruitful domain in which to conduct this new behaviour-based research programme into robot architectures.

3 Behaviour-based Assembly Robots

At Edinburgh a different view was taken. While agreeing with Brooks's general criticisms of AI, the outright condemnation of knowledge representation was felt to be going too far. RAPT for example had suffered severe problems when we attempted to bring in considerations of sensing and uncertainty. Because it had seemed at the time the obvious way to go we had decided to bring in sensing and uncertainty at the same level in the system at which geometric relationships were reasoned about. In fact "decided" is too strong a word, since this was not a conscious decision. The paradigm we were implicitly following claimed that intelligent behaviour was the result of knowing the right facts and reasoning about them. Since sensing and uncertainty were inextricably involved with the geometry of the assembly the decision to integrate them at this level was so obvious as to be trivial.

3.1 *Dividing the Problem between Planner and Agent*

In fact so obviously right is the classical approach from within its own terms that it is hard to argue against it in those terms. Let us therefore take another view. Consider the general problem of making a plan for an agent to perform. Suppose you were an averagely stupid mobile robot and I had to supply you with a reliable plan of how to get to the railway station. Considering all the obstacles which hazard could erect in the way of a stupid robot wishing to catch a train this plan would have to be very complex. In fact, there probably does not exist today any mobile robot which anyone could write a reliable station-finding program for in less than five years.

On the other hand, supposing you were an averagely intelligent human being, I could supply you with a reliable station-finding plan on the back of an envelope in five minutes.

In the first case we have a very simple agent, which requires a very complex plan, produced by a very complex planner. In the second case we have a very complex agent, which requires a very simple plan, produced by a very simple planner. Clearly in solving this general problem of doing a commanded task we can choose to put the complexity either in the agent or in the planner. If we add the complexity of the agent to the complexity of the planner to get the total system complexity, an interesting question arises. What is the optimum division of labour between planner and agent which gives the minimum total system complexity?

Now the classical knowledge-based approach to assembly robot programming presumes that however one decides to build one's higher level planners and compilers, in the end they will produce a robot program in terms of motions in space, i.e., a robot program written in one of today's robot programming languages. That is not the simplest possible agent. For example, we could decide that the robot shouldn't need to know how to move itself in straight lines, and make the planner work out how to do this. This would make the robot program much bigger and more complicated to produce and greatly increase the knowledge and reasoning required by the planner/programmer.

So what makes the current assembly robot, programmed in terms of motions in Cartesian space, the optimum assembly agent? Given the knowledge-based assumption of classical AI, the answer is quite simply Cartesian Geometry itself, since that hooks directly into the Newtonian world of shape, mass, motion, force sensing, and camera views, in which our scientific understanding of the assembly process exists. There is no doubt that this is the basis of a sound formal description of the process of assembly. Is it necessarily the case, however, that using a description of a process as an ingredient in its implementation is the best way of doing it?

3.2 *The Question of Implementing the Description*

It is possible to build a ball-catching robot which depends on a Newtonian understanding of the problem. There is no doubt that this can be done with today's technology, probably by a good student with suitable equipment and simplification (white ball against a black cloth, etc.) in a year or few, e.g. [2]. We know that we do not catch balls like this, in the sense that our brains don't parameterise and solve the appropriate Newtonian equations. On the other hand, given the mathematical capabilities of the modern computer, it may well be computationally easier to implement *robot* ball-catching in this way than to mimic the way the human brain does it.

Now consider the sine waves which can be observed in the movement of the legs of a walking millipede. If you wanted to build a robot millipede which walked like this you could program a sine wave generator to specify the leg positions, and then control each leg to follow the commanded motion. This would be a computationally very wasteful way of doing it, since the natural dynamics of leg movement, with appropriate inter-leg delays, will naturally produce this sine wave effect with orders of magnitude less computation. In other words, while using a formal description of a behaviour as an ingredient in implementing the behaviour is obviously *one* way of implementing the behaviour, it is not necessarily the *only* way, nor necessarily the best.

Brooks's claim is that while knowledge representation can provide the basis of a description of intelligent behaviour, it is silly to try to use it as an ingredient in the implementation. While we agree with this as a general criticism of the over-ambitious claims of knowledge-based AI, we do not agree that it is true in every case. For example, it may well be the case that the best chess-playing programs of today, while they do not play chess like people, will nevertheless be the most computationally efficient way for computers to beat people at chess. Note too that the seductive popularity of the knowledge-based approach to AI stems from the fact that in some domains we (human beings) have found a formal knowledge-based approach to be invaluable in extending our own natural and informal skills.

On the one hand, if we accept for the sake of argument that the brains of dumb animals do not use the knowledge-based approach of classical AI, the degree of commandable intelligent behaviour which sheepdogs or logging elephants (for example) routinely display shows us how much can be accomplished outside the knowledge-based paradigm. On the other hand, human beings, who possess the best brains we know of, do find the knowledge-based approach invaluable in certain domains. Note in particular that human beings are the only animals capable of complex assembly, and that in complex assembly tasks we routinely make use of formal knowledge (diagrams, notes, measurements, etc.).

All this suggests that for certain kinds of tasks there may be a profitable marriage to be had between knowledge-based systems and others; and that assembly is likely to be such a domain. Consequently if one likes the general tenor of Brooks's criticisms of classical AI, yet feels that he goes too far in outright damnation of knowledge-based systems, then assembly robotics may well be a good experimental domain in which to explore this. Note too that assembly robotics is concerned with a relatively well-ordered and predictable world in which very complex tasks must be accomplished, as compared to the less predictable world of simple mobile robots, in which they have to accomplish relatively simple tasks, while being alert for emergencies. Therefore the high demands for parallelism and timely response which Brooks' subsumption architecture must face in mobile robots, and which has led to such suggested enhancements as REX [16], are not problems which trouble the assembly robot. These problems which most naturally concern reactive mobile systems and reactive planners are not so pressing in assembly robotics. Here the key problems are the architectural problems of symbol grounding and representation, as elucidated by Harnad [15], and Agre and Chapman's indexical-functional representation a key idea [7].

3.3 *Classical and Behaviour-Based Hybridisation*

Robotic assembly provides a very clean break between planner and agent. In the simplest system there is no co-operation between the two parts of the hybrid system. The planner makes a plan of how the agent is to do the assembly. The agent then performs the plan. The interface between the two parts of the system is simply a file: the plan. The most elementary kind of plan is a program. More complex plans include (perhaps implicitly) information about "why" as well as "how", which enables the agent to use them in an advisory rather than a prescriptive fashion.

Using an existing assembly robot as a basis, the obvious experiment was to construct the most capable behaviour-based assembly system possible and to devise a planner capable of making plans for it to do assemblies. The simplest possible system would be sensorless, but designed to permit the simple introduction of sensing.

Now assembly planning depends intricately on the details of geometric shape. CAD systems capable of handling the geometry of the parts of industrial assemblies are very large complex programs. It would therefore be wise to choose an experimental domain which contained the important features of assembly, but was geometrically as simple as possible.

3.4 *The Experimental Domain*

What are the important features of assembly? The parts fit together in ways which are dependent on their shapes, and the trajectories and orderings of their fittings together are dependent on their shapes.

We chose the SOMA puzzle. Here the part fitting depends on the shape, and the ordering of the part fittings depends on considerations of both shape and gravitational stability of the partially complete assembly. Invented by the Danish mathematician Piet Hin [13], this consists of the parts which can be constructed by face-to-face conjoining four or fewer equal cubes in such a way as to form different irregularly shaped parts. It turns out that there is only one part with three cubes, and only six with four cubes. Since 6x4=24, 24+3=27, and 27 is a perfect cube (3x3x3) this raised the question of the possibility of making a 3x3x3 cube from these parts. In fact it is possible to make a cube from these parts in 240 different ways [4]. Since the assembly problem becomes quite different if you perform it a different way up (due to gravity), and a cube has six ways up, then from the point of view of an assembly robot there were clearly 6x240=1440 different ways of doing this assembly, each in effect a different plan of a different assembly.

Thus the SOMA cube has the basic characteristics of assembly, and is capable of providing a good deal of exercise for both an assembly planner and an assembly agent.

3.5 *Development of the Agent*

Since the capabilities of the agent would define the terms in which the planner operated, the agent should clearly be developed first.

In line with Brooks's definition, we decided that a behaviour should reliably accomplish some useful task in the world in the presence of typical uncertainties. We decided to call the external behaviour a *behaviour*, and the collection of software and hardware implementing it a *behavioural module*. Since the first version of the system was to be without sensors, and therefore had no choice but to execute its plan in strict sequence at all levels, the behavioural modules would here be identical to subroutines, and the plan would therefore consist of a set of parameterised subroutine calls.[1]

The general idea was to try to make each subroutine cope with a larger amount of uncertainty than it would be presented with, while presenting to further modules less uncertainty. In a sense each behavioural module would act like a funnel, capturing its input from a wide range of uncertainty, and delivering it to subsequent modules with lessened uncertainty. These modules should be parameterised in such a way as to make them as general as possible. These are general modularising principles. For example, the great simplification in computer design and construction enabled by the LSI and VLSI modularisation of digital logic functions treat electrical level and timing uncertainty problems in just this way.

---

[1] Some people are puzzled by the idea of a sensorless robot coping with uncertainty. In assembly there are many sensorless tricks available by which uncertainty can be reduced. For example, chamfers can be used to guide pegs into holes, centering gripper fingers can be used to centralise off-centre parts, nudges can be used to align a part surface with the nudging surface, and so on.

3.6 *Hardware*

The experimental robot was an Adept robot with 5 degrees of freedom, and programmed in VAL2, a primitive BASIC-like language with robotic extensions. In terms of both robotic extensions and programming facilities, VAL2 was at the time one of the best commercially available robot programming languages.

Several varieties of SOMA puzzle were acquired, some plastic, some wood, some large, some small. The parts were no better made than they needed to be, i.e. a bit squint, not quite the right size, and so on. This was a deliberate way of introducing uncertainty into the system in order to exercise its reliability.

3.7 *The Behavioural Modules*

There were basically three of these:–

- GET PART: This picked up a part, coping with translation errors of up to about half a cube side, and rotational errors of up to about 30 degrees. It reduced these to about 2 degrees and 1mm of error.
- REGRASP PART: This regrasped a part in those cases where the part could not be picked up by the grip required to put it down.
- PUT DOWN PART: This placed the part in the assembly. Vertical uncertainty was coped with by using slipping fingers and pushing the part a bit too far down. Horizontal uncertainty was coped with by the planner arranging small gaps between the parts. At the end of the assembly, these gaps were removed by patting the assembly together.

The development of the initial system is described in detail in [20].

3.8 *The Assembly Planner*

The planner was designed as a hierarchical planner which would backtrack exhaustively through the search space. It was written in PROLOG. In effect each level produces a complete workable plan for an agent of appropriate capability. The hierarchy bottoms out when it reaches the capabilities of the behaviour-based agent.

- The first level solved the puzzle. It accepted as input the shape of the parts, their disposition on the table, and the outline of the shape to be made. It determined how the parts could be disposed to fill the shape, and what rotations the part had to undergo from its disposition on the table to its disposition in the assembly. For the cube there are 1440 such plans[2]. Given the 7 parts and approximately 12 ways each has of being fitted into the assembly this 1440 has been distilled from an unpruned search space of about 14,000,000,000 possibilities.
- The second level took this disposition of parts in the assembly, and found a gravitationally stable ordering of them. In order to keep the bulky hand and wrist of the robot from interfering with the assembly it also imposed the constraint that the assembly grew higher step by step. For each of the 1440 dispositions of parts in the cube there are about 70 of these orderings, i.e. approx 100,000.

[2] This figure of 1440 excludes rotations and reflections [4].

- The third level planned grasps. It listed all possible put down grasps, taking into account adjacent parts forbidding fingers on those sides, and all possible pick up grasps (there are usually two of each). Where there was a match between these lists no regrasp was necessary. Otherwise a regrasp was planned. The planner looked for a gravitationally stable placement of the part which allowed both one of the pick up grasps and one of the put down grasps to be applied. There are now about 3,000,000 variations in total.

  If it could not find such a regrasp it decided first whether the regrasp was impossible simply because of this particular ordering, or whether it was impossible in general (due to limitations of the robot). In this latter case no assembly could be performed with that part in that orientation in the assembly. So backtracking was forced back to that decision in the first level of planning which had placed that part in that orientation. If the problem was that some other part (or parts) were in the way of the fingers the planner analysed the situation to discover the latest part-placing decision which if changed might enable the regrasp to occur, and forced backtracking back to the point at which the decision was made.

  This is failure-directed backtracking. The more common dependency-directed backtracking makes use of dependencies (and associated backtrackable choices) known at the time of the choice, whereas failure-directed backtracking has to analyse the current failure in order to discover which choice to blame.

  Grasp planning introduced very severe constraints. Most assemblable dispositions required a few failure directed backtrackings, and about half of the part dispositions proved to be not assemblable because of finger/part clashes. In effect grasp constraints reject about 99

- The fourth and fifth levels never backtracked, they always succeeded. The fourth level introduced the horizontal gaps between parts used to accommodate placing errors.[3]. The fifth level translated the plan from a PROLOG nested list into parameterised VAL2 subroutine calls.

The development of the planner is described in summary in [20, 22].

4 Success of the System

The reliability of the system was assessed by running many different plans many times, every one a plan which the system had never before tried. The agent ran the plans at twice the design speed in order to provoke failures. 3% of failures were found. Half of these were due to minor software bugs in the planner. About 80% of the reminder were due to one particularly unstable kind of regrasp, curable by a simple change in one system parameter [21].

It had been claimed that the system would permit the easy integration of sensing. Chongstitvatana developed a vision system which guided the gripper to the part to be grasped [8, 9]. The key idea was forming local linear approximations between robot joint movements and movements of the robot gripper image in x,y pixels in the image, and iterating motions until the gripper was seen to be in the correct relation to the object.

[3] If one uses SOMA parts made of 5 cubes, then this level fails (backtracks) on detection of a subassembly.

Given two appropriately disposed cameras, correct relation in both images guaranteed correct positioning in the world. In line with the behaviour-based philosophy this system did not need to know the kinematics of the robot nor the parameters or location of the cameras, and thus was more robust than classical knowledge-based systems which require careful calibration. The robot arm could be bent, the camera moved, the lens changed, and so on, without affecting the successful operation of the system. This demonstrated that solving the sensing-integration problem within the behaviour-based paradigm produced a system which not only had a different architecture, but which was more robust than one designed within the classical Newtonian/Cartesian knowledge-based paradigm.

Although developed in 1987, this is still one of the very few high level automated assembly systems in existence. It is (as far as the author knows) the only one which is automated to such a high level, deciding assembly order and part disposition for itself. It is not entirely fair to claim all of this as due to the virtues of the hybrid system, since some of it is due to the SOMA puzzle domain, which reduces the geometric complexity of industrial strength assembly to simple arithmetic. It has clearly demonstrated, however, that the original hopeful expectations of the hybrid approach to assembly were well fulfilled. Careful analysis of the system helped us to refine our original vague ideas about the proper features and architecture of behaviour-based assembly systems and behaviour-based systems in general [22, 23].

Brooks's subsumption architecture sees behaviours as functional capabilities which may emerge in interactions between system components. Thus although they may be viewed externally as units of behaviour, internally they need not be unitary. We here insist on implementing behaviours as behavioural modules. This makes them not only units of functional capability both externally and internally, but also abstraction devices which enable the existence of a tractable representation of the world, and by means of their practical capability connect (ground) that representation to the world. Note that this is the indexical-functional kind of representation which Chapman and Agre claim provides unusual robustness and computational simplicity [7]. This is the crucial feature of the architecture which gives the planner its simplicity and the agent its reliability.

5 Behaviour, Purpose, and Meaning

Any research programme in AI is based upon certain ideas about what constitutes (some aspects of) mind, and thus is based upon a stance in philosophy of mind. The variety of paradigms[4] current in AI, such as classical symbolic AI, connectionism, or behaviour-based systems, reflect a variety of stances in philosophy of mind. As Varela et al point out ([31]) what distinguishes "enactive systems" (as they refer to behaviour-based systems) from the others is the conviction that representation is not the key issue, i.e. the mind is not primarily a mirror of nature, but a dynamic phenomenon emerging from the activity of an agent situated in its local world in a historical context. Like the view that cognition is fundamentally computational and representational (e.g. Hobbes and Hume), this view is much older than AI (e.g. Heidegger and Husserl).

The cybernetic thinkers of the 60s noted the extra power given this view by the development of control theory (e.g. [3, 27]), which smuggles the awkwardly subjective

[4]In the Kuhnian sense ([17]).

and mentalistic notion of purpose within the borders of science. Distasteful though some positivist purists find purpose, it is very tedious to try to describe the behaviour of goal-seeking things, whether thermostats or theologians, without reference their purposes. McCarthy's much-ridiculed "purposeful thermostat" [24] makes a serious point which ethologists are now taking up [10]. The purpose of some agent or device cannot be described without reference both to its local interactive world and history. Thus it is no co-incidence that the proponents of behaviour-based systems criticise the unsituated and unhistorical nature of the computational (symbolic) view of mind, and that behaviour-based systems are constructed from goal-seeking subsystems.

Searle in his Chinese Room argument ([28]) asserts that alleged computational implementations of mind must necessarily fail to catch the essence and can only mimic the external behaviour: the lack of subjective phenomena ("nobody at home") shows that there is a missing ingredient in the computational recipe of mind. Harnad argues that the important missing ingredient is grounding of the symbols [15]. There are various interpretations of these arguments, depending on whether one understands "meaning" as denoting purely behavioural capacity (as the Turing Test does for artificial intelligence), as necessarily involving conscious awareness (as Searle requires), or as requiring the right behaviour to have been produced by the right kind of machinery (e.g. Brian Smith's *Knowledge Representation Hypothesis* [30]. Note that in this hybrid architecture that the symbolic world of the planner is grounded (the symbols maintained in secure attachment to the world) by the fact that the symbols are derived from the purposes and parameters of the behavioural modules. The same feature of the architecture which simplifies the planner and makes its plans reliable supplies inherent semantics to the terms in which it operates. This is Agre and Chapman's indexical-functional kind of representation. Not only does it provide robust grounding and computationla economy, but it is derived from a Heideggerian critique of symbolic AI, and in particular of symbolic planning. This is what provides the missing semantics in the behavioural interpretation of the Chinese Room criticism[5].

6 Conclusion

It is argued that the problems of intractability and complexity which have beset many attempts at assembly planning are due to an inappropriate architecture in which too much is asked of the symbolic (planning) level, because of a tempting confusion between formal description and implementation which is inherent in the Knowledge Representation Hypothesis underlying the symbolic AI research programme [30].

A hybrid architecture in which a classical symbolic assembly planner makes plans for a behaviour-based robot plan execution agent is described. The assembly domain is the SOMA puzzle, a set of bent and forked bricks capable of hundreds of different assemblies. Novel plans for novel assemblies are executed reliably in the presence of uncertainty. It is argued that this is a consequence of the hybrid architecture. The architecture of the agent is based on the general principles underlying Brooks's subsumption architecture for mobile robots, but with the crucial difference that externally unitary behaviours are implemented as internally unitary modules. This permits the capabilities of the modules to act as abstraction devices which create the world in which the planner

[5] A more detailed derivation of such system *autosemantics* from the *autoteleology* of goal-seeking subsystems is sketched in [14].

operates, an ideal Platonic world in which there is no uncertainty. In this way the classical symbolic planner is provided both with a world well suited to its capabilities, and with reliable execution of its plans.

This hybrid architecture is based on a situated, interactive, and historicist view of mind similar to the view from which Dreyfus criticises classical symbolic AI [11]. It answers the criticism of the weak interpretation of Searle's Chinese Room [28] by providing the missing symbol grounding [15] via the purposeful nature of the behavioural modules of the agent.

References

[1] Ambler, A.P., and Popplestone, R.P., "Inferring the positions of bodies from specified spatial relationships", Artificial Intelligence 6, 2 (1975), pp 228-233.

[2] Andersson R.L., "A robot ping-pong player experiment in real-time intelligent control", MIT Press 1988.

[3] Bateson, G., "The Cybernetics of "Self" : A Theory of Alcoholism", in Psychiatry, Vol 34, no 1, pp 1-18, 1971; reprinted in "Steps to an Ecology of Mind", Ballantine Books, NY, 1972.

[4] Berlekamp, Conway and Guy, vol 2 of "Winning Ways", Academic Press, 1982.

[5] Brooks, R.A., "Achieving Artificial Intelligence Through Building Robots", AI Memo 899, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, May 1986.

[6] Chapman, David, *Planning for Conjunctive Goals*, Tech Report 802, MIT AI Lab, Mass 1985.

[7] Chapman, David, and Agre, Philip, *Abstract Reasoning as Emergent from Concrete Activity*, in "Reasoning about Actions and Plans", eds Georgeff and Lansky, Morgan Kaufmann 1986.

[8] Chongstitvatana, P., "The Design and Implementation of Vision-based Behavioural Modules for a Robotic Assembly System", PhD thesis, Edinburgh University, 1992.

[9] Conkie, A., and Chongstitvatana, P., "Active Mobile Stereo Vision for Robotic Assembly", Proceedings of the 23rd Symposium on Industrial Robots, Barcelona, Spain, 1992; also available as RP 590, Department of Artificial Intelligence, University of Edinburgh.

[10] Dennett, Daniel, *The intentional stance*, MIT Press paperback ed., MIT Press, 1989.

[11] Dreyfus H.L., "What computers still can't do: a critique of artificial reason", MIT Press 1992.

[12] Fleming, A., "Analysis of Uncertainties in a Structure of Parts 1", and "Analysis of Uncertainties in a Structure of Parts 2", DAI Research Papers No. 271 and 272, Department of Artificial Intelligence, University of Edinburgh, 1985.

[13] Gardner, M., "More Mathematical Puzzles and Diversions", 1961, Penguin 1966 edition.

[14] Hallam, J.C.T., and Malcolm, C.A., "Behaviour, Perception, and Action — the View from Situated Robotics", Phil. Trans. R. Soc. Lond. A 1994.

[15] Harnad, S., "Minds, Machines, and Searle", Journal of experimental and Theoretical Artificial Intelligence, Vol 1 Iss 1, Jan 1989.

[16] Kaelbling, Leslie Pack, *An architecture for intelligent reactive systems*, Artificial Intelligence Center, SRI International, Technical note (SRI) 400, 1986.

[17] Kuhn, T.S., "The structure of scientific revolutions", 2nd ed., enl. University of Chicago Press 1970

[18] Lozano-Perez, T., and Brooks, R., "An Approach to Automatic Robot Programming", MIT AI Lab, AIM 842, 1985.

[19] Lozano-Perez, T., "Handeye, a robot task planner", MIT Press 1992

[20] Malcolm, C.A., Planning and Performing the Robotic Assembly of Soma Cube Constructions, MSc Dissertation Edinburgh University, September 1987, 87 pages.

[21] Malcolm, C.A., and Smithers, T., Programming Assembly Robots in terms of Task Achieving Behavioural Modules: First Experimental Results, Proc International Advanced Robotics Programme, Second Workshop on Manipulators, Sensors and Steps towards Mobility, Manchester, UK, Oct 24-26 1988; also Edinburgh University DAI RP 410.

[22] Malcolm, C., and Smithers, T., "Symbol Grounding via a Hybrid Architecture in an Autonomous Assembly System", proceedings of Cost-13 Workshop, Representation and Learning in an Autonomous Agent, Nov 16-18, 1988, Lagos, Portugal, 30 pages; also Edinburgh University DAI RP 420.

[23] Malcolm, C., Smithers, T., and Hallam, J., "An Emerging Paradigm in Robot Architecture", invited paper at the Intelligent Autonomous Systems Conference (2) in Amsterdam, Dec 11-14, 1989; also available as Edinburgh University DAI RP 447.

[24] McCarthy, John, "Ascribing Mental Qualities to Machines", Stanford University Reports STAN-CS-79-725, 1979.

[25] Popplestone, R.J, Ambler, A.P, Bellos, I, *RAPT: A Language for describing assemblies*, **Industrial Robot**, Sep 1978, 131-137.

[26] Popplestone, R.J., Ambler, A.P., and Bellos, I., "An Interpreter for a Language for Describing Assemblies", Artificial Intelligence Vol. 14, No. 1, pp. 79-107, 1980.

[27] Powers, W.T., "Behaviour: The Control Of Perception", Wildwood House, 1973.

[28] Searle, J., "Minds, Brains, and Programs", Behavioral and Brain Sciences, 3, p417-457, 1980.

[29] Shoham, Yoav, *What is the Frame Problem?*, in "Reasoning about Actions and Plans", eds Georgeff and Lansky, Morgan Kaufmann 1986.

[30] Smith, B.C., "Prologue to Reflection and Semantics in a Procedural Language", in Readings in Knowledge Representation, eds Brachman & Levesque, Morgan Kaufmann, 1985.

[31] Varela, F.J., Thomson E., and Rosch E., "The Embodied Mind", MIT Press 1991