Constraints: An International Journal, 2, 83–86 (1997)
© 1997 Kluwer Academic Publishers. Manufactured in The Netherlands.

Constraint-Based Design of Embedded Intelligent Systems

ALAN K. MACKWORTH mack@cs.ubc.ca
Laboratory for Computational Intelligence, Department of Computer Science, University of British Columbia, Vancouver, B.C., Canada V6T 1Z4

Abstract. Substantial progress has been achieved using the standard Constraint Satisfaction Problem framework. However, there is a major unsolved challenge confronting the constraint research community: the constraint-based design of embedded intelligent systems. This requires a new online model of constraint satisfaction and new computational tools for specifying, modeling, verifying and implementing constraint-based, hybrid, intelligent systems, such as robots. The Constraint Net model of Zhang and Mackworth allows the design of hybrid intelligent systems as situated robots: modeling the robot and the environment symmetrically as dynamic systems. If the robot's perceptual and control systems are designed as constraint-satisfying devices then the total robotic system, consisting of the robot symmetrically coupled to the environment, can be proven correct. Some theoretical and practical advances based on this model are described, including experiments with the constraint-based design of robot soccer players.

Keywords: constraint satisfaction, dynamic systems, hybrid systems, robotics, intelligent systems, constraint net, robot soccer

1. Introduction

The Constraint Satisfaction Problem (CSP) paradigm has evolved and matured over the last two decades. The algorithms developed in the CSP paradigm were made more available and more useful when they were incorporated into the Constraint Programming (CP) language paradigm. This success should be celebrated and more needs to be done in both paradigms.

2. Embedded Intelligent Systems

Despite this success, however, a major challenge still facing the constraint research community is to develop useful theoretical and practical tools for the constraint-based design of embedded intelligent systems. An archetypal example of an application in this class is the design of controllers for sensory-based robots [6, 4]. If we examine this problem we see that almost all the tools developed to date in the CSP and CP paradigms are inadequate for the task, despite the superficial attraction of the constraint-based approach.

The fundamental difficulty is that, for the most part, the CSP and CP paradigms presume an offline model of computation. But intelligent systems embedded as controllers in real physical systems must be designed in an online model. Moreover, the online model must be based on various time structures: continuous, discrete and event-based. The requisite online computations, or transductions, are to be performed over various type structures including continuous and discrete domains. These hybrid systems require new models of computation, constraint satisfaction and constraint programming. To this end, Zhang and Mackworth [5] defined constraint satisfaction as a dynamic system process that approaches asymptotically the solution set of the given, possibly time-varying, constraints. Under this view, constraint programming is the creation of a dynamic system with the required property.

In the classic 'knowledge-based' Good Old-Fashioned Artificial Intelligence and Robotics (GOFAIR) paradigm, robot architectures are based on offline perception and reasoning [3]. This approach has failed to make substantial progress for several reasons. One difficulty is the engineering problem of building robots by integrating offline perception and reasoning systems with online control-based motor systems; this integration is difficult, ugly and inefficient. 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 that exploits regularities of the particular environmental niche of the robot [2]. However, with a radical re-interpretation of 'knowledge-based', we *can* design, build and verify quick and clean knowledge-based situated robot systems.

3. The Constraint Net Model

The Constraint Net (CN) framework [7] is a formal and practical model for building hybrid intelligent systems as situated agents [6]. In CN, a robotic system is modeled 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 [7]. In applying it to a robot operating in a given environment, one separately models the dynamics of the robot plant, the robot control program, and the environment. The total system may 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.

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.

In CN the modeling language and the specification language are totally distinct since they have very different requirements. The modeling language is a generalized dynamical system language. Two versions of the specification language, Timed Linear Temporal Logic

[9] and Timed ∀-automata [5], have been developed with appropriate theorem-proving and model-checking techniques for verifying systems.

4. Constraint-Based Robots

Many robots can be designed as online constraint-satisfying devices [5, 8, 9]. A robot in this restricted scheme can be verified more easily. Moreover, given a constraint-based specification and a model of the plant and the environment, automatic synthesis of a correct constraint-satisfying controller becomes feasible, as shown for a simple ball-chasing robot in [9].

Theory is vacuous without an appropriate application to drive designs, experiments and implementations. These ideas are being tested by their application to the task of designing, building and verifying perception, communication and planning systems for robot soccer players with off-board or on-board vision systems.

In the Dynamo Project in our laboratory, we are experimenting 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 behavior and very rapid, lightweight fully autonomous robots. Using this testbed we have demonstrated various robot tasks [1], including playing soccer [4].

5. Conclusion

Using situated robots as one of our target applications we are developing the framework of a new approach to the problem of constraint-based design of embedded intelligent systems. This is a benchmark problem posed as a challenge for constraint-based theories of computation.

Acknowledgments

I am grateful to Rod Barman, Cullen Jennings, Stewart Kingdon, Jim Little, Valerie McRae, Don Murray, Dinesh Pai, Michael Sahota, Vlad Tucakov and Ying Zhang 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

1. Barman, R. A., Kingdon, S. J., Little, J. J., Mackworth, A. K., Pai, D. K., Sahota, M., Wilkinson, H., & Zhang, Y. (1993). DYNAMO: Real-time experiments with multiple mobile robots. In *Intelligent Vehicles*

Symposium, pages 261–266, Tokyo.

2. Horswill, I. D. & Brooks, R. A. (1988). Situated vision in a dynamic world: Chasing objects. In *AAAI-88*, pages 796–800, St. Paul, MN.
3. Mackworth, A. K. (1993). On seeing robots. In Basu, A. and Li, X., editors, *Computer Vision: Systems, Theory, and Applications*, pages 1–13. World Scientific Press, Singapore.
4. Sahota, M. & Mackworth, A. K. (1994). Can situated robots play soccer? In *Proc. Artificial Intelligence 94*, pages 249 – 254, Banff, Alberta.
5. Zhang, Y. & Mackworth, A. K. (1994a). Specification and verification of constraint-based dynamic systems. In Borning, A., editor, *Principles and Practice of Constraint Programming*, number 874 in Lecture Notes in Computer Science, pages 229 – 242. Springer-Verlag.
6. Zhang, Y. & Mackworth, A. K. (1994b). Will the robot do the right thing? In *Proc. Artificial Intelligence 94*, pages 255 – 262, Banff, Alberta.
7. Zhang, Y. & Mackworth, A. K. (1995a). Constraint Nets: A semantic model for hybrid dynamic systems. *Theoretical Computer Science*, 138:211 – 239.
8. Zhang, Y. & Mackworth, A. K. (1995b). Constraint programming in constraint nets. In V. Saraswat and P. Van Hentenryck, editor, *Principles and Practice of Constraint Programming*, chapter 3, pages 49–68. The MIT Press, Cambridge, MA.
9. Zhang, Y. & Mackworth, A. K. (1995c). Synthesis of hybrid constraint-based controllers. In Antsaklis, P., Kohn, W., Nerode, A., and Sastry, S., editors, *Hybrid Systems II*, Lecture Notes in Computer Science 999, pages 552 – 567. Springer Verlag.