Needles in a Haystack : Plan Recognition in Large Spatial Domains Involving Multiple Agents

Mark Devaney and **Ashwin Ram**

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
markd@cc.gatech.edu
ashwin@cc.gatech.edu

Abstract

While plan recognition research has been applied to a wide variety of problems, it has largely made identical assumptions about the number of agents participating in the plan, the observability of the plan execution process, and the scale of the domain. We describe a method for plan recognition in a real-world domain involving large numbers of agents performing spatial maneuvers in concert under conditions of limited observability. These assumptions differ radically from those traditionally made in plan recognition and produce a problem which combines aspects of the fields of plan recognition, pattern recognition, and object tracking. We describe our initial solution which borrows and builds upon research from each of these areas, employing a pattern-directed approach to recognize individual movements and generalizing these to produce inferences of large-scale behavior.

Introduction

Plan recognition, the problem of inferring goals, intentions, or future actions given observations of ongoing behavior, has been widely studied in Artificial Intelligence in the context of a number of different applications. Some of these applications include the inference of a user's beliefs in natural language or story understanding (e.g., Di Eugenio 1995), inference of a user's intents in intelligent user interfaces (e.g., Wærn 1996), identification of an autonomous agent's goal in order to coordinate activity (e.g., Huber and Durfee 1995), and for program understanding in the field of software engineering (e.g., Woods and Yang 1998).

However, even in these diverse application contexts, three major assumptions are typically made:

Single agent The plan is being carried out by a single agent acting alone. Work in "multi-agent" plan recognition shares this assumption as well, the distinction being that an agent is considering the possible *individual* plans of several agents (e.g., Huber and Durfee 1995; Tambe 1995) or that there are other agents active in the world which may be causing changes (e.g., Traum and Allen 1991).

Complete and correct information The observations available are correct, i.e., they correspond to the actual behavior of the observed agent, and they are complete, i.e., no actions were performed which were not noticed. This assumption generally follows from the assumption that the watched agent is either actively facilitating observation ("intended plan recognition") or is at least not actively obstructing it ("keyhole plan recognition").

Small-scale data The input to the recognition process comes from a fairly limited set of potential actions and occurs over relatively short time periods. This assumption, particularly when combined with the single agent assumption, allows use of computationally intensive methods such as exhaustive search, probabilistic methods, and reasoning from first principles.

In contrast, we are researching the problem of plan recognition in a very complex real-world domain in which these assumptions do not hold. Specifically, we are concerned with methods which may be used to identify and predict behaviors in a domain involving hundreds of agents moving together over large distances and long time scales and in which the observer receives noisy and incomplete observational data.

The problem presents an interesting combination of issues from plan and pattern recognition, agent and object tracking, qualitative representation, and geographical information system (GIS) theory. We are interested in investigating how the task of plan recognition can be accomplished when traditional assumptions do not hold, a situation we expect to become increasingly prevalent as domains of research are expanded to more real-world scenarios.

Problem description

The domain in which we are conducting our research consists of data obtained from training battles conducted by actual troops at the US Army's National Training Center (NTC). These battles involve hundreds of participants, last several days, and take place over a very large geographical area. The problem is constrained such that the only input available to the recognition system is limited to low-level telemetry information – positions of participants (with identifiers) over time, as would be the case with human observation of the same battles. In particular, we make no assumptions about the availability of information such as terrain, weather conditions, etc., which could be useful but cannot be assumed to exist in many real world situations.

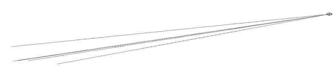

Figure 1: The needle : a gathering maneuver

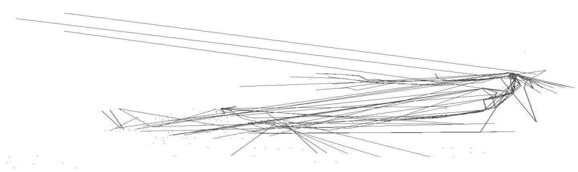

Figure 2: The haystack : all movements

The task of our research is to identify from this data repeated patterns of movement which correspond to planned "maneuvers" being executed by the participants. This is possible because military doctrine, much like plays in football, largely consists of prescribed sets of tactical maneuvers which are executed in fairly stereotypical ways. The observed battles are not random melees but are the (largely) purposeful actions of intelligent agents acting in pursuit of high level goals or plans. The execution of these plans requires coordinated activities which generate identifiable patterns of movement that can be employed by a recognition process to both identify these plans and predict future movements.

To illustrate the nature of the problem, figure 1 is an example of a short-term maneuver corresponding to a meeting in which a significant number of agents collect at a point, most likely in preparation for some future coordinated activity. The lines indicate the paths traveled by the participating units and the (small) circle spatially bounds their ending points.

The difficulty of performing this recognition is shown by figure 2 which depicts all of the input data up to the time at which the gathering occurred, the space of information from which the maneuver must be "picked out". This corresponds to the paths of each unit from the beginning of the battle to the time the meeting occurs.

Working with domain experts, we have identified several such strategic patterns which our system needs to isolate and recognize during the course of actual battles. Both the inherent qualities of the domain and the realities of collecting data from the real world require assumptions fundamentally different from those usually made in plan recognition research:

Multiple agents The step-by-step actions of individual agents are significantly less important than the coordinated activities of *groups* of agents. In complex multi-agent domains, "interesting" maneuvers require the simultaneous participation of multiple agents, and furthermore, these maneuvers are very often only representable as relationships between agents and not as agglomerations of identical actions performed by single agents. For example, "meeting at a location" is an inherently multi-agent maneuver and one which is independent of the specific details of the movements of any of the individual agents involved. Moreover, the agents participating in the meeting maneuver will have a wide variety of patterns of movement and it is only when considered in relation to the other participants do they have "meaning". The degree of noise in the input can also be a factor in the choice to disregard the actions of individual agents as much more confidence can be ascribed to hypotheses which are supported by information acquired from multiple agents.

Incomplete and incorrect information Data collection from the real world is inherently noisy, and this fact is compounded in large-scale domains. For example, in the domain of training battles described here, telemetry information is obtained from GPS sensors physically attached to participants and it is still highly incomplete and incorrect. Ideally, for each agent, the sensors would supply a position report for each time step of the battle. In reality, agents are "lost" during the course of a battle and no positions are reported for those time intervals. Additionally, there is a great deal of noise in the form of incorrectly-reported positions, making it appear that an agent is traveling in a highly erratic path. In an actual battle or surveillance task, telemetry information would have to be obtained from satellite and other intelligence gathering sources and would exhibit an even greater degree of noise and incompleteness.

Large-scale data Domains involving multiple agents acting over long time intervals or requiring high rates of sampling can generate enormous quantities of information. The number of agents can range from the order of tens in sporting events, for example, to hundreds, as is the case in the domain described here. Depending on the length of activity in the domain and the rate at which information is sampled, there can be anywhere from tens to thousands of state descriptions generated *per agent*. A further distinguishing characteristic of a real-world domain is that the data not only arrives in real time, but more importantly, must be processed in real time (Barès, et. al. 1994). A final assumption resulting from the scale of the data is that much of the input does not correspond to any meaningful pattern of activity. It is not noise *per se* in that it is correct information, it represents random or unplanned movements rather than purposeful action.

While the combination of these assumptions presents a relatively novel problem, previous research in a number of fields has addressed some of the issues raised here. Work by Mohnhaupt and Neumann (1991) examines the task of generating movement events such as "park" and "turn-off" from telemetry information in the domain of moving vehicles. However, they employ rather complex data structures and algorithms which do not scale efficiently to the type of domain in which we are working. A similar problem, referred to as spatio-temporal surveillance, is addressed by Howarth and Buxton (1992), also in the domain of vehicle movements, but which relies on the ability to identify "regions" in the geographic space, which we are unable to do. The goals of their research, though, are very similar to ours, with the exception that we are primarily interested in longer-term events which are composed from more primitive actions, rather than the primitive actions themselves.

Huber and Durfee (1992) address the problem of predicting movements in a real-world task involving autonomous robots. However, this approach assumes that an agent will move directly to its destination without avoiding obstacles or "feinting", and is further limited to the actions of a single agent, two assumptions which do not hold in our domain. However, their use of probabilistic methods has proven useful in overcoming noise and is something which we intend to investigate further. Tambe (1995) also describes research in tracking the behavior of agents in a real-world task, but like Huber and Durfee, focuses on the behavior of individual agents or groups of agents acting identically.

Another very closely related problem is the identification of stereotypical groupings of military forces or "templates". Woods (1993) addresses this problem for static positions and employs a constraint satisfaction framework along with a technique for abstracting the representation to overcome noise and complexity in the domain. While our research is focused on dynamic patterns of movement rather than position, we plan to investigate the potential usefulness of a similar approach, particularly for representation and recognition of more complex patterns of activity.

Solution

A solution to the problem of plan recognition in a domain such as the one described here must be capable of identifying changing relationships between agents but not have high computational or storage requirements nor be sensitive to noise in the data. Our solution has required both computations of parsimonious yet robust representations as well as an efficient pattern matching algorithm.

Constructing representations

Given that the only input directly available to the plan recognition process is the position of agents in a global coordinate system over large time periods, there is a need to both augment the amount of information that can be provided to the recognition process and a competing need to condense the amount of information to prevent computational intractability. Our approach has been a layered one in which successively higher levels of representation are computed from the raw data, generating decreasing amounts of data which convey more focused information. The lowest level of representation is information about individual agents such as position and velocity, followed by representations of relationships between pairs of agents, and finally, representations of patterns of agent-pair relationships.

Individual agents The first level of representation identifies important characteristics of individual agents. A subset of the features identified as useful by Mohnhaupt and Neumann (1991) is used at this level to describe agents at each time step : velocity, heading (orientation), and position (absolute). Velocity is computed as distance traveled during the prior temporal interval. Heading, or direction of travel, is computed from the path of travel which is first smoothed using a non-lookahead smoothing function and then converted to a relative compass orientation, an approach often used in the field of qualitative reasoning (e.g., Hernández 1994). These are the only features used at this time, although there are additional features which can be constructed such as relative positions (e.g., behind) or qualitative descriptions of movement such as "starting" or "stopping". Whether additional features will be required to represent more complex plans remains to be seen, however our general approach does not preclude the use of finer or more complex representations as long as they are computable within the constraints of the problem. It is also possible to construct higher-level attributes from raw data using domain-specific knowledge (e.g., Devaney and Ram, 1997), but this, of course, incurs additional computational overhead. Currently we are most interested in what type of success can be achieved using the minimum amount of knowledge, but the processes described here are independent of which attributes are employed. Ultimately, the decision of what properties to compute is a tradeoff between the cost of computing those properties and the degree to which they influence the quality of solutions obtainable.

Agent-pair relationships One of our primary assumptions of plan recognition involving multiple agents is the necessity of representing interactions among groups of units. To initially identify potential groups, the next stage of representation describes the relationships between all *pairs* of agents. These representations are comparisons of the primary attributes of individual units, and again, because of the potential for computational complexity in domains of this scale, we seek to represent the least amount of information possible which can still be of use. For this reason, the differences are represented qualitatively, with only important distinctions being preserved. Relative distances are computed as a percentage of the maximum possible distance given the bounding rectangle of all units, a representation that has proven useful for spatial reasoning (Gahegan 1995). Heading is stored simply as a binary comparison (same/not-same), although it may prove necessary to employ a slightly finer-grained range representation in the future (such as that described by Clementini, Di Felice, and Hernanández 1997). Finally, relative velocity is stored as a three-attribute feature (both moving, neither moving, different).

These properties of agent-pairs are referred to as the "primitive" relationships because they are obtained directly from the low-level data. Again, these relationships may need to be augmented or refined in order to capture more complex patterns in the data, but these have proven useful in our initial research.

There are eighteen possible combinations of the three attributes, described in table 1.

The number of primitive relationships depends, of course, on the number of primary attributes and their possible values and can get fairly large if more detailed representations are employed. In our domain we have

Table 1: Taxonomy of primitive relationships

Heading	Velocity	Distance	name
not same	not same	same	[uninteresting]
not same	not same	decreasing	s-approach
not same	not same	increasing	s-move away
not same	neither moving	same	not moving
not same	neither moving	decreasing	(impossible)
not same	neither moving	increasing	(impossible)
not same	both moving	same	[uninteresting]
not same	both moving	decreasing	approach
not same	both moving	increasing	move away
same	not same	same	[uninteresting]
same	not same	decreasing	s-approach
same	not same	increasing	s-move away
same	neither moving	same	not moving
same	neither moving	decreasing	(impossible)
same	neither moving	increasing	(impossible)
same	both moving	same	move together
same	both moving	decreasing	approach
same	both moving	increasing	move away

found these attributes sufficient for capturing interesting patterns of movements, and the number of potential primitives in a more robust representation is actually not as large as may first seem. This is due to the fact that many combinations are physically impossible (e.g., it is not possible for two units to have no velocity yet the distance between them change), and many are "uninteresting" in the sense that they do not capture a significant change in the relationship between units. Finally, many of the relationships are symmetrical and can thus be converted into a single equivalence class. For example, out of the eighteen possible combinations of the primitive attributes used in the training battle domain, four are impossible, three are inherently uninteresting, and six of the relationships form three equivalence classes, leaving only six types of primitive relationship which need to be represented and maintained.

Even this relatively simple representation of agent pair relationships is $O(n^2)$, which precludes storing these values over time intervals of any significant length given that n can be on the order of hundreds. Therefore, it is essential to retain these relationships only as long as necessary. This is performed by the plan recognition process, which identifies and retains only those relationships which combine to form interesting patterns in the data.

Plan recognition

The agent-pair relationships described above are best viewed as *events*, actions which can be described by a verb (Neumann 1991), essentially changes in the relationships between primitive attributes of agents (Howarth 1994). Research in object tracking has been primarily concerned with movements at this level, however we are interested in more complex patterns of action as produced by the execution of plans as well as patterns involving groups rather than single agents.

In our approach, the primitive relationships are the building blocks of more complex patterns which are formed from temporal combinations of events, such as the gather maneuver described in figure 1 in which an event of units moving together is followed by an event of them waiting in position. This representation is recursive in nature, in that higher-level patterns are formed from combinations of other patterns. For example, a "gather-disperse" maneuver is a combination of the gather and the dispersal patterns which are themselves combinations of lower-level patterns, etc.

The other unique aspect of our research is that we are ultimately concerned with identification of *groups* of agents engaged in common patterns of behavior and that these group behaviors are formed from the different behaviors of individual agents occurring together. The fact that diverse individual behaviors combine to form group behaviors together with the fact that there is no one distance metric which could capture groupings for all possible patterns, precludes an approach which relies on clustering of the data prior to the recognition of patterns and plans. Rather, the key to our solution has been to reverse the usual approach and detect patterns first among agent-pairs and then leverage this knowledge to form groups based on co-occurring actions.

Since we are operating under constraints of real time and large-scale data, it is crucial that any algorithms be sensitive to potential computational complexity and the amount of space needed to hold computed information. Since there are $(n^2)/2$ groups of unit pair relationships, a large number given that n is on the order of hundreds in this domain, it is computationally infeasible to attempt to compute all possible groupings or even to maintain all unit pair relationships for more than a limited number of time steps. Instead, we employ a pattern-directed approach (Hayes-Roth and Waterman 1978) whereby the retention of information is governed by its potential "interestingness". In other words, the only information retained about agent-pair relationships is that which has the potential to instantiate a known pattern.

Briefly, the algorithm consists of computing all possible agent-pair relationships at each time step, then comparing these with a library of pattern templates. Relationships which match any of these templates are retained in a list of potentially satisfiable patterns indexed by the agents participating in the relationship. Once an agent-pair has been identified as potentially instantiating a pattern or patterns, its events are retained for succeeding time intervals by linking them into the pattern(s) which it partially instantiates. These partial instantiations are examined at each time step as well, and one of three conditions occurs:

complete instantiation The new event completely instantiates a pattern template and the event sequence is replaced by that higher-level representation.

decay The new event does not match the template and the pattern has remained uninstantiated for too long, at which point it is discarded.

chaining The new event neither fully instantiates the pattern nor causes it to decay completely, so it is just added to the event chain.

Table 2: Plan recognition algorithm

At each time step t
1) For each agent i, compute state of $agent_i$ at $time_t$
2)For each agent i and j (i != j)
  A) Compute agent-pair relationships for $agent_i$, $agent_j$ at $time_t$
  B) If $relationship_{ij}$ matches beginning of new pattern template, instantiate a new potential $pattern_{ij}$
  C) For each existing $pattern_{ij}$
    i) add $relationship_{ij}$ to event chain of $pattern_{ij}$
    ii) mark $pattern_{ij}$ as fully instantiated if $relationship_{ij}$ matches next event in $pattern_{ij}$ and that event is the last in the chain.
    iii) increment decay value of $pattern_{ij}$ if $relationship_{ij}$ does not match next event in $pattern_{ij}$
    iv) discard potential $pattern_{ij}$ if decay threshold exceeded
3) For each fully instantiated $pattern_{ij}$, combine all patterns involving i or j which occur co-temporally.

Event sequences correspond to a temporal ordering, and we employ a temporal representation based on Allen's (1991) temporal calculus which has proven sufficient to capture the maneuvers identified by our domain experts as interesting. Essentially, a pattern consists of a series of events, some of which are co-occurring and others which occur in succession. In our implementation, an event is considered co-occurring with another if there is any amount of temporal overlap between the two states. We employ a similarly generous definition of succession – two events are considered to occur in succession if one begins after the other one has ended. The co-occurrence relation is relatively unproblematic, however, our notion of succession raises the issue of how much time can elapse between state changes for a pattern to still be counted as such. There is a conflicting need to allow intermediate states which may be caused by noise or deliberate deception but also to require that patterns have some temporal "cohesiveness". The solution to this problem involves associating a decay value with partially instantiated patterns which is increased as unmatched states accrue. The partial patterns are discarded when this decay exceeds a threshold.

The identification of group behaviors occurs when an agent-pair fully instantiates a known pattern. At the end of processing all agent-pair relationships, the algorithm combines all newly instantiated full patterns based on spatial and temporal co-occurrence by examining patterns which have common participants. This results in a set of high-level behaviors involving multiple units and these behaviors can be ranked based on the number of participants, indicating a relative "interestingness".

The algorithm is described in more detail in table 2.

Evaluation

In contexts where the researcher knows what the right answer should be, evaluation is more or less a matter of running a system and measuring its success at finding that right answer under a number of conditions. This, however, requires that sufficient domain knowledge exists for a solution to be calculated and that someone with that domain knowledge exists and is available for evaluation of the system. In other contexts, evaluation is more subjective because there is no one "right" answer and the output must be judged by some criteria determined by the researcher as to its interestingness.

Our evaluation thus far has employed the use of several domain experts (US Army officers) to identify patterns of movement in actual battle data which they deem relevant or interesting. These patterns occur with moderate frequency within battles and among different battles, further indicating their usefulness as sources of prediction. The patterns currently are at the level of the "gather" maneuver depicted in figure 1, involving event chains of several time steps in length. Our system is able to isolate and identify these patterns in real battle data which exhibits high degrees of complexity and noise.

A formal and comprehensive evaluation will be required to more fully examine the claims made, but should follow the general pattern of the informal evaluations already made. We have made plans for further consultation with domain experts who should be able to augment the library with interesting and useful patterns. Once the library has been become sufficiently complex a systematic evaluation can be made, running the pattern recognition system on the entire corpus and annotating the battles with the patterns identified. This output can then be judged by both experts and non-experts as "sensible", i.e., were all patterns identified as specific maneuvers reasonably indicative of those maneuvers.

Finally, while our current evaluation has focused on the ability of the system to represent and identify patterns having significance in the military domain in which we are working, our research claims are made at a higher level, that of spatio-temporal patterns as produced by the movements of multiple agents. Therefore, future evaluation will also be concerned with the ability of the system to represent and recognize patterns which occur with high frequency without consideration of domain-specific importance.

Conclusions and future work

Research in plan recognition has traditionally focused on domains in which the task is to identify the goals and future behavior of a single agent acting from a relatively small space of potential activities in a context of perfect or near-perfect observability. In contrast, we describe the problem of plan recognition in complex multi-agent domains in which these assumptions do not hold. These domains require an approach which focuses on the interactions *among* agents, is sensitive to the potential for computational complexity, and does not require complete observability.

Our solution has been to employ fairly simple representations which capture successively higher levels of description of input data and to retain only that data which has the potential to be part of a known plan. The motivation for a parsimonious representation has been due to computational considerations and also due to the desire for the development of a system which is able to autonomously *learn* or *discover* patterns directly from the raw data. While the choice of primitives used in the representation will ultimately depend on the particular application domain, the general approach has proven both computationally tractable and effective.

Future work will focus on exploring the scalability of the approach in order to identify limitations on the size of the plan library and the complexity of its plans while remaining within the real-time constraints of the problem. The issue of real-time response is critical in many applications, and ideally a recognition system in a real-world domain such as the one described here should be able to rapidly generate hypotheses while input data streams in. This requires the ability to generate multiple candidate matches using partial information and to rank them by some confidence metric, and is particularly important when the plan library becomes larger and more complex. Currently we have not had the need to address these issues because of the fairly small size of our plan library, but as the number and complexity of the pattern templates increases these issues will become increasingly important.

While we suspect that relatively minor modifications and enhancements will need to be made in our basic representations, we are confident that the general approach will prove successful as the size and complexity of the patterns is increased. The ability to reason about interactions among agents in competitive or cooperative situations is becoming increasingly useful as more and more attention is devoted to multi-agent research. The trend toward applying AI techniques to real-world problems is also placing demands on these techniques that they be scalable as well as effective.

Acknowledgements

This research was funded by the US Army Research Laboratory under grant DAKF11-97-D-0001-0007. The authors wish to thank Sven Koenig and the anonymous AAAI reviewers for their helpful comments and suggestions.

References

Allen, J.F. 1991. Temporal reasoning and planning. In J.F. Allen, et. al. *Reasoning About Plans*, 1-67. San Mateo, CA : Morgan Kaufmann.

Barès, M., et. al. 1994. XPLANS: Case-based reasoning for plan recognition. *Applied Artificial Intelligence* 8:617-643.

Clementini, E., Di Felice, P., and Hernández, D. 1997. Qualitative representation of positional information. *Artificial Intelligence* 95:317-356.

Devaney, M. and Ram, A. 1997. Situation development in a complex real-world domain. In *Proceedings ICML '97 Workshop on Applications of ML to the Real World.*

Di Eugenio, B. 1995. Plan Recognition and Natural Language Understanding. In *Proceedings IJCAI '95 Workshop : The Next Generation of Plan Recognition Systems : Challenges for and Insight from Related Areas of AI.*

Gahegan, M. 1995. Proximity operators for qualitative spatial reasoning. In A. U. Frank & W. Kuhn (Eds.), *Spatial Information Theory. A Theoretical Basis for GIS, International Conference (COSIT '95)*, Lecture Notes in Computer Science 988, 31 - 43, Springer-Verlag.

Hernández, D. 1994. *Qualitative Representation of Spatial Knowledge*, Lecture Notes in Computer Science 804, Springer Verlag.

Howarth, R.J. 1994. Spatial Representation, Reasoning and Control for a Surveillance System. Ph.D. diss., Queen Mary and Westfield College.

Howarth, R.J. and Buxton, H. 1992. Analogical representation of spatial events for understanding traffic behaviour. In *Proceedings European Conference on Artificial Intelligence (ECAI)*, 785 - 789.

Huber, M. J. and Durfee, E. H. 1992. Plan recognition for real-world autonomous robots: Work in progress. In *Working Notes AAAI Symposium : Applications of AI to Real-World Autonomous Robots.*

Huber, M. J. and Durfee, E. H. 1995. Deciding when to commit to action during observation-based coordination. *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS)*, 163 - 170.

Mohnhaupt, M. and Neumann, B. 1991. Understanding object motion: Recognition, learning, and spatiotemporal reasoning. *Robotics and Autonomous Systems* 8:65 - 91.

Neumann, B. 1989. Natural language description of time-varying scenes. In D. Waltz (Ed.), *Semantic Structures : Advances in Natural Language Processing*, 167 - 204. Lawrence Erlbaum Associates.

Tambe, M. 1995. Recursive agent and agent-group tracking in a real-time, dynamic environment. In *Proceedings of International Conference on Multi-Agent Systems (ICMAS)*, 368-375.

Traum, D. R. and Allen, J.F. 1991. Causative forces in multi-agent planning. In Y. Demazeau & J-P. Müller (Eds.), *Decentralized AI*, 89 - 105. Elsevier.

Wærn, A. 1996. Recognising human plans: Plan recognition in human-computer interaction. Ph.D. diss., Swedish Institute of Computer Science.

Waterman, D.A. and Hayes-Roth, F. eds. 1978. *Pattern-directed Inference Systems.* Academic Press.

Woods, S. 1993. A method of interactive recognition of spatially defined model deployment templates using abstraction. In H. Merklinger et al. (Eds.), *Proceedings of the Knowledge-based Systems and Robotics Workshop*, 665-675. Department of National Defence, Government of Canada.

Woods, S. and Yang, Q. 1998. Program understanding as constraint satisfaction. *Journal of Automated Software Engineering* 5(2). Kluwer Academic Publishers.