Constraint-Based Agents

Alexander Nareyek
German National Research Center for Information Technology (GMD)
Research Institute for Computer Architecture and Software Technology (FIRST)
Rudower Chaussee 5
D - 12489 Berlin, Germany
alex@first.gmd.de

Abstract

The following paper defines a framework for *constraint-based agents*, as used within the EXCALIBUR project. The underlying dynamic real-time environment demands special properties of constraint-based agents, such as dynamic adaptation and real-time behavior. Underlying models and algorithms have to ensure these features.

Introduction

The increasing availability of distributed information and computation capacities, from client-server solutions to intranets and the internet, raises new opportunities and requirements on computer science. The term "agent" gains more and more relevance. (Wooldridge & Jennings 1995) define autonomy, social ability, reactivity, and pro-activeness as essential properties of these agents.

The agent concept can be used to simplify the solution of large problems by distributing them to some collaborating problem solving units. This kind of strategy is called *distributed problem solving.* On the other hand the agent concept supports more general interactions of already distributed units, which is subject to *multi-agent system* research. The focus of this paper is on multi-agent systems within dynamic real-time environments.

A crucial aspect of an agent is the way its behavior is determined. If there shall be no restriction to reactive actions an underlying planning system is needed. A lot of research has been done on planning, and a wide range of planning systems was developed, like STRIPS (Fikes & Nilsson 1971), UCPOP (Penberthy & Weld 1992), or PRODIGY (Veloso et al. 1995).

The advances in constraint programming suggest its application in planning problems, where planning is treated as a constraint satisfaction problem. Recent remarkable results with Graphplan (Blum & Furst 1997) and Satplan (Kautz & Selman 1996) proove the appropriateness of this approach.

Required Properties

Planning (resp. constraint programming) has to deal with the special properties of *dynamic adaptation, real-time behavior* and *social abilities.* The following sections will discuss these matters in more detail. Figure 1 shows some related work within the field of constraint programming. The cross indicates the aspired integration.

Dynamic Adaptation

Because of the limited view of an agent and the changing environment, an agent has to be adaptive. New information will restrict or relax the possible actions of an agent. When usual constraint solvers face relaxations they recompute the entire system again. Thereby the efficiency of the computation gets worse as well as the stability of a solution. Incremental approaches (Ramalingam & Reps 1993) overcome this by updating only the affected parts, rather than recomputing the whole system.

As this problem plays an important role in various fields, a lot of work on incremental approaches has been done which aim at the so called *Dynamic Constraint Satisfaction Problem.* A short survey can be found in (Verfaillie & Schiex 1994).

There can also be a need for dynamic adaptation *during* the planning phase. As the optimal plan length is not known in advance, the size of the constraint system might change during the search for a solution. Examples of constraint-based planning systems which perform a problem expansion during the search are Graphplan and *Descartes* (Joslin & Pollack 1996).

Real-time Behavior

In a dynamic real-time world an agent does not have unlimited time to think about an optimal plan. Some planning systems incorporated this idea, and replaced the *deliberative* planning by *reactive* behavior rules, where reasoning is abandoned. The most famous reactive system is PENGI (Agre & Chapman 1987).

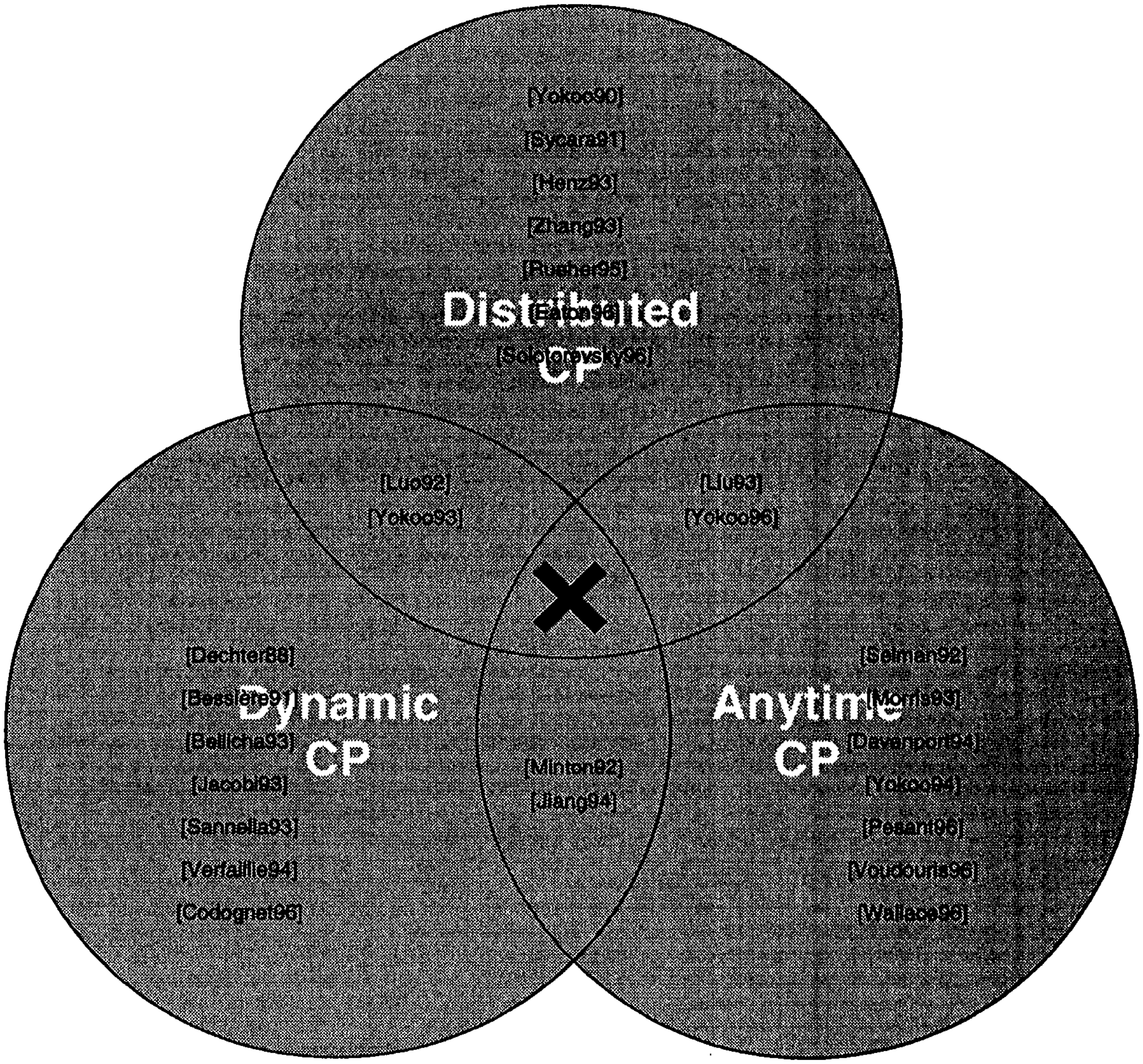

Figure 1: Related work and integration target

Also hybrid architectures like PRS (Georgeff & Lansky 1987) were developed.

Just as the early planning systems, most of the existing constraint-based systems compute the search for solutions off-line. An approach to eliminate this weakness can be the application of anytime algorithms (Zilberstein & Russell 1995). These techniques provide a solution at any time, whilst the quality of the solution is subject to a permanent improvement. In particular large problems and short time limits result in considerable advantages over classical methods (Wallace & Freuder 1996). An advantage over hybrid architectures is the continuous transition from reaction to deliberation. First steps towards an application of anytime algorithms in planning were made by (Dean & Boddy 1988).

In (Kautz & Selman 1992) and (Kautz & Selman 1996) planning is treated as a satisfaction problem and solved by the application of iterative local search techniques. Local search methods like simulated annealing, GSAT, Walksat, tabu search or genetic algorithms qualify for the use in anytime systems. Furthermore the method of iterative improvement is predestined for a combination with Dynamic Constraint Satisfaction, Constraint Satisfaction Optimization, and Partial Constraint Satisfaction.

Most of the iterative methods are incomplete, and it is possible to get trapped in local optima or on plateaus. But with increasing dynamics of the agent's environment, the importance of this disadvantage decreases.

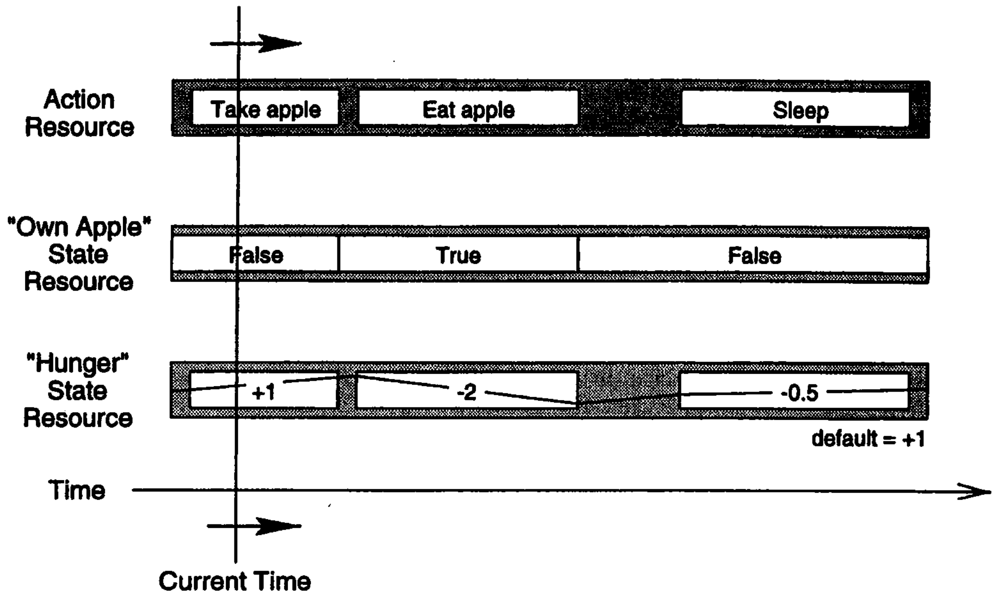

Figure 2: Plan task scheduling

Social Abilities

In a distributed environment the interaction between the agents plays an important role. When agents cooperate to pursue group goals, they should come to a globally (sub)optimal solution, which is likely to be different from locally (sub)optimal solutions. Also questions of coordination, the structure of the agent organization (centralized/decentralized), and communication protocols have to be handled.

In the domain of constraint programming most conceptional work focuses on distributed problem solving, rather than multi-agent systems. In the *Distributed Constraint Satisfaction Problem* as defined in (Yokoo, Ishida & Kuwabara 1990), agents play the role of contributing computation units instead of autonomous actors with specific intentions. While these techniques are useful to pursue superior group goals, higher-level concepts of *Distributed Artificial Intelligence* are more relevant regarding social interactions within multi-agent systems, such as (Levesque, Cohen & Nunes 1990), (Tidhar et al. 1992) and (Wooldridge & Jennings 1996).

From the Situation Calculus to Temporal Intervals

A major design question is the appropriate representation of time. Most existing planning systems use a STRIPS-like representation, which is based on the situation calculus (McCarthy & Hayes 1969). This approach uses forward branching time point structures.

There has been a lot of criticism on this representation. Problems include continuous changes, simultaneous actions, and actions or events that last over a period of time. Especially in multi-agent domains with temporally complex actions, the STRIPS-like representations is rather ineligible.

An approach to overcome these weak points are interval-based representations. Allen's interval temporal logic was introduced in (Allen 1983) and (Allen 1984), and is discussed in detail in (Allen & Ferguson 1994). Time intervals are the basic units, which are correlated by relations like BEFORE or OVERLAPS. (Freksa 1992) revises this approach by semi-intervals, where the relations are defined between primitives, which determine the intervals' start and end.

An example of a first-class application domain of the interval approach is the constraint-based scheduling, see e.g. (Goltz & John 1996). In planning the interval-based time representations is used rarely. The ZENO planning system (Penberthy & Weld 1994) is one of few exceptions.

Planning as a Close Relative of Scheduling

Planning models based on the interval-based time representation can be seen as a close relative of typical scheduling models. The assumption of (Kautz & Selman 1992) concerning the differences in the computational nature of planning and scheduling might be caused by their STRIPS-like approach.

Figure 2 shows a simple plan as a Gantt chart, which sketches the relation to scheduling. On one or multiple action resources the action tasks (like EAT APPLE) are placed, where the usual non-overlap constraints have to be ensured. The beginning and the end of these tasks are determined by control variables. By constraints between the action tasks and the state resources with their tasks pre- and post-conditions are maintained.

Each task has a value, which is mapped to a value of the resource's domain. For action resources this mapping is the identity, as the tasks already describe the actions to execute. State resources can have a more complex mapping, whereby the problem of continuous changes can be solved, too. For example a gradient as value of a task can be mapped (e.g. by addition to a default) to the state resource to cause a relative change per time (see HUNGER STATE RESOURCE).

The main difference to scheduling are the **dynamics**, as we do not know in advance how many of which tasks have to be performed. The case of a restriction to a special set of tasks would be comparable to the length bounds of plans in Descartes, Satplan, or Graphplan.

Conclusion

The paper introduced a framework for constraint-based agents, and discussed specific properties of dynamic adaptation, real-time behavior and social abilities. Furthermore a modeling approach for the agent's planning was outlined. Other important aspects of the EXCALIBUR project were omitted, such as agents' learning.

The goal of the EXCALIBUR project is to develop a generic architecture for autonomously operating agents within a complex computer game environment. The main research tasks are the extension of constraint programming to fulfill the requirements of a real-time, dynamic, and distributed environment, and the proper use of learning algorithms. Further information about the project are available at:

`http://www.first.gmd.de/concorde/EXCALIBURhome.html`

References

Agre, P., and Chapman, D. 1987. PENGI: An Implementation of a Theory of Activity. In Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87), 268–272.

Allen, J. F. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26(11): 832–843.

Allen, J. F. 1984. Towards a General Theory of Action and Time. *Artificial Intelligence* 23: 123–154.

Allen, J. F., and Ferguson, G. 1994. Actions and Events in Interval Temporal Logic. *Journal of Logic and Computation* 4(5): 531–579.

Bellicha, A. 1993. Maintenance of Solution in a Dynamic Constraint Satisfaction Problem. In Proceedings of Applications of Artificial Intelligence in Engineering VIII, 261–274.

Bessière, Ch. 1991. Arc-Consistency in Dynamic Constraint Satisfaction Problems. In Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), 221–226.

Blum, A. L., and Furst, M. L. 1997. Fast Planning Through Planning Graph Analysis. *Artificial Intelligence.* To appear.

Codognet, P.; Diaz, D.; and Rossi, F. 1996. Constraint Retraction in FD. In Proceedings of the FST&TCS 16.

Davenport, A.; Tsang, E.; Wang, Ch. W.; and Zhu, K. 1994. GENET: A Connectionist Architecture for Solving Constraint Satisfaction Problems by Iterative Improvement. In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), 325–330.

Dean, T., and Boddy, M. 1988. An Analysis of Time-Dependent Planning. In Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88), 49–54.

Dechter, R., and Dechter, A. 1988. Belief Maintenance in Dynamic Constraint Networks. In Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88), 37–42.

Eaton, P. S.; and Freuder, E. C. 1996. Agent Cooperation Can Compensate For Agent Ignorance In Constraint Satisfaction. In Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-96) Workshop on Agent Modeling, 1996.

Fikes, R. E., and Nilsson, N. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence* 5(2): 189–208.

Freksa, C. 1992. Temporal Reasoning Based on Semi-Intervals. *Artificial Intelligence* 54: 199–227.

Georgeff, M. P., and Lansky, A. L. 1987. Reactive Reasoning and Planning. In Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87), 677–682.

Goltz, H.-J., John, U. 1996. Methods for Solving Practical Problems of Job-Shop Scheduling Modelled in CLP(FD). In Proceedings of the Conference on Practical Application of Constraint Technology (PACT'96), 73–92.

Henz, M.; Smolka, G.; and Würtz, J. 1993. Oz — A Programming Language for Multi-Agent Systems. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93), 404–409.

Jacobi, S., and Hower, W. 1993. Modifying a Constraint Network. In Proceedings of the International Workshop at the Congress on Computer Systems and Applied Mathematics (CSAM'93), 223–236.

Jiang, Y.; Richards, T.; and Richards, B. 1994. No-Good Backmarking with Min-Conflict Repair in Constraint Satisfaction and Optimization. In Proceedings of the Second Workshop on the Principles and Practice of Constraint Programming (PPCP'94), 36–48.

Joslin, D., and Pollack, M. E. 1996. Is "early commitment" in plan generation ever a good idea? In Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), 1188-1193.

Kautz, H., and Selman, B. 1992. Planning as Satisfiability. In Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI-92).

Kautz, H., and Selman, B. 1996. Pushing the Envelope: Planning, Propositional Logic, and Stochastic Search. In Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), 1194–1201.

Levesque, H. J.; Cohen, P. R.; and Nunes, J. H. T. 1990. On Acting Together. In Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90), 94–99.

Liu, J., and Sycara, K. P. 1993. Emergent Constraint Satisfaction through Multi-Agent Coordinated Interaction. In Proceedings of the Fifth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-93).

Luo, Q. Y.; Hendry, P. G.; and Buchanan, J. T. 1992. A New Algorithm for Dynamic Distributed Constraint Satisfaction Problems, Technical Report, RR–92–63. University of Strathclyde, Glasgow.

McCarthy, J., and Hayes, P. J. 1969. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In Meltzer, B., and Mitchie, D. (eds.), *Machine Intelligence 4*, Edinburgh University Press.

Minton, S.; Johnston, M. D.; Philips, A. B.; and Laird, P. 1992 . Minimizing Conflicts: a Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems. *Artificial Intelligence* 58: 161–205.

Morris, P. 1993. The Breakout Method for Escaping from Local Minima. In Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93), 40–45.

Penberthy, J. S., and Weld, D. S. 1992. UCPOP: A Sound, Complete, Partial Order Planner for ADL. In Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning, 102–114.

Penberthy, J. S., and Weld, D. S. 1994. Temporal Planning with Continuous Change. In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), 1010–1015.

Pesant, G., and Gendreau, M. 1996. A View of Local Search in Constraint Programming. In Proceedings of the Second International Conference on Principles and Practice of Constraint Programming (CP96), 353–366.

Ramalingam, G., and Reps, Th. 1993. A Categorized Bibliography on Incremental Computation. In Proceedings of the 20th annual ACM Symposium on Principles of Programming Languages, 502–510.

Rueher, M. 1995. An Architecture for Cooperating Constraint Solvers on Reals. In Podelski, A. (ed.), *Constraint Programming: Basics and Trends*, Springer LNCS.

Sannella, M. 1993. The SkyBlue Constraint Solver. Technical Report 92-07-02. Dept. of Computer Science and Engineering, University of Washington.

Selman, B.; Levesque, H.; and Mitchell, D. 1992. A New Method for Solving Hard Satisfiability Problems. In Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92), 440–446.

Solotorevsky, G.; Gudes, E.; and Meisels, A. 1996. Modeling and Solving Distributed Constraint Satisfaction Problems (DCSPs). In Proceedings of the Second International Conference on Principles and Practice of Constraint Programming (CP96), 561–562

Sycara, K.; Roth, S.; Sadeh, N.; and Fox, M. 1991. Distributed Constrained Heuristic Search. *IEEE Transactions on Systems, Man, and Cybernetics* 21(6): 1446–1461.

Tidhar, G.; Rao, A.; Ljungberg, M.; Kinny, D.; and Sonenberg, E. 1992. Skills and Capabilities in Real-Time Team Formation, Technical Report, 27. Australian Artificial Intelligence Institute, Melbourne, Australia.

Veloso, M.; Carbonell, J.; Pérez, A.; Borrajo, D.; Fink, E.; and Blythe, J. 1995. Integrating Planning and Learning: The PRODIGY Architecture. *Journal of Experimental and Theoretical Artificial Intelligence* 7(1).

Verfaillie, G., and Schiex, Th. 1994. Solution Reuse in Dynamic Constraint Satisfaction Problems. In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), 307–312.

Voudouris, Ch., and Tsang, E. 1996. Partial Constraint Satisfaction Problems and Guided Local Search. In Proceedings of Practical Application of Constraint Technology (PACT'96), 337–356.

Wallace, R. J., and Freuder, E. C. 1996. Anytime Algorithms for Constraint Satisfaction and SAT problems. *SIGART Bulletin* 7(2).

Wooldridge, M., and Jennings, N. R. 1995. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review* 10(2): 115–152.

Wooldridge, M., and Jennings, N. R. 1996. Towards a Theory of Cooperative Problem Solving. Proceedings of the Sixth European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-94), 15–26.

Yokoo, M.; Ishida, T.; and Kuwabara, K. 1990. Distributed Constraint Satisfaction for DAI Problems. The 10th International Workshop for DAI, 1990.

Yokoo, M. 1993. Constraint Relaxation in Distributed Constraint Satisfaction Problems. In Proceedings of the Fifth IEEE International Conference on Tools with Artificial Intelligence, 56–63.

Yokoo, M. 1994. Weak-commitment Search for Solving Constraint Satisfaction Problems. In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), 313–318.

Yokoo, M., and Hirayama, K. 1996. Distributed Breakout Algorithm for Solving Distributed Constraint Satisfaction Problems. In Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96).

Zhang, Y., and Mackworth, A. K. 1993. Parallel and Distributed Finite Constraint Satisfaction: Complexity, Algorithms and Experiments. In Kanal, L.; Kumar, V.; Kitano, H.; and Suttner, C. (eds.), *Parallel Processing for Artificial Intelligence*, Elsevier Science Publishers.

Zilberstein, S., and Russell, S. 1995. Approximate Reasoning Using Anytime Algorithms. In Natarajan, S. (ed.), *Imprecise and Approximate Computation*, Kluwer Academic Publishers.