The Dynamics of Intelligence: Constraint-Satisfying Hybrid Systems For Perceptual Agents

Alan Mackworth
Laboratory for Computational Intelligence
Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1Z4, Canada
mack@cs.ubc.ca, www.cs.ubc.ca/spider/mack

Abstract

Methods for designing and building perceptual agents should be clean, powerful and practical. But no methodology satisfies all three criteria, yet. Our methodologies are evolving dialectically. The symbolic methods of Good Old-Fashioned Artificial Intelligence and Robotics (GOFAIR) constitute the original thesis. The antithesis is reactive Insect AI. The emerging synthesis, Situated Agents, needs formal rigor and practical tools. A robot is a hybrid intelligent dynamical system, consisting of a controller coupled to its body. The Constraint Net (CN) model of Zhang and Mackworth is a unitary framework for building hybrid intelligent systems as situated agents. Most other robot design methodologies use hybrid models of hybrid systems, awkwardly combining offline computational models of high-level perception, reasoning and planning with online models of low-level sensing and control. In CN, the designer specifies the robot's vision, control and motor systems uniformly as online systems. The constraint-based architecture for agent perceiver/controllers consists of multi-layer constraint-satisfying modules. If the perceptual and control systems are designed as constraint-satisfying devices then the total robotic system, consisting of the robot symmetrically coupled to its environment, may, sometimes, be proven correct. In some cases, a controller may be synthesized from a constraint-based specification. This framework has co-evolved with applications to several robotic tasks, including the the challenge of building various visually-controlled robot soccer players. The dynamics of intelligence can be captured by constraint-satisfying hybrid systems.

Introduction

We need practical and formal design methodologies for building integrated perceptual agents. Here I argue for a formal approach to the emerging synthesis, Situated Agents. The approach is based on the view that an agent or robot is, typically, a hybrid intelligent dynamical system, consisting of a controller coupled to its body. The robot is symmetrically coupled to a dynamic environment.

Similarly, knowledge-based image interpretation needs to be re-interpreted. The traditional Good Old-Fashioned Artificial Intelligence and Robotics (GOFAIR) approach proposes that domain-specific knowledge is used by the robot/agent at run-time to disambiguate the retinal array into a rich world representation. The argument is that the impoverishment and ambiguity of the visual stimulus array must be supplemented by additional knowledge. This approach has failed to make substantial progress for several reasons. One difficulty is the engineering problem of building robots by integrating offline knowledge-based vision systems with online control-based motor systems. Especially in active vision systems (Ballard 1989) this integration is difficult, ugly and inefficient (Mackworth 1993). Because of such objections, some in the AI-robotics community have rejected the knowledge-based approach adopting instead an *ad hoc* Gibsonian situated approach to perception and action that exploits regularities of the particular environmental niche of the robot (Horswill & Brooks 1988; Andersson 1988; Chapman 1990; Brooks 1991). However, in (Mackworth 1996), I argued that, with a radical re-interpretation of 'knowledge-based', we *can* design, build and verify quick and clean knowledge-based situated robot vision systems.

The Agent Design Problem

The robotic agent design problem is formidable, regardless of whether the robot is designed or modified by a human, by nature (evolution), by another robot (bootstrapping), or by itself (learning). A robot is, typically, a hybrid intelligent system, consisting of a controller coupled to its body. The controller and the body each consist of discrete-time, continuous-time or event-driven components operating over discrete or continuous domains. The controller has perceptual subsystems that can (partially) observe the state of the environment and the state of the body.

Robot design methodologies are evolving dialectically (Mackworth 1993). The symbolic methods of GOFAIR constitute the original thesis. The antithesis is reactive Insect AI. The emerging synthesis, Situated Agents, has promising characteristics, but needs formal rigor and practical tools. The critiques and rejection, by some, of the GOFAIR paradigm have given rise to the sundry Situated Agent approaches (Rosenschein & Kaelbling

1986; Winograd & Flores 1986; Ballard 1989; Lavignon & Shoham 1990; Brooks 1991; Mackworth 1993; Zhang & Mackworth 1994b).

The Robot Soccer Challenge

Theory is vacuous without an appropriate application to drive designs, experiments and implementations. In 1992, I proposed robot soccer as a grand challenge problem for the field (Mackworth 1993) since it has the task characteristics that force us to confront the fundamental issues in a practical way for a perceptual, collaborative, real-time task with clear performance criteria. At the same time, I described the first system for playing robot soccer. Since then it has been a very productive testbed both for our laboratory (Barman *et al.* 1993; Sahota & Mackworth 1994; Mackworth 1996; Zhang & Mackworth 1995c; 1998a; 1998b) and for many other groups around the world, stimulating research toward the goal of building visual robotic systems that are clean and practical.

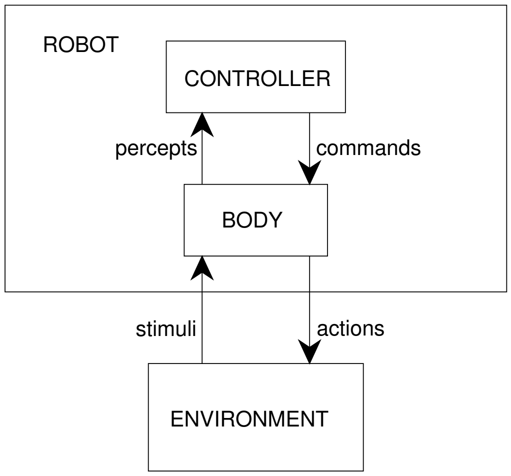

Figure 1: The structure of a robotic system

Constraint Nets

Ying Zhang and Mackworth have developed the Constraint Net (CN) model (Zhang & Mackworth 1995a), as a model for building hybrid intelligent systems as Situated Agents. In CN, a robotic system is modelled formally as a symmetrical coupling of a robot with its environment. Even though a robotic system is, typically, a hybrid dynamic system, its CN model is unitary. Most other robot design methodologies use hybrid models of hybrid systems, awkwardly combining offline computational models of high-level perception, reasoning and planning with online models of low-level sensing and control.

CN is a model for robotic systems software implemented as modules with I/O ports. A module performs a transduction from its input traces to its output traces, subject to the principle of causality: an output value at any time can depend only on the input values before, or at, that time. The model has a formal semantics based on the least fixpoint of sets of equations (Zhang & Mackworth 1995a). In applying it to a robot operating in a given environment, one separately specifies the behaviour of the robot body, the robot control program, and the environment. The total system can then be shown to have various properties, such as safety and liveness, based on provable properties of its subsystems. This approach allows one to specify and verify models of embedded control systems. Our goal is to develop it as a practical tool for building real, complex, sensor-based robots. It can be seen as a development of Brooks' subsumption architecture (Brooks 1991) that enhances its modularity and scalability while avoiding the limitations of the augmented finite state machine approach.

A robot situated in an environment can be modelled as three machines: the robot body, the robot controller and the environment, as shown in Figure 1. Each can be modelled separately as a dynamical system by specifying a CN with identified input and output ports. The robot is modelled as a CN consisting of a coupling of its body CN and its controller CN by identifying corresponding input and output ports. Similarly the robot CN is coupled to the environment CN to form a closed robot-environment CN.

The CN model is realized as an online dataflow-like distributed programming language with a formal algebraic denotational semantics and a specification language, a real-time temporal logic, that allows the designer to specify and prove properties of the situated robot by proving them of the robot-environment CN. We have been able to specify, design, verify and implement systems for a robot that can track other robots (Zhang & Mackworth 1992), a robot that can escape from mazes and a two-handed robot that assembles objects (Zhang & Mackworth 1994b), an elevator system (Zhang & Mackworth 1993) and a car-like robot that can plan and execute paths under non-holonomic constraints (Zhang & Mackworth 1995c).

Although CN can carry out traditional symbolic computation online, such as solving Constraint Satisfaction Problems and path planning, notice that much of the symbolic reasoning and theorem-proving may be outside the agent, in the mind of the designer. GOFAIR does not make this distinction, assuming that such symbolic reasoning occurs explicitly in, and only in, the mind of the agent.

The question "Will the robot do the right thing?" (Zhang & Mackworth 1994b) is answered if we can:

1. model the coupled robotic system at a suitable level of abstraction,
2. specify the required global properties of the system's evolution, and
3. verify that the model satisfies the specification.

In CN the modelling language and the specification

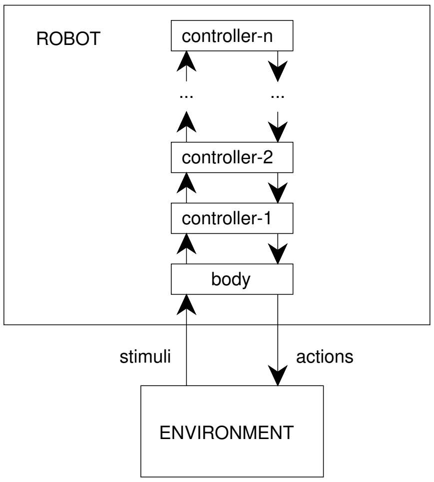

Figure 2: A hierarchical hybrid robot controller

language are totally distinct since they have very different requirements. The modelling language is a generalized dynamical system language. Two versions of the specification language, Timed Linear Temporal Logic (Zhang & Mackworth 1995c) and Timed ∀-automata (Zhang & Mackworth 1994a), have been developed with appropriate theorem-proving and model-checking techniques for verifying systems.

In (Poole, Mackworth, & Goebel 1998, Chapter 12) we describe how to build a situated robot controller using CN as realized in a logic program.

Constraint-Satisfying Controllers

Many robots can be designed as online constraint-satisfying devices (Zhang & Mackworth 1994a; 1995b; 1995c). A robot in this restricted scheme can be verified more easily. Moreover, given a constraint-based specification and a model of the body and the environment, automatic synthesis of a correct constraint-satisfying controller becomes feasible, as shown for a simple ball-chasing robot in (Zhang & Mackworth 1995c).

A constraint is simply a relation on the phase space of the robotic system, which is the product of the controller, body and environment spaces. A controller is defined to be *constraint-satisfying* if it, repeatedly, eventually drives the system into an ϵ-neighborhood of the constraint using a constraint satisfaction method such as gradient descent or a symbolic technique.

A constraint-satisfying controller may be *hierarchical* with several layers of controller above the body, as shown in Figure 2. In this case each layer must satisfy the constraints, defined on its state variables, appropriate to the layer, as, typically, set by the layer above. The layers below each layer present to that layer as a virtual robot body in a suitably abstract state space (Zhang & Mackworth 1995c; 1998b). The lower layers are, typically, reactive and synchronous (or in continuous time) on continuous state spaces; the upper layers are more deliberative and asynchronous (or event-triggered) in symbolic, discrete spaces.

Soccer-Playing Robots

The ideas presented here have been developed and tested by application to the challenge of designing, building and verifying active perception systems for robot soccer players with both off-board and on-board vision systems.

In the Dynamo (Dynamics and Mobile Robots) project in our laboratory, we have experimented, since 1991, with multiple mobile robots under visual control. The Dynamite testbed consists of a fleet of radio-controlled vehicles that receive commands from a remote computer. Using our custom hardware and a distributed MIMD environment, vision programs are able to monitor the position and orientation of each robot at 60 Hz; planning and control programs generate and send motor commands at the same rate. This approach allows umbilical-free behaviour and very rapid, lightweight fully autonomous robots. Using this testbed we have demonstrated various robot tasks (Barman *et al.* 1993), including playing soccer (Sahota & Mackworth 1994) using a 2-layer deliberative/reactive controller architecture.

One of the Dynamo robots, Spinoza, is a self-contained robot consisting of an RWI base with an RGB camera on a pan-tilt platform mounted as its head and a trinocular stereo camera in its base. As a illustration of these ideas, consider the task for Spinoza of finding, tracking, chasing and kicking a soccer ball, using the pan-tilt camera. After locating the moving ball Spinoza is required to track it, move to within striking distance of the ball and strike it. The available motor commands control the orientation of the base, the forward movement of the base, and the pan and tilt angles of the camera. The parameters can be controlled in various relative/absolute position modes or rate mode. The available rate of pan substantially exceeds the rate of base rotation. A hierarchical constraint-based active-vision controller, using prioritized constraints and constraint arbiters, can be specified for Spinoza that will, repeatedly, achieve and maintain (or re-achieve) the desired goal subject to safety conditions such as staying inside the soccer field, avoiding obstacles and not accelerating too quickly. If the dynamics of Spinoza and the ball are adequately modelled by the designer then this constraint-based vision system will be guaranteed to achieve its specification.

Yu Zhang and I have recently extended these ideas to build 3-layer constraint-satisfying controllers for a complete soccer team (Zhang & Mackworth 1998a). The controllers for our new softbot soccer team, UBC Dynamo98, are modelled in CN and implemented in Java, using the Java Beans architecture (Zhang & Mackworth 1998b). They control the soccer players' bodies in the Soccer Server developed by Noda Itsuki for RoboCup. These experiments provide evidence that the constraint-based CN approach is a clean and practical design framework for perceptual robots.

Conclusions

I have described the motivation, some results and some current directions of a long-term project intended to develop a new approach to the specification, design, implementation and evaluation of robotic systems. The practical result is a new methodology for building intelligent, perceptual systems.

In short, the dynamics of intelligence can be captured by constraint-satisfying hybrid systems.

Acknowledgments

I am most grateful to Ying Zhang and Yu Zhang for our collaborations. I also thank Rod Barman, Cullen Jennings, Stewart Kingdon, Jim Little, Valerie McRae, Don Murray, Dinesh Pai, David Poole, Michael Sahota, and Vlad Tucakov for help with this. This work is supported, in part, by the Natural Sciences and Engineering Research Council of Canada and the Institute for Robotics and Intelligent Systems Network of Centres of Excellence.

References

Andersson, R. L. 1988. *A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control.* Cambridge, MA: MIT Press.

Ballard, D. H. 1989. Reference frames for active vision. In *Proceedings IJCAI-89*, 1635–1641.

Barman, R. A.; Kingdon, S. J.; Little, J. J.; Mackworth, A. K.; Pai, D. K.; Sahota, M.; Wilkinson, H.; and Zhang, Y. 1993. Dynamo: Real-time experiments with multiple mobile robots. In *Intelligent Vehicles Symposium*, 261–266.

Brooks, R. A. 1991. Intelligence without reason. In *IJCAI-91*, 569–595.

Chapman, D. 1990. Vision instruction and action. Technical Report MIT AI TR-1085, MIT, Cambridge, MA.

Horswill, I. D., and Brooks, R. A. 1988. Situated vision in a dynamic world: Chasing objects. In *AAAI-88*, 796–800.

Lavignon, J., and Shoham, Y. 1990. Temporal automata. Technical Report STAN-CS-90-1325, Stanford University, Stanford, CA.

Mackworth, A. K. 1993. On seeing robots. In Basu, A., and Li, X., eds., *Computer Vision: Systems, Theory, and Applications.* Singapore: World Scientific Press. 1–13.

Mackworth, A. K. 1996. Quick and clean: Constraint-based vision for situated robots. In *IEEE Int'l. Conf. on Image Processing*, 789–792.

Poole, D. L.; Mackworth, A. K.; and Goebel, R. G. 1998. *Computational Intelligence: A Logical Approach.* New York: Oxford University Press.

Rosenschein, S. J., and Kaelbling, L. P. 1986. The synthesis of machines with provable epistemic properties. In Joseph Halpern., ed., *Proc. Conf. on Theoretical Aspects of Reasoning about Knowledge.* Los Altos, CA: Morgan Kaufmann. 83–98.

Sahota, M., and Mackworth, A. K. 1994. Can situated robots play soccer? In *Proc. Artificial Intelligence 94*, 249 – 254.

Winograd, T., and Flores, F. 1986. *Understanding Computers and Cognition.* Reading, MA: Addison-Wesley.

Zhang, Y., and Mackworth, A. K. 1992. Modeling behavioral dynamics in discrete robotic systems with logical concurrent objects. In Tzafestas, S. G., and Gentina, J. C., eds., *Robotics and Flexible Manufacturing Systems.* Elsevier Science Publishers B.V. 187–196.

Zhang, Y., and Mackworth, A. K. 1993. Design and analysis of embedded real-time systems: An elevator case study. Technical Report TR 93-4, UBC, Vancouver, B.C.

Zhang, Y., and Mackworth, A. K. 1994a. Specification and verification of constraint-based dynamic systems. In Borning, A., ed., *Principles and Practice of Constraint Programming*, number 874 in Lecture Notes in Computer Science. Springer-Verlag. 229 – 242.

Zhang, Y., and Mackworth, A. K. 1994b. Will the robot do the right thing? In *Proc. Artificial Intelligence 94*, 255 – 262.

Zhang, Y., and Mackworth, A. K. 1995a. Constraint Nets: A semantic model for hybrid dynamic systems. *Theoretical Computer Science* 138:211 – 239.

Zhang, Y., and Mackworth, A. K. 1995b. Constraint programming in constraint nets. In V. Saraswat and P. Van Hentenryck., ed., *Principles and Practice of Constraint Programming.* Cambridge, MA: The MIT Press. chapter 3, 49–68.

Zhang, Y., and Mackworth, A. K. 1995c. Synthesis of hybrid constraint-based controllers. In Antsaklis, P.; Kohn, W.; Nerode, A.; and Sastry, S., eds., *Hybrid Systems II*, Lecture Notes in Computer Science 999. Springer Verlag. 552 – 567.

Zhang, Y., and Mackworth, A. K. 1998a. A multi-level constraint-based controller for the Dynamo98 robot soccer team. In *RoboCup-98.* Paris, France: Springer-Verlag. To appear.

Zhang, Y., and Mackworth, A. K. 1998b. A constraint-based controller for soccer-playing robots. In *Proceedings of IROS '98*, 1290 – 1295.