Hierarchical Conflict Management

David Goldstein
North Carolina Agricultural & Technical State University
Greeensboro, NC 27411
USA
Voice: (910) 334-7245
Fax: (910) 334-7244
goldstn@ncat.edu

Keywords: Real-time reasoning, Assumption-based reasoning, Hierarchical, Conflict resolution

Abstract

Two distinct concerns seem apparent in conflict management strategies, one reflecting real-time decision-making and the other constrained deliberation. This paper attempts to provide more detail concerning our experiments in this area and provide a more unified model of conflict resolution as the management of resources in constrained environments.

This paper is structured in the following manner. First, we describe the scope of the problems which interest us. We then describe our approaches to address these problems. Finally, we describe the implementation of our research and our goal of developing mechanisms applicable to resolving conflicting constraints in a hierarchy of decision makers.

Introduction

We have been examining conflict resolution (1) in real-time forward-chaining systems at the rule level and (2) conflict management with respect to the hypotheses proposed by interacting agents. While these two areas initially seem disparate, they both consider choosing between multiple alternatives based upon constraints. Further, the more information added to either system, the better the methods should be to perform. This paper considers, at a macro level, the questions: (1) what concepts are common to these two approaches and (2) can these commonalties lead to a general mechanism for addressing conflict resolution in a hierarchical manner.

Our Approaches

Our initial research stemmed from real-time knowledge-based systems. Initially, we attempted to devise clever mechanisms for "conflict resolution" to determine the most appropriate chain of reasoning to pursue among the multiple chains currently available to a forward-chaining system. More specifically, given n satisfied rules in the "conflict set" of a forward-chaining reasoning system, how might we choose the most appropriate rule for execution. Commonly used approaches, including recency and means-ends analysis, essentially ignore the real-time characteristics of problems.

Our approach to conflict resolution modifies the priorities of individual agenda items (i.e., rules and their matching data elements) during each reasoning cycle. We prioritize rules based upon the likelihood of the actions helping to meet an application's deadlines. More specifically, we develop a partial ordering of actions reflecting a partial ordering of rule priorities. We then use these relative rule priorities, along with resource dependencies, to develop time-varying priority functions. These functions are then converted into spline functions for efficient computation (Figure 1). We have modified our inferencing mechanisms to efficiently compute these priorities at run-time [1].

We examine employing such a conflict resolution algorithm by considering the generation of plausible courses of action and selecting one such course prior to some time t. We discuss determining the parameters driving the dynamic priorities in [1]. We intend to eventually also consider how much reasoning must take place (measured by counting and weighing the tokens in the RETE network) to modify the rule priorities. By tying the contents of the RETE network to the rule priorities, we hope to be able to extend an approach such as that in [2] to predict near-future behavior of a system.

The above approach seemingly addresses only the lowest levels of conflict management in a system. Therefore, we now address developing mechanisms for conflict management among multiple, interacting agents.

The mechanisms we have previously used for conflict management were rather autocratic [3]. However, our current experiments in Distributed Artificial Intelligence deal with the extreme of distributed control (and so we are devising new strategies for conflict management); agents can reason about other agents if their goals require such knowledge, but if an agent G's goals are disjoint from other agents then processing can proceed (with literally hundreds of agents being dynamically created or destroyed) without any other agent having knowledge of G. While numerous conflict management models have been proposed reflecting negotiation, stock markets, game-playing, etc., the approach we are exploring is based upon transferring procedural knowledge and assumptions among agents.

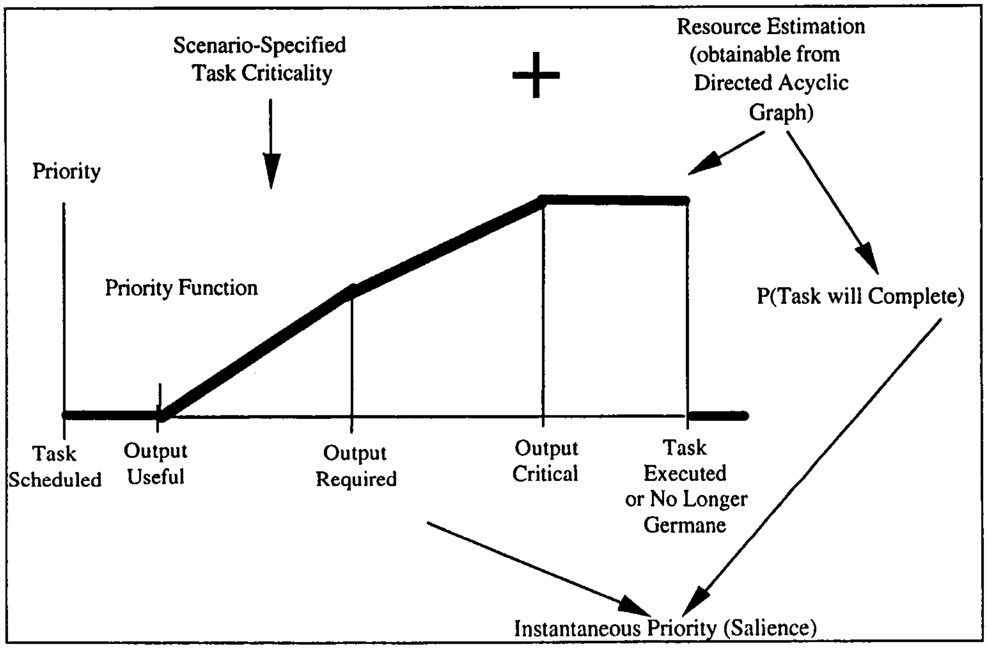

Figure 1: Prioritizing a Rule via a Spline Function

We view agents as a collection of entities engaged in a dialogue in a canonical language. Periodically, contradictory statements (plans) are stated (created). Based upon the actual world, at least one statement from the set of all statements involved in the conflicting set of statements must be retracted. We use the assumptions to determine which piece of the procedural knowledge should be invalidated first.

Given a set of resources with some value per unit and a set of prioritized goals, procedural knowledge has a cost associated with achieving each specific goal. Each goal similarly has an associated, computable benefit. The difference between these two quantities can be viewed as a net gain in potential resources for having achieved a goal (note: one should not be able to utilize negative resources nor do we allow implicit borrowing of resources.)

The net potential resource gains form an ordering of procedural knowledge to be executed. However, given that certain constraints must be met for this knowledge to be executed, our conflict management mechanism removes the low-priority knowledge from consideration and the corresponding constraints and constraining assumptions. We repeat this process until no conflicts are present. We then choose the remaining statements (decide in favor of the remaining agent's courses of action.) We view removing assumptions as taking one of two forms, either (1) literally discarding the hypotheses related to the low priority assumptions or (2) generating alternative hypotheses via employing domain-independent mechanisms to achieve alternative plans as per [4]. While we have not yet implemented our library to generate alternatives to inconsistent, proposed plans, we hope to develop a domain-independent rule-base to map domain-specific concepts and classes of objects to generally applicable techniques.

These two conflict resolution strategies may seem disparate initially, but they do have some common roots; they both examine resource conflicts using rule-based techniques. As we develop our domain independent techniques for conflict resolution, we hope to better understand the conflict resolution process so as to perform agenda management in a more efficient manner. Further, we hope to eventually be able to develop a hierarchy of conflict management schemes from the inner-agent to society-wide levels such that one can model the resolution processes using similar mathematics at each level.

Status

The agent architecture has been realized as the Distributed Artificial Intelligence Toolkit (DAIT), a series of tools including an enhanced version of NASA's C Language Integrated Production System (CLIPS) [5][6]. Agents can communicate via facts (mathematical tuples), templates (database records) and objects.

The underlying meta-models (semantics) of data items is an important consideration. We use an extension of the Product Data Exchange Specification (PDES) for our meta-models [7]. Our meta-models exhibit our philosophy that basic concepts understood by each application in a system must first be described. Because PDES includes representations of quantities (such as force, distance, and hue) diverse concepts (such as behaviors, phenomena, and physical objects) can be represented.

Conclusion

We have presented two conflict management schemes operating at different levels in a hierarchy of conflict resolution. We intend to model these more rigorously in the future to better understand the essential characteristics of conflict management in a hierarchy of decision makers. We intend to use these schemes to develop domain-independent conflict resolution mechanisms which can be employed at any level in the hierarchy by mapping general mechanisms to domain-specific objects and concepts.

References

[1] D. Goldstein, "A Fault-tolerant Agent-based Architecture for Manufacturing", Concurrent Engineering: Research and Applications, to appear 1994.

[2] F. Barachini and H. Mistelberger, "Run-Time Prediction for Production Systems", in Proceedings of the Tenth National Conference on Artificial Intelligence, American Association for Artificial Intelligence, Menlo Park, CA, July 1992, pp. 478 - 485.

[3] D. Goldstein and J. Tiernan, "Domain-independent Conflict Management via Agent-supplied Metrics", 1993 IJCAI Workshop on Conflict Management, International Joint Conference on Artificial Intelligence, France, September 1993.

[4] M. Klein, "Supporting Conflict Resolution in Cooperative Design Systems", in Proceedings of the 10th International Workshop on Distributed Artificial Intelligence, Bandera, TX, October 1990.

[5] J. Giarrantano, "CLIPS User's Guide", NASA/Johnson Space Center, Houston, USA, 1991

[6] D. Goldstein, "The Distributed Artificial Intelligence Toolkit", AI Expert, Miller-Freeman Publishing, January, 1994, pp 30-34.

[7] International Standards Organization, Product Data Exchange Specification, Part 32, National Institute of Standards and Technology.