Focus of Control Through Goal Relationships

Victor R. Lesser, Daniel D. Corkill, Robert C. Whitehair and Joseph A. Hernandez
Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01003

Abstract

Goal relationships resulting from the initial data and subsequent processing can he used to dynamically construct a partial topology of the solution space based on what appear to be feasible solutions. This structure can be used to make control decisions that significantly reduce the amount of search required to solve a problem in a complex domain. We examine the utility of this approach in the context of a multi-level, cooperative knowledge source model of problem solving. We present a taxonomy of goal relationships for constructing partial topologies of the solution space and show that mechanisms using this information can be built as natural extensions of an integrated data-directed and goal directed archi tecture. Examples and performance results demonstrating how these additions improve the system's ability to evaluate potential activities are provided.

1 Introduction

Making appropriate control decisions in a complex, multi-dimensional search space is a difficult task. This difficulty arises because relationships among partial results and potential activities are not readily observable. For example, in a blackboard architecture the same partial results can be used in many contexts. Therefore, producing a specific result may affect many alternative solutions. In addition, the problem space is represented at multiple abstraction levels on the blackboard, and multiple solution paths for the same result may be available. This provides the problem solver with flexibility in choosing problem-solving activities but also allows results to be rederived using alternative paths without recognizing the redundancy until the last step. Furthermore, the asynchronous, opportunis tic style of problem solving leads to situations where it is unclear whether a solution is missing due to a lack of data, in which case the solution will never be generated, or due to a lack of processing, in which case additional work will eventually produce the solution. Thus, employing complex problem-solving capabilities while at the same time making intelligent control decisions is a formidable task (Hayes-Roth, 1985, Hayes-Roth and Lesser, 1977, Lesser and Erman, 1977, Nii, 1986b, Nii, 1986a].

Several years ago, the cooperating knowledge source architecture of Hearsay-II [Erman et a/., 1980] was extended to unify data-directed and goal-directed control [Corkill et a/., 1982].[1] This was a first step toward developing the needed interrelationships among actions and results necessary for making intelligent control decisions. In the interim, we have gained considerable experience with this control architecture. In particular, we have identified the need for new types of goals and for additional relationships among goals. These extensions allow us to more accurately relate the predicted results of future activities to existing results in order to make more informed control decisions.

In the next section we briefly review the unified data-directed and goal-directed control architecture. Section 3 describes the new goal relationships from a general perspective and presents specific examples from two domains. Section 4 outlines how these new mechanisms work in a blackboard based problem solver. Section 5 is a brief presentation and discussion of our experimental results.

2 A Review of Goal-based Control

Figure 1 presents a high level schematic for the integrated data directed and goal-directed control architecture as implemented in the DVMT [Lesser *et a/.*, 1987, Lesser and Corkill, 1983]. The basic Hearsav-II architecture is modified to include a goal blackboard and a goal processor. The goal blackboard, which mirrors the data blackboard in dimensionality, contains goals representing intentions to create particular results on the data blackboard. Goals provide an abstraction over the potential actions for achieving a particular type of

"This research was sponsored, in part, by the National Science Foundation under CER Grant DCR-8500332, and by the Office of Naval Research under University Research Initiative Grant Contract N00014-86-K-0764, and under Contract NO0014-79-C-0439.

[1] Through a .slightly different approach, Johnson and Hayes-Roth [Johnson and Hayes-Roth, 1987] have also extended the data-directed process of the classic blackboard problem solver to include goal-directed control.

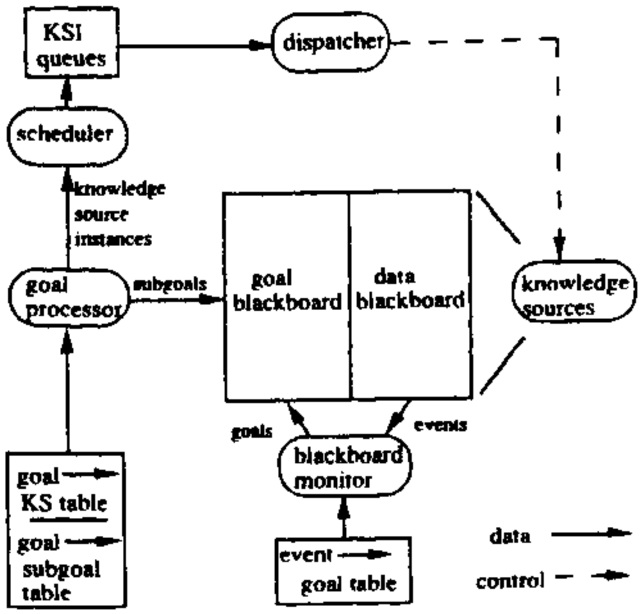


Figure 1: Integrated Data and Goal-Directed Control

result and allow the system to reason about its intentions independently of the particular knowledge source (KS) actions at its disposal.

The two general classes of goals are data-directed and goal-directed. The blackboard monitor uses domain knowledge to create data-directed goals in response to the addition or modification of partial results on the data blackboard. Each data-directed goal specifies the range of potential solutions resulting from the use of the triggering data.

Since the creation of a goal does not guarantee sufficient information on the data blackboard to execute a KS to satisfy the goal, the goal processor runs a precondition procedure for the applicable KSs *to* make a detailed examination. When results indicate that a KS has sufficient information to satisfy the goal, the goal triggers a KS instantiation (KSI). The scheduler assigns the KSI a priority rating and places it on the scheduling queue. The scheduler assigns priority by first determining the set of goals that may be satisfied by a KSI's predicted output. It then computes the KSI's rating as a function of the ratings of the potentially satisfied goals and the credibility of the predicted results. If sufficient information is not available to run a KSI, the goal processor can create goal-directed goals (subgoals) to generate the needed data.[2] The decision to create subgoals is dependent on the importance of achieving the goals potentially satisfied by the KSI and on the cost of creating the subgoals. Subgoal ratings are based on the rating of the goal that triggered the KSI. If the parent goal has a high rating, its subgoals might increase the rating of KSIs that generate data needed to satisfy the preconditions of the KSI triggered by the parent goal.

[2]Goal-directed goals are also generated by the goal processor for other purposes.

Goal-based control does not require that control decisions be made in a top-down, goal-directed manner. Goals are used to make both data-directed and goal-directed control decisions, and the classic data-directed/goal-directed dichotomy is represented in our approach by the relative ratings among goals. By adjusting its KSI rating computations, the scheduler can bias the system towards goal-directed or data-directed control. Goal-based control attempts to incorporate domain data to build an appropriate control abstraction that will predict the type of results that can possibly be generated. This permits the system to develop non-local focus-of-control strategies that take into account the interactions of work on data in different parts of the problem-solving space.

3 Goal Relationships

In addition to the goal/subgoal relationship, other relationships among goals can be exploited to improve the effectiveness of goal-based control. *These new goal relationships resulting from the initial data and subsequent processing can be used to dynamically construct a partial topology of the solution space based on what appear to be feasible solutions.* This structure can be used to make control decisions that significantly reduce the amount of search required to solve a problem in a complex domain. Specifically, three important questions necessary for effective control can be answered by relating goals to each other:

- Can the same results be obtained by working on different goals?
- Will working on two distinct goals generate equivalent partial solutions?
- Will work on a goal differentiate between mutually exclusive solutions?

In order to formally define the goal relationships necessary to answer these questions, we first define the following concepts: the *potential solution set* of a goal and the *component set* of a potential solution.

potential solution set for a goal g: $S(g) = \{x \mid x$ is a potential solution of $g\}$, where x is a potential solution of g if its characteristics lie within a range specified by g. $S(g)$ defines the space of all possible solutions of g. A specific $x \in S(g)$ is only a potential solution since it may not be supported by low-level data or it may not be consistent with the solution to some other goal that interacts with g.

component set of a potential solution x: the set of partial results consistent with some subset of the characteristics of x, $C_s(x) = \{y \mid \forall g, y \in S(g) \Rightarrow x \in S(g) \wedge x \neq y\}$. $y \in C_s(x)$ implies that the characteristics of y are a subset of the characteristics of x and, therefore, that y could possibly be used to derive x.

The goal relationships can now be defined as follows:

(a) *Sample Grammar:*

S → NP aux VP
NP → noun | det noun
VP → verb | verb PP | verb NP
PP → preposition NP

(b) *Sample Component Set:*

C_s("The boy ran up the hill.") = {$NP_{The\ boy}$, det_{The}, $noun_{boy}$, aux_{nil}, $VP_{ran\ up\ the\ hill}$, $verb_{ran}$, $PP_{up\ the\ hill}$, $prep_{up}$, $NP_{the\ hill}$, det_{the}, $noun_{hill}$}

(c) *Competition:*

g_1 : form a PP using $\{\text{"}by\text{"}, w_i, ..., w_j\}$
g_2 : form a PP using $\{\text{"}to\text{"}, w_i, ..., w_j\}$

g_1 and g_2 are competing since there is no goal with a solution x where x includes solutions to g_1 and g_2 as components.

(d) *Assistance:*

g_1 : form an NP using $\{w_i, ..., w_j\}$
g_2 : form a PP using $\{\text{"}to\text{"}, w_i, ..., w_j\}$

Every PP contains an NP, so satisfaction of g_2 implies satisfaction g_1. But an NP does not have to be used in a PP, so g_1 is not subsumed by g_2.

(e) *Subsumption:*

g_1 : form a PP using $\{w_i, ..., w_j\}$
g_2 : form a VP using $\{w_h, ..., w_j\}$

g_2 subsumes g_1 since any PP satisfying g_1 will be included in some VP satisfying g_2.

(f) *Cooperation:*

g_1 : form an AUX using $\{w_i, ..., w_j\}$
g_2 : form a VP using $\{w_k, ..., w_l\}$

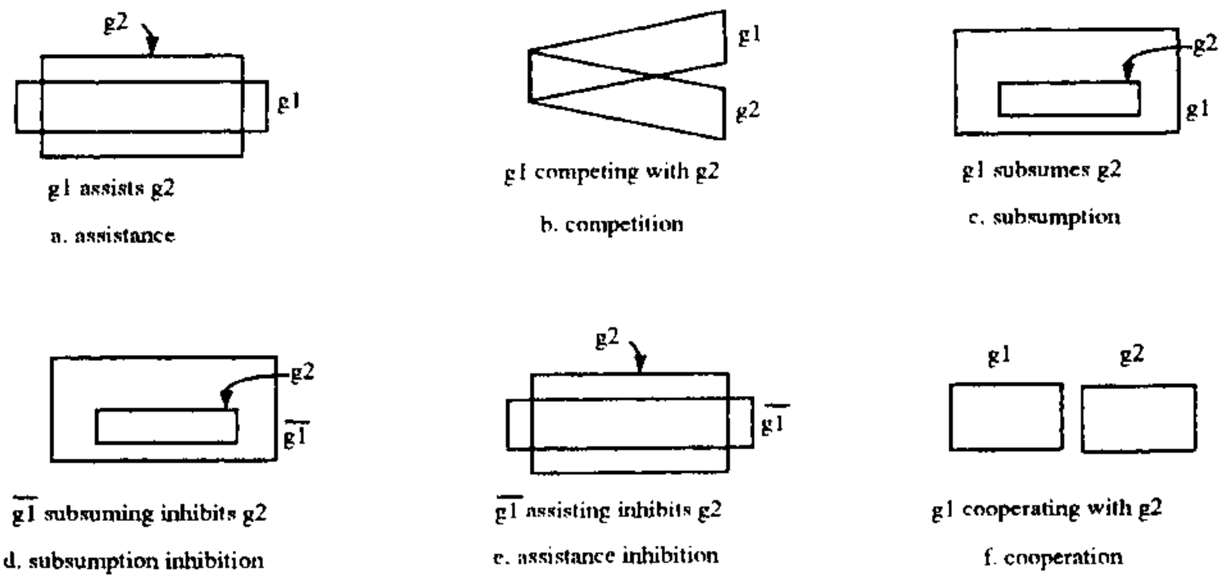
Goal, "g_f : form an S", has solution X where solutions to g_1 and g_2 are components of X, so g_1 and g_2 are cooperating.

(g) *Independence:*

g_1 : form an S using $\{w_i, ..., w_j\}$
g_2 : form an S using $\{w_m, ..., w_n\}$

g_1 and g_2 are not competing and their individual solutions cannot be combined to satisfy another goal.

Figure 2: Goal Relationship Diagrams, Example 1



g1 assists g2
a. assistance

g1 competing with g2
b. competition

g1 subsumes g2
c. subsumption

$\overline{g1}$ subsuming inhibits g2
d. subsumption inhibition

$\overline{g1}$ assisting inhibits g2
e. assistance inhibition

g1 cooperating with g2
f. cooperation

Figure 3: Goal Relationship Diagrams, Example 2

subsumption: Goal g1 subsumes a second goal, g2, if the specifications of g2 are completely encompassed by the specifications of g1. Formally, g_1 subsumes g_2 if $\forall y, y \in S(g_2) \Rightarrow \exists w$ s.t. $y \in C_s(w) \wedge w \in S(g_1)$.

assistance: One goal, g1, is said to assist another goal, g2, if satisfaction of g1 implies satisfaction of g2. The assistance relationship identifies those goals that represent alternative approaches to generating a particular solution. g_1 assists g_2 if $\forall w, w \in S(g_1) \Rightarrow \exists y$ s.t. $y \in C_s(w) \wedge y \in S(g_2)$.

competition: Two goals, g1 and g2, are competing if there is no possible partial solution that will satisfy both goals. By checking if two goals are competing, the system can determine if the knowledge sources they have triggered will generate distinct results. If two goals are not competing, the system cannot make this determination because the possibility exists that a solution to one goal is also a solution to the other goal. As a consequence, working on both goals could produce redundant results. Goals g_1 and g_2 are competing if $\forall w \forall y, w \in S(g_1) \wedge y \in S(g_2) \Rightarrow \neg(\exists g_f \exists W \exists Y$ s.t. $\{W, Y\} \subset S(g_f)$, W and Y can be simultaneously acceptable solutions and $w \in C_s(W) \wedge y \in C_s(Y))$.

cooperation: Two goals are cooperating if it is possible for the goals to produce information that may be incorporated into a single result at some point in the future. Goals g_1 and g_2 are *completely cooperating* if $\forall w \forall y, w \in S(g_1) \wedge y \in S(g_2) \Rightarrow \exists g_f \exists x$ s.t. $x \in S(g_f) \wedge \{w, y\} \subset C_s(x)$.

independence: Two goals are independent if they are not competing and if it is not possible for them to be incorporated into a single solution at some point in the future. Goals g_1 and g_2 are independent if they are not competing and if $\forall w \forall y, w \in S(g_1) \wedge y \in S(g_2) \Rightarrow \neg(\exists g_f \exists x$, s.t. $x \in S(g_f) \wedge \{w, y\} \subset C_s(x))$.

subsumption-inhibition: A new type of goal, called an *inhibiting-goal*, has been added to identify redundant work that can be eliminated. Goal g1 inhibits a second goal, g2, if g1 is an inhibiting goal and g1 subsumes g2.

assistance-inhibition: Goal g1 partially inhibits goal g2 if g1 is an inhibiting-goal and g1 assists g2. The assistance-inhibition relation limits work to those areas of g2 not encompassed by g1.

A sample grammar and example goal relationships for a natural language parsing system that works asynchronously and opportunistically from any point in the input are shown in Figure 2. Preprocessed signal data input is represented as $\{w_1, w_2, ..., w_n\}$.

Figure 3 shows similar goal relationships for a distributed vehicle monitoring system that functions in a spatial domain where a goal can be represented as a two dimensional parallelogram, $[((x_{ll}, y_{ll}), (x_{ul}, y_{ul}))$ $((x_{lr}, y_{lr}), (x_{ur}, y_{ur}))]$. A potential solution of a goal is a sequence of lines connecting consecutive points $((x_a, y_0)...(x_{(a+k)}, y_k)...(x_c, y_{(x_c - x_a)}))$ where at least one point lies within the parallelogram, a line intersects at least one side of the parallelogram, and no line intersects either the top or bottom of the parallelogram.

```
FOR each newly created hypothesis, h
  IF (h.level ≥ *inhibiting-goal-creation-level*) AND
     (h.rating ≥ *inhibiting-goal-creation-threshold*)
  THEN
     Create inhibiting goal G_h and associated
       inhibiting subgoals, SG_{G_h}
     For each triggering goal subsumed by G_h or SG_{G_h}
       Terminate efforts to satisfy the triggering goal
     For each triggering goal assisting G_h or SG_{G_h}
       Restrict processing in areas encompassed by
         G_h or SG_{G_h}
```

Figure 4: Inhibiting Goal Creation Algorithm

4 Experiments

The following two examples demonstrate the usefulness of goal relationships. The first helps prevent redundant processing through the use of inhibiting goals, and the second enables a knowledge source to be rated more accurately based on its local context.

4.1 Inhibiting Goals

A new goal type, an inhibiting goal, has been added to control redundant processing. Without inhibiting goals, the only way to minimize redundant activity was to decrease the ratings of the goal and subgoals that led to a strongly believed result. This method prevents additional work only on the original solution path used to derive the high-level result. It does not limit activity on any of the alternative paths that might lead to the same result. In order to effectively control redundant processing, a separate mechanism is needed to eliminate derivation of *any* intermediate result that would eventually produce the high-level result. Thus, when a satisfactory, high-level result is produced, this new mechanism insures that the system only works on data it determines to be independent, competing, or cooperating in relation to the high-level result.

An inhibiting goal and its associated inhibiting subgoals are generated when the system determines that sufficient work has been done on refining a high-level result. All KSs are then inhibited from producing results that are subsumed by the inhibiting goal or inhibiting subgoals. The specification of the inhibiting goal is taken from the characteristics of the high-level result. By specifying a tolerance around the inhibiting goal, its characteristics can be generalized to extend the range of inhibition. This can eliminate solutions that are similar, though not identical. This is appropriate in environments where answers that have characteristics close to the correct answer are acceptable. The algorithm is given in Figure 4.

In general, inhibiting goals do not preclude the formation of competing solutions that overlap inhibited regions. These solutions will still be formed by extending data outside the inhibited area. Figure 5 illustrates such

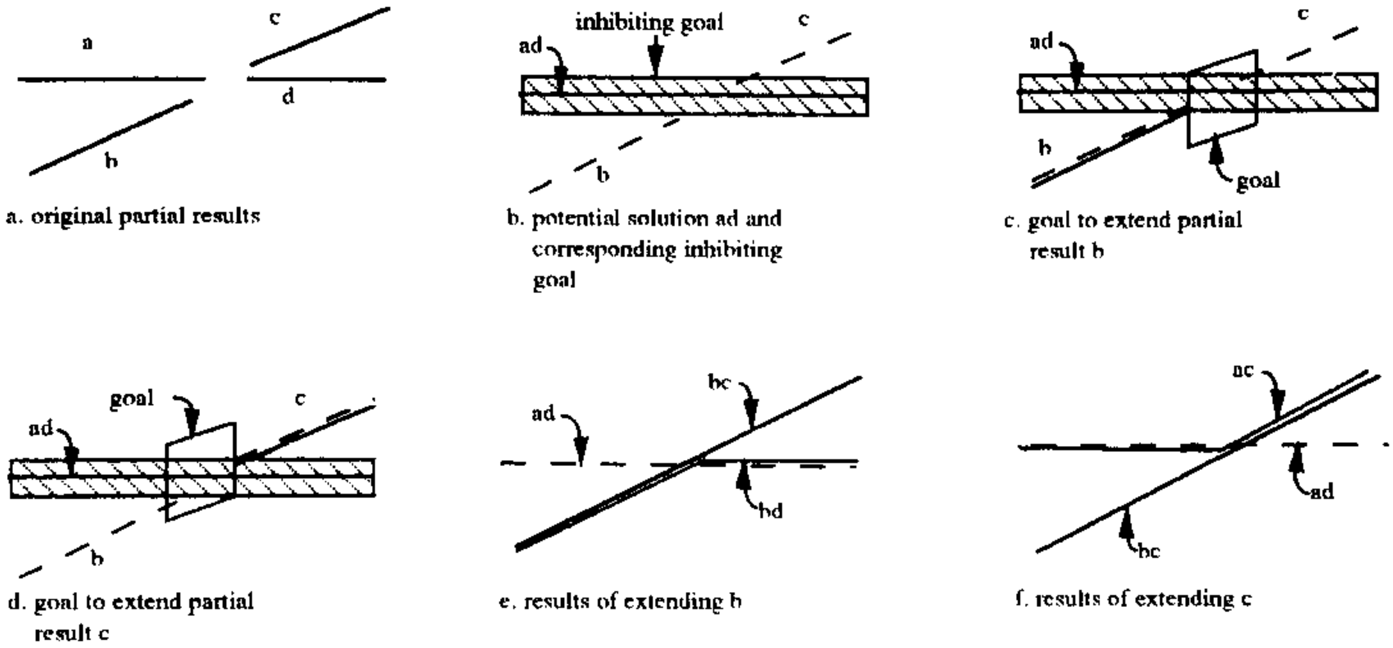


Figure 5: Inhibiting Goal Example

a situation in the previously described distributed vehicle monitoring system. In this scenario, four potential solutions can be formed by merging the segments labeled a, 6, *c* and *d* into the potential solutions *ad, ac, be* and *bd.* If the system were to form *ad* first and create an inhibiting goal to restrict further processing on *ad, ac* could still be formed by extending *c* to *ac* since the goal to extend *c* is neither subsumed nor assisted by the inhibiting goal corresponding to *ad.* It might still be possible for some competing solutions to be precluded in anomalous situations where multiple inhibited regions interact. However, this depends almost entirely on how the intermediate potential solutions are formed.

4.2 Local Context

Along with inhibiting activity based on high-level results, there is also a need to inhibit activity based on a more local context. For example, if any goal that triggered a KS is satisfied before the KS runs, and if the KS can not improve on the results that satisfied the goal, the KS should have its rating decreased. Consequently, which KS satisfies a goal is an important issue. If the scheduler gave priority to the KS1 with the most comprehensive triggering goal, its results might satisfy other goals and eliminate the need to execute their triggered KSIs. The following situation demonstrates this point.

Consider the pending activities KSI_1 and KSI_2, generated from work on two different derivation paths. If executed, KSI_1 would produce result R_l that would subsume R_2, the result of executing $KS!_2$. Executing KSI_1 first would make KSI_2's results redundant, so the scheduler should give $KSI_:$ priority. However, from a local, data-directed perspective, KSI_2 might be given higher priority, even though KSI_1 is the more promising of the two. This can occur if the scheduler incorporates an average of input data credibility in its KSI rating function and if KSI_1 uses lower rated data in addition to highly rated data used by KSI_2. For example, KSI_1 may generate R_1 by extending highly credible data into areas of weak data, and KSI_2 may use only the highly credible data to produce the highest rated component of R_1.

Our ear "er approach to rating KSIs tried to balance the quality of the predicted result with its scope. However, we found that the appropriate balance was situation dependent. Assigning too much priority to scope had the undesirable consequence of making the search too depth-first, while assigning too much priority to quality led to redundant activity as illustrated in the above example. Instead, to choose among the pending KSIs, we need to explicitly take into account the relationships among their predicted outputs.

Using the goal relationships specified in the previous section, the system can form a local understanding of why a KSI is scheduled to be invoked and may instead invoke a different KSI that produces the same results more efficiently. This procedure, called the Local Context Mechanism, can be added to the system as an addition to the scheduler. In general, before executing a KSI, the Local Context Mechanism examines the KSI's triggering goals and searches for a more comprehensive KSI that would also satisfy these goals. If this more comprehensive KSI produces an actual result that is as good in

```
FOR every goal, g, in a KSI's triggering-goal list
   For each hypothesis, h, satisfying g
      IF h subsumes any element of the KSI's predicted
         outputs AND the predicted output cannot
         improve h in any way
      THEN
         Remove the subsumed item from the KSI's set of
            predicted outputs
         Recalculate the KSI's rating

'* Prior to invoking the highest rated KSI *)
:F the KSI's assisting goal list is non-nil THEN
   Invoke the KSI triggered by the most comprehensive
      assisting-goal
ELSE
   Invoke the original KSI
```

Figure 6: Local Context Mechanism

the subsuming area as that expected from the less comprehensive KSI, the subsumed results are removed from the output set of the less comprehensive KSI. The Local Context Mechanism is implemented as a combination of the two algorithms shown in Figure 6.

Table 1: Experiment Summary.

Exp	Env	Algs?	KS ex	Hyps	Goals
E1	1	no	184	246	488
E2	1	yes	64	157	443
E3	2	no	207	335	661
E4	2	yes	130	297	712
E5	3	no	806	938	1894
E6	3	yes	437	728	1714

Abbreviations

Exp: Experiment
Env: The problem solving environment;
1) single vehicle track, no noise
2) single vehicle track, random noise
3) crossing vehicle tracks, random noise
Algs?: Whether the Inhibiting Goal and Local Context algorithms are used.
KS ex: The number of knowledge source executions required to find the solution(s)
Hyps: Number of hyps generated during problem solving.
Goals: Number of goals generated during problem solving.

5 Experimental Results

The goal relationship algorithms were implemented in the Distributed Vehicle Monitoring Testbed (DVMT). The DVMT simulates a network of vehicle monitoring nodes, where each node applies simplified signal processing knowledge to acoustically sensed data in an attempt to identify, locate and track patterns of vehicles moving through a two-dimensional space. A node is responsible for a specific area and attempts to recognize and eliminate errorful sensor data as it integrates the correct data into an answer map. Each node has a blackboard architecture with knowledge sources and blackboard levels of abstraction appropriate for vehicle monitoring. Knowledge sources perform the basic problem solving tasks of extending and refining hypotheses (partial solutions). As described earlier, data-directed and goal-directed goals are used to control problem solving activities.

Experimental results are summarized in Table 1. Three environments were used for testing; a simple environment with a single vehicle track and no sensor noise, an environment with the same vehicle track but with random noise added, and a complex environment with two crossing vehicle tracks and a significant amount of noise. Features used for comparison were the number of knowledge source executions required to produce the solution(s), the number of hypotheses created, and the number of goals created. The system performed more efficiently with the new goal relationship algorithms. For the environment with a single track, 3G% fewer hypotheses and 9% fewer goals were produced and the system required 65% fewer KS executions to compute the answer. In the second environment, 11% fewer hypotheses and 8% more goals were produced and the solution was found with 37% fewer KS executions. Finally, in the complex environment, 22% fewer hypotheses and 10% fewer goals were produced and the solutions were found with 46% fewer KS executions.

In each of the environments, the new algorithms were effective in preventing redundant processing in areas where strongly believed, high-level results were found. This enabled the system to allocate resources for work in noisy areas and areas where the sensed signels were weak. Although the new algorithms caused the system to generate additional goals, most noticeably in the second environment, the resulting improvement in focusing capabilities resulted in a considerable reduction in the number of knowledge sources executed. Finally, the cost of processing goal relationships was significantly less than the cost of processing KSs. The savings were sufficient to result in dramatic decreases in execution time required when the system used goal relationship mechanisms.

6 Conclusion

We have presented a taxonomy of goal relationships that include inhibiting goals. In addition, we have shown that mechanisms for accurately controlling the flexibility provided by the multi-level, cooperative knowledge source model of problem solving can be built as natural extensions to the integrated data-directed and goal-directed architecture.

Goal relationships provide important information for making intelligent control decisions, and they are a useful tool for representing the current state of problem solving in a complex search space. Their use is applicable to tasks in which combined data-directed and goal-directed control is appropriate. In order to exploit goal relationship mechanisms, it is necessary that the quality and characteristics of a KSPs output be roughly predictable and that data-directed goals be constructed so as to contain all the possible outputs that a KS can produce based on the triggering data. Additionally, the computation of goal relationships, data-directed goals, and KS output set approximations must be inexpensive compared to the cost of executing a KS. Finally, it is also important that a data-directed goal's solution set not include a large number of potential solutions that can not be generated by a KS working on the triggering data. Otherwise, many goal relationships will be overlooked.

We are incorporating these concepts into mechanisms for real time control and investigating the use of cooperating, independent, and competing goal relationships for use in complex focusing heuristics and in problem solving termination. We are also examining the potential benefits of adding additional goal attributes indicating the expected amount of resources needed to satisfy a goal, the amount of work already invested in satisfying a goal, the expected number of solutions to a goal, and the likelihood of satisfying a goal. Finally, we intend to expand the notion of cooperation and attempt to reproduce the high-level planning results of Durfee and Lesser [Durfee and Lesser, 1986] through the use of goal relationships.

References

[Corkill *et al,* 1982] Daniel D. Corkill, Victor R. Lesser, and Eva Hudlicka. Unifying data-directed and goal-directed control: An example and experiments. In *Proceedings of the National Conference on Artificial Intelligence,* pages 143-147, Pittsburgh, Pennsylvania, August 1982.

[Durfee and Lesser, 1986] Edmund H. Durfee and Victor R. Lesser. Incremental planning to control a blackboard-based problem solver. In *Proceedings of the Fifth National Conference on Artificial Intelligence,* pages 58-64, August 1986.

[Erman *et al,* 1980] Lee D. Erman, Frederick Hayes-Roth, Victor R. Lesser, and D. Raj Reddy. The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys,* 12(2):213-253, June 1980.

[Hayes-Roth and Lesser, 1977] Frederick Hayes-Roth and Victor R. Lesser. Focus of attention in the Hearsay-II system. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* pages 27-35, Tiblisi, Georgia, USSR, August 1977.

[Hayes-Roth, 1985] B arbara Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence,* 26(3):251 321, July 1985.

[Johnson and Hayes-Roth, 1987] M. Vaughn Johnson and Barbara Hayes-Roth. Integrating diverse reasoning methods in the BB1 blackboard control architecture. In *Proceedings of the National Conference, on Artificial Intelligence,* pages 30-35, Seattle, Washington, July 1987.

[Lesser and Corkill, 1983] Victor Lesser and Daniel Corkill. The Distributed Vehicle Monitoring Tcstbed: A tool for investigating distributed problem solving networks. *AI Magazine,* 4(3):15-33, Fall 1983. (Also to appear in *Blackboard Systems,* Robert S. Engelmore and Anthony Morgan, editors, Addison-Wesley, in press, 1988 and in *Readings from AI Magazine 1980-1985,* in press, 1988).

[Lesser and Erman, 1977] Victor R. Lesser and Lee D. Erman. A retrospective view of the Hearsay-II architecture. In *Proceedings of the Fifth InternationalJoint Conference on Artificial Intelligence,* pages 790-800, Tiblisi, Georgia, USSR, August 1977.

[Lesser *et al,* 1987] Victor R. Lesser, Daniel D. Corkill, and Edmund H. Durfee. An update on the Distributed Vehicle Monitoring Testbed. Technical Report 87-111, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts 01003, December 1987.

[Nii, 1986a] H. Penny Nii. Blackboard application systems: Blackboard systems from a knowledge engineering perspective. *AI Magazine,* 7(3):82-106, Conference 1986.

[Nii, 1986b] H. Penny Nii. Blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures. *AI Magazine,* 7(2):38-53, Summer 1986.