Waffler: A Constraint-Directed Approach to Intelligent Agent Design

John Anderson
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2
andersj@cs.umanitoba.ca

Abstract

One of the most significant applications of constraint-directed reasoning to the design of intelligent agents is its extremely broad and general mechanisms for knowledge representation. This paper explores the use of constraint-directed reasoning as a foundation for real-time intelligent agents in complex domains, and describes an architecture that employs constraint-directed reasoning to deal with both high and low-level aspects of agent functionality. It represents both a summary of work performed to date, as well as work in progress.

1. Introduction

Constraint-directed reasoning has emerged as a powerful problem-solving paradigm used widely in artificial intelligence applications. While much work in this area lies in developing formalisms for representing constraints and algorithms for processing constraints, the applications-oriented study of constraint-directed reasoning has also proven to be very fruitful, both to the further understanding of constraint-directed reasoning mechanisms as well as to the application areas involved. The foremost example of this is Fox's [1983] well-known work on constraint-directed scheduling, but many other examples also exist, such as the application of constraint-directed reasoning to classical planning [Stefik, 1981] and to multi-agent planning and coordination [Evans et al., 1992]. The work described in this paper (parts of which are described in greater detail elsewhere [Anderson, 1995; Anderson and Evans, 1996a, 1996b]) involves the use of constraint-directed reasoning as a foundation for real-time intelligent agents in complex domains.

We have developed and are currently extending a constraint-directed approach to real-time reaction and deliberation in complex domains based on improvisation: the real-time integration of knowledge of the agent's immediate situation along with common routines representing previous experience. This approach is embodied in an agent architecture known as *Waffler*, and allows an agent to react to its environment immediately (following both local and global goals), as well as to devote as much deliberative activity as time permits and even more importantly, as knowledge of the particular problem-solving situation deems warranted. Both the basic knowledge representation mechanisms used by Waffler agents and the agents' own internal control mechanisms have a strong basis in constraint-directed reasoning.

2. Improvisation and Waffler Agents

The behaviour of a Waffler agent is based on the observed behaviour of humans in dealing with the extensive variability that occurs during the course of day-to-day activities through *improvisation*. Improvisation is observable in the bulk of human activities [Anderson, 1995; Hodgson and Richards, 1974; Jencks and Silver, 1972], and allows humans to respond flexibly and creatively by employing compiled routines supplemented with more general background knowledge in real time. In order to apply a mundane routine effectively and flexibly in the face of significant variation and interaction, humans possess a vast collection of more general knowledge that allows us to integrate alternatives seamlessly with our routine. We can divert from our routine when it makes sense to do so, and return to it without any great effort. We can also use this same background knowledge in conjunction with our usual routine to cope with even greater degrees of variation (e.g. significant differences in setting or available tools). In short, the compiled plan knowledge formed by our previous experience with a particular activity represents a resource that is relied upon to reduce the intellectual effort that would normally be associated with the activity, in order to perform in a timely manner. We rely on this resource strongly in cases where the current situation follows our previous experience. In situations where our previous experience differs, we can use associated knowledge to improvise on our previous experience to the degree that the situation warrants. A Waffler agent duplicates these abilities.

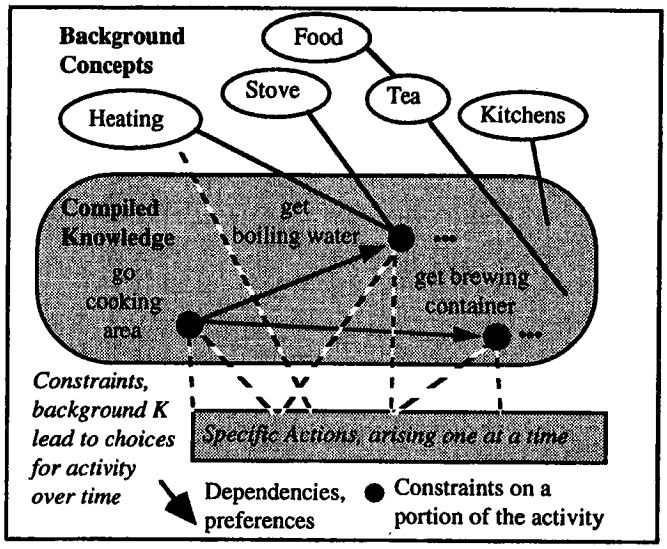

Figure 1. Example of an Intention.

A Waffler agent's compiled plan knowledge and background knowledge are incorporated into distributed constraint-based knowledge structures known as *intentions*. An example of a simple intention used by a Waffler agent to make tea in a dynamic simulated kitchen is shown in Figure 1. Any intention consists partly of the agent's routine knowledge of the activity; that is, a general description of how a normative instance of the activity should proceed. This description is made up of the constraints that the activity places on the agent's behaviour (for example, preferences for actions or further intentions, or requirements for the activity). An intention also contains links to the background knowledge (organized as a network) out of which the agent's compiled routine has evolved. This knowledge also consists largely of constraints, and represents knowledge behind the preferences and other constraints that make up the agent's routine. This division allows the agent's routine to make suggestions for activity immediately where appropriate, and background information to do the same using a search process whose length will vary depending on how closely associated particular pieces of background knowledge are to the agent's routine. This in turn allows the agent access to immediate responses that are useful in the typical case of the activity, and the ability to search and deliberate as time and the significance of the situation permit.

Constraints from both the intention itself and the background knowledge associated with it (to the degree the agent chooses to explore the latter) direct the agent toward individual actions over time. Actions may include adopting further intentions, changing the milieu of constraints that form a basis for the agent's moment-by-moment choices for action. A series of actions thus emerges over time as a result of the initial adoption of an intention in conjunction with others adopted at the same time and independent events that occur in the environment as the intention unfolds.

3. Constraints Employed in Waffler Agents

Constraints in intentions represent regularities in the world around the agent and the influence of those regularities on agent behaviour. Despite the wide range of knowledge being represented, we have found as have others (e.g. [Fox, 1983; Evans et al., 1992]) that only a reasonably small number of distinct types of constraints are needed (Figure 2). Constraints are used in this approach to represent a broad range of concepts, from physical restrictions (*physical* constraints) and relationships between entities (*expectation*, *requirement*, and *temporal* constraints) to abstract policies (*behavioural goals* and *preference* constraints), to representing normative responses to particular situations and control of internal agent components (*normative* and *focus* constraints) [Anderson, 1995]. Each of these types has similar components, including a specified active lifetime, activation requirements, and an attachment to some larger knowledge structure. Constraints are organized in a loose hierarchy, abstracting both knowledge and control, and perform different functions depending on the level at which they are defined.

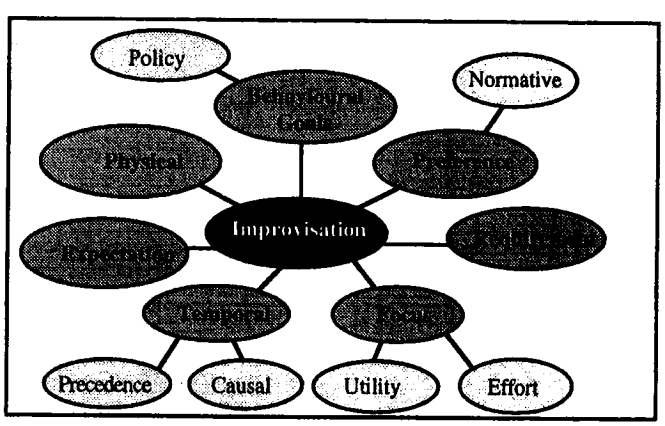

Figure 2. Categorization of constraints.

4. Constraints and Internal Agent Mechanisms

The use of constraints as the primary knowledge representation mechanism directly supports the ability of a Waffler agent to perform in real time and to perform in a satisficing manner with the knowledge at its disposal. In an intention such as the one shown in Figure 1, for example, the compiled routine may contain among other things a constraint indicating that the agent should prefer working with an electric kettle when boiling water as opposed to some other tool. This constraint expresses a preference that is normally applied in the course of the activity with no exploration as to the reasons behind the preference. When this tool is unavailable or the agent wishes to reason beyond the routine (due to error, knowledge of potential error, or to high-level constraints that affect how an agent performs an activity), the agent can make use of further constraints behind that preference (background knowledge) that describe the role and function of the kettle in the overall routine. The agent can then use those constraints as a basis for reasoning about alternative ways of performing the activity, to the degree the agent wishes to devote intellectual effort to this. For example, constraints about heating water will lead the agent to a set of objects with characteristics suitable for this purpose.

Because constraints are modular, constraints external to an intention can also have immediate effects on it. The presence of a constraint such as hurrying (brought on by a combination of intentions or some external event) may affect it in certain predictable ways that are part of the routine itself. That is, the presence of a *hurry* constraint from outside the intention may allow certain routine components to become active that would be otherwise ignored. Such a constraint will also affect the agent itself: how much information the agent considers, strategies for deliberation, etc.

In any significant domain, there will clearly be a large number of potentially relevant constraints available at any point in time. However, the agent's cognitive effort in this approach is for the most part not spent on looking for constraint violations, as in most constraint directed reasoning systems. While we are concerned about violations in some cases (e.g. expectations), here most constraints act positively: their presence compels the agent toward or away from specific courses of reasoning or activity, just as the landscape influences the direction of one's travel. The key to real-time performance is the *selective* processing of constraints in order to make satisficing decisions in the time available. This is done through the multi-level organization of constraints in intentions, in tandem with facilities provided in the Waffler architecture for limiting the number of constraints considered.

An overview of the Waffler architecture is shown in Figure 3, and a summary of the use of constraints in these processing mechanisms appears in Figure 4. The agent itself is divided into several computational processes and two major stores of knowledge. The agent's *long-term memory* contains all the possible intention and conceptual knowledge possessed by the agent. The central role in this architecture, however, is played by the agent's *working memory*, which directly supports the selective recall and processing of constraint knowledge over time. Working memory is of limited size, and represents the amount of information the agent can process in parallel (in our implementation a timeshared simulator is used, and the size of working memory represents the amount of information the agent can process in a time-slice). Any constraint in working memory is thus viewed as having immediate effects on reasoning or behaviour, and relations among concepts in working memory are assumed to be immediately realizable (via direct memory connections). These are not unrealistic assumptions, since the number of concepts or intentions physically allowed in working memory at the same time can always be set to a small enough limit to make this so.

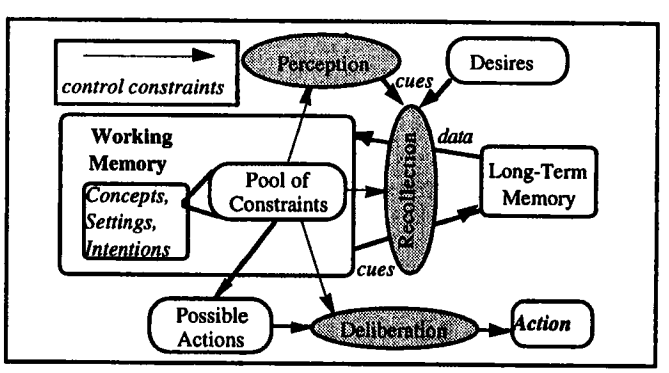

Figure 3. The Waffler Architecture

Items migrate from long-term to working memory through the use of memory triggers. Each object, concept, or intention may have simple trigger conditions associated with it that, when satisfied, allow this item to be brought into working memory. Trigger conditions may be satisfied

through perception, through new desires, or through objects already present in working memory. Trigger conditions are only considered when present in working memory, and are brought into working memory when adjacent or connected concepts are present in working memory. This directly supports the ability to gradually move to more detailed background knowledge over time.

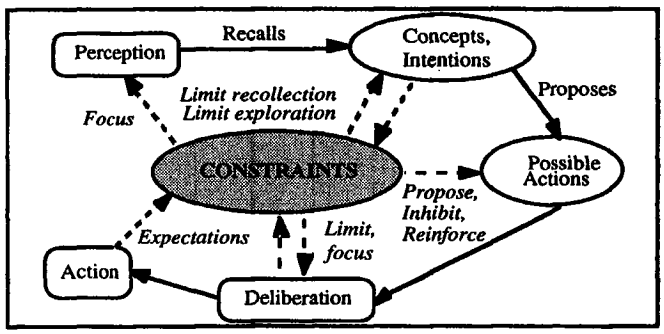

Figure 4. Constraints in the Waffler Architecture.

At any point, the collection of constraints in the agent's working memory will give rise to a particular set of alternatives for action, which the agent can deliberate over. The more constraints available to working memory, the fewer alternatives need be deliberated upon, and hence deliberation is secondary to working memory in this approach. However, because working memory is very limited, the agent will often have only incomplete picture of its overall choices and have to decide if further exploration is warranted, and so deliberation is still significant. Deliberation in the Waffler architecture is dealt with using constraints on the amount of cognitive effort that can be directed toward a particular end. This illustrates the most significant aspect of the Waffler architecture: not only do constraints represent the structure of an agent's activities, they also control the architecture itself.

Individual constraints in working memory influence the agent toward (or away from) some alternative for action through a quantitative or qualitative measure of utility. This measure is composed of an innate estimate of the constraint's importance, modified by the importance of the chain of intentions through which the constraint has come into working memory and by specific conditions the constraint is concerned with. At any time, a given number of constraints (those that have passed through working memory) will have contributed their utility to particular alternatives, and a given number (those attached to relevant concepts not in working memory) will have not have contributed. In order to make decisions using incomplete information such as this, the agent has in its working memory a constraint on utility, representing the minimum utility necessary for an alternative to be acted upon. This constraint may be global or local to a particular activity the agent is performing, and serves to limit deliberation by allowing the agent to select the most important alternatives. It is not static, and can be affected by intentions and concepts in working memory as well as higher-level knowledge. For example, when conflicting intentions are present, high-level knowledge may alter the agent's utility constraint to be higher in order that the agent have greater confidence that it is selecting the right action in these situations. If no action can be performed, the agent may wait for more information to be processed through working memory. This affords the possibility of a higher utility for one or more alternatives, or for constraints in working memory to alter the agent's utility constraint, allowing less highly-ranked alternatives to be selected. It also presents the possibility that the agent could miss some time-constrained opportunity. Utility may also be used in conjunction with activity specific measures, as described in [Anderson, 1995; Anderson and Evans, 1996a]

Constraints are also used to control many other aspects of this architecture, such as memory management. The most appropriate memory management policy varies with the situation (e.g. if the agent is in a time-constrained situation with regard to some task, working memory should be biased toward concepts contributing to this task). Constraints associated with intentions may suggest an appropriate policy, or more likely, higher level constraints (associated with the setting or more general background knowledge) will recognize when specific policies are required. Constraints also define a perceptual focus for the agent, and can also focus the agent toward retrieving particular concepts from long-term memory and thus follow a particular line of reasoning.

5. Summary

While this work represents a practical approach to real-time planning in broad domains, it also illustrates the power and flexibility of constraints as a knowledge representation mechanism for agents in such domains. Constraints play several major roles in real-time activity, all of which are encompassed by this approach:

- Constraints serve as *inhibitors of activity*:, by inhibiting the agent from working with certain objects, performing certain actions, or participating in certain types of activities. This inhibition may be in the form of a complete debarral, a quantitative rating on some scale, or a qualitative measurement. For example, a social setting imposes many constraints barring or inhibiting an agent from performing actions that might be perfectly acceptable were the agent alone.

- In the same manner, constraints also serve to *compel* the agent toward performing certain actions, or to expend extra effort toward a given activity.

- Constraints *represent interactions* between actions or activities, or between an action or activity and some part of the world around the agent.

- Constraints serve to *control the amount of physical effort* spent on an activity, the amount of cognitive effort spent in making a decision, or the amount of consideration given to other constraints. For example, the agent may restrict the amount of time spent deliberating when a certain type of action is involved, or the amount of time spent deliberating toward a certain activity.

- Constraints also serve to *represent the agent's expectations of the future*. When an agent performs an action or witnesses an event, for example, the expectations the agent makes of the action or event is used to base future activity. These expectations can be represented as restrictions on a future world: restrictions whose violation may lead to significant changes to the agent's intended activities.

- Constraints serve to *limit the extent of improvisation*, controlling the amount of effort spent on finding some improvised alternative during the course of an activity. Improvisation can range from fairly mundane variations on a particular activity to more extensive jumps of imagination. This can result in great variation both in the amount of effort put toward an activity and the propriety of the eventual solution. Restrictions on these aspects can be used to control how far along these dimensions improvisation can proceed.

- Constraints serve to *modify an agent's routines* based on current conditions within the environment. A routine activity such as grocery shopping can be greatly modified by the addition of a constraint that it must be done quickly, for example.

- Constraints serve to *limit the recollection* of agents and the amount of information that is examined during the course of activity.

We have implemented this architecture using the Gensim simulation testbed [Anderson and Evans, 1995] to provide a simulated world for the agent (consisting of five time-shared computational processes) to inhabit. The current environment is a simulated kitchen, in which accidents and other unexpected timely events can occur in the midst of the agent's ongoing activities (tasks such as making tea, answering the telephone, cleaning up, and other activities that normally take place in a kitchen. This implementation has illustrated the abilities of this architecture to cope with unexpected events (errors, competing intentions) in real time during the course of an agent's activities. The implementation also demonstrates the flexibility of this approach in situations where the agent's environment differs from that expected by its routine. Implemented examples illustrate that when tools are made unavailable, the agent can devote more time to reasoning about substitutions and increasingly novel methods of improvising on its routine as time and knowledge allows. All of this flexibility is due directly to the constraint-based representation and reasoning mechanisms employed by the architecture.

We are currently extending this work to involve *multi-agent improvisation*: [Anderson, 1997] the real-time application of constraint-directed routines that have explicitly separated individual and social levels, allowing individual agents to improvise within mutually acceptable (but varying) limits. At a surface level, this architecture is most immediately compared to that of Bratman et al [1988] in its use of intentions and in the ability to derive new options and deliberate about them. There are several significant differences. First, intentions in the Waffler architecture are much more sophisticated than those used by Bratman et al. They contribute alternatives for action and connections to background knowledge that can indirectly supply alternatives, as opposed to influencing activity through a vague sense of commitment to the intention itself. This architecture also provides specific methods for generating and dealing with options as a core of the architecture itself, rather than leaving these and other components as implementation-dependent details. Finally, relying heavily on constraints as a knowledge

representation mechanism allows an agent to reason as deeply or shallowly about a particular situation as time allows.

Further papers, including the references to my work below, may be obtained on the WWW at the URL: *http://www.cs.umanitoba.ca/~andersj/research.html.*

References

[Anderson, 1997] Anderson, John, "Supporting Intelligent Agents in Individual-Based Ecosystem Models", *Proceedings of the Eleventh Annual Conference on Geographic Information Systems* (GIS-97), Vancouver, Canada, February, 1997, pp. 3-6.

[Anderson, 1995] Anderson, John, *Constraint-Directed Improvisation for Everyday Activities*, Ph.D. Dissertation, Department of Computer Science, University of Manitoba, March, 1995. 389 pp.

[Anderson and Evans, 1996a] Anderson, John, and Mark Evans, "Real-Time Satisficing Agents for Complex Domains", *Proceedings of the Ninth Florida AI Symposium*, Key West, May, 1996

[Anderson and Evans, 1996b] Anderson, John, and Mark Evans, "Constraint-Directed Improvisation", *Proceedings of the Eleventh Biennial Canadian Society for the Computational Studies of Intelligence Conference (AI-96)*, Toronto, May, 1996

[Anderson and Evans, 1995] Anderson, John, and Mark Evans, "A Generic Simulation System for Intelligent Agent Designs", *Applied Artificial Intelligence* 9:5, 1995, pp. 527-562.

[Bratman et al., 1988] Bratman, Michael E., David J. Israel, and Martha E. Pollack, Plans and Resource-Bounded Practical Reasoning, *Computational Intelligence*, Vol. 4, 1988, pp. 349-355.

[Evans et al., 1992] Evans, Mark, John Anderson, and Geoff Crysdale, "Achieving Flexible Autonomy in Multi-Agent Systems using Constraints", *Applied Artificial Intelligence* 6:1, 1992, pp. 103-126.

[Fox, 1983] Fox, Mark S., *Constraint-Directed Search: A Case Study of Job Shop Scheduling*. Ph.D. Dissertation, School of Computer Science, Carnegie-Mellon University, 1983. 184 pp.

[Hodgson and Richards, 1974] Hodgson, John, and Ernest Richards, *Improvisation* (London: Eyre Methuen Ltd.), 1974. 209 pp.

[Jencks and Silver, 1972] Jencks, Charles, and Nathan Silver, *Adhocism: The Case for Improvisation* (New York: Doubleday), 1972. 216 pp.

[Stefik, 1981] Stefik, Mark, "Planning with Constraints", *Artificial Intelligence* 16:2, 1981, pp. 111-140.