A Bayesian Blackboard for Information Fusion

Charles Sutton
University of Massachusetts
140 Governors Drive
Amherst, 01002, MA
USA
casutton@cs.umass.edu

Clayton Morrison, Paul R. Cohen, Joshua Moody, Jafar Adibi
USC Information Sciences Institute
Center for Research on Unexpected Events
4676 Admiralty Way, CA 90292
USA
clayton,cohen,moody,adibi@isi.edu

Abstract – *A Bayesian blackboard is just a conventional, knowledge-based blackboard system in which knowledge sources modify Bayesian networks on the blackboard. As an architecture for intelligence analysis and data fusion this has many advantages: The blackboard is a shared workspace or "corporate memory" for collaborating analysts; analyses can be developed over long periods of time with information that arrives in dribs and drabs; the computers contribution to analysis can range from data-driven statistical algorithms up to domain-specific, knowledge-based inference; and perhaps most important, the control of intelligence-gathering in the world and inference on the blackboard can be rational, that is, grounded in probability and utility theory. Our Bayesian blackboard architecture, called* AIID, *serves both as a prototype system for intelligence analysis and as a laboratory for testing mathematical models of the economics of intelligence analysis.*

1 Introduction

Intelligence analysts deal with vast amounts of information that arrives asynchronously, from a variety of heterogeneous sources, with varying accuracy and credibility. Analysts must construct interpretations of what is happening, inferring participants' intentions and which actions should be taken in response. Interpretation goes far beyond simply finding patterns in raw data: Patterns are not interpretations, syntax is not semantics, and data-mining is of limited utility. Lets go further: Pure data-driven algorithms will never produce interpretations — hypotheses about the meaning of data — because the hypothesis space is unmanageably large. The business of interpreting evidence is both data- and model-driven, knowledge about the world is indispensable, as are reasoning strategies sufficient to maintain numerous, simultaneous, hypothetical interpretations.

We have developed a prototype of a *Bayesian blackboard* called AIID an Architecture for the Interpretation of Intelligence Data. As the name suggests, a Bayesian blackboard combines the technologies of blackboard systems and Bayesian belief networks. It extends traditional blackboard techniques with a principled method for representing uncertainty, and it extends traditional belief network techniques by incrementally building models. One consequence of this marriage is that the control of intelligence gathering in the world and inference on the blackboard can be rational, that is, grounded in probability and utility theory.

2 Blackboards and Bayesian Networks

Blackboard systems are knowledge-based problem solvers that work through the collaboration of independent reasoning modules. They were developed in the 1970s and originally applied to signal-processing tasks. The first, HEARSAY-II [1], was used for speech recognition, employing acoustic, lexical, syntactic, and semantic knowledge. Other systems were applied to problems as diverse as interpretation of sonar data, protein folding, and robot control [2].

Blackboard systems have three main components: the blackboard itself, knowledge sources (KSs), and control. The *blackboard* is a global data structure that contains hypotheses or partial solutions to a problem. The blackboard is typically organized into sections by levels of abstraction. For example, HEARSAY-II had different levels for phrases, words, syllables, and so forth. *Knowledge sources* are small programs which post results of local computations to the blackboard. (Ideally, knowledge sources interact only by posting to the blackboard.) Different KSs use different types of knowledge: for example, one might use a grammar to generate words which are likely to occur next, while another might detect phonemes directly from the acoustic signal. While no single knowledge source can solve the problem, working together they can. Getting knowledge sources to "work together" is the task of blackboard *control*. Generally it works like this: KSs watch for particular kinds of results on the blackboard; for instance, a phrasal KS might look for hypotheses about adjacent words. When a KS is "triggered" it creates a *knowledge source activation record* (KSAR) in which it requests the opportunity to run, make inferences, and modify the blackboard. These KSARs are ranked, and the top-ranked KSAR is invited to do its work.

The operation of a blackboard system can be seen as search for hypotheses that explain the data at each level of abstraction, using the KSs as operators. Rather than search bottom-up (i.e., from the data level to the most abstract level) or top-down, blackboard systems can search opportunistically, dynamically rating KSARs based on the current data and on the partial solutions that exist so far.

Heuristic methods generally have been used [3] to represent uncertainty: for example, HEARSAY-II used a numerical confidence score that ranged from 1 to 100. One of our contributions is to provide hypotheses on the blackboard with a real probabilistic semantics. To understand our approach, one must know a little about belief networks.

Belief networks are graphical structures in which the nodes represent propositions with associated probability distributions. For instance, in Figure 2 one sees the proposition that two units, denoted by variables `?U1` and `?U2` are fixing and flanking a third, denoted by `?R1`. The probability of this proposition is *conditional*, it depends on the probabilities of the other nodes in the graph, in this case the nodes that represent individual fixing and flanking maneuvers. Belief networks, then, are directed graphs in which nodes represent propositions with conditional distributions (if the nodes point to other nodes) or unconditional distributions (if the nodes are "evidence" propositions). The conditional distributions are stored in conditional probability tables, or CPTs. For an introduction to belief networks, see [4].

Belief networks that describe several similar objects often have many copies of common subnetworks. For example, in the military domain, every unit has attributes like UNIT-TYPE (e.g., tanks, infantry, artillery) and DIRECT-FIRE-RADIUS. These attributes have relationships that do not depend on the particular unit: for example, tanks can shoot farther than infantry. If we simply have nodes called UNIT-TYPE-FOR-UNIT1, DIRECT-FIRE-RADIUS-FOR-UNIT1, etc., then the humans constructing the network need to specify separate, identical CPTs for each unit, which is impractical because there could be many units, and we do not know in advance how many.

Several authors [5, 6, 7] have addressed this problem by breaking up large belief networks into smaller subnetworks. Subnetworks have designated input nodes—which have no conditional distribution, requiring that their distribution be specified in a different subnetwork—and resident nodes, which do have CPTs. A standard belief network can be created from subnetworks by unifying the input nodes of one subnetwork with the resident nodes of another. Repetitive structure can be specified just once in a subnetwork and instantiated multiple times to exploit redundancies in the domain.

Object-oriented Bayesian networks (OOBNs) [6, 8] employ strong encapsulation between subnetworks. Each subnetwork defines a set of output variables, and combinations between subnetworks are made only by connecting the output variables of one subnetwork to the input variables of another. Each subnetwork can be seen as a single cluster node in a higher-level belief network, so that an OOBN defines a single probability distribution over its variables. Since the subnetworks are connected by a knowledge engineer, rather than automatically, OOBNs are not a technique for incrementally building models based on incoming evidence.

Network fragments [7] are another approach to constructing modular subnetworks. Unlike OOBNs, nodes can be resident in more than one fragment, so they designate influence combination methods for combining distributions from multiple fragments. So network fragments can combine in more unexpected ways than in OOBNs, which precludes specialized inference algorithms, but can be more flexible for specifying complicated belief networks.

3 Bayesian Blackboard Architecture

The blackboard of AIID represents the system's current beliefs about the domain. The blackboard contains a possibly disconnected belief network that includes previous observations, background knowledge, and hypotheses about the data. In the military domain, the blackboard contains nodes that include sightings and hypothesized locations of enemy units, locations of key terrain, and hypotheses about the enemy's tactics and strategy. A sample blackboard is shown in figure 1.

As in the subnetwork literature, we use a first-order extension to belief networks to represent multiple similar entities more conveniently, analogous to the extension of propositional logic to predicate logic. Instead of naming the random variables by a single atom, e.g. UNIT-MASS, each node has a *node-type*, for example, UNIT-MASS, and a set of *arguments*, for example, "Tank Regiment 1." Logic variables can be used as arguments in KSs to describe a relationship that does not depend on the particular argument values. The combination of a node-type and arguments uniquely specifies a node on the blackboard.

Information on the blackboard can occur on different temporal scales. For example, we can represent a short meeting between two people as a punctual event, while an activity like "Planning-Attack" takes an extended amount of time. We handle these scales using two temporal representations: a tick-based representation, and an interval representation. At lower levels of the blackboard, where we are considering things like meetings and current locations, each network node is indexed by the time it occurs, and the entire network is a Dynamic Bayesian Network (DBN) [9]. At higher levels of the blackboard, which correspond to long-term actions and intentions, we represent events by the interval in which they occur. Each event has a start-time and an end-time that are explicit nodes in the network. These two representations are integrated in AIID.

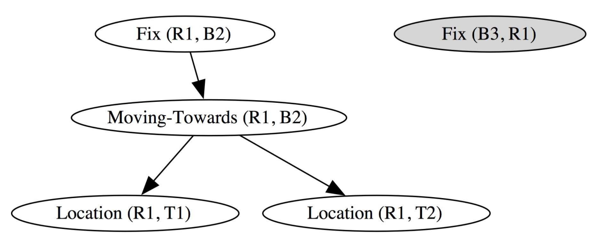

Fig. 1: A sample blackboard in the military analysis domain

3.1 Knowledge Sources

Knowledge sources are procedures that modify the blackboard. Knowledge sources can post new nodes to the blackboard, add edges, alter CPTs, and remove nodes. Every KS

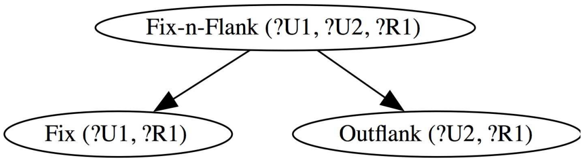

Fig. 2: A sample knowledge fragment for the military analysis domain

has three components, which can be arbitrary procedures: a confidence, a precondition, and an action. The confidence returns a number that indicates how intrinsically useful the KS is. The precondition is run when the blackboard changes and returns true if the KS is applicable. The action is the procedure that actually modifies the blackboard.

As in conventional blackboard systems, KS actions can be full-fledged programs. For example, a KS might use arbitrary heuristics to post simplifying assumptions to make reasoning more tractable. In the military domain, for example, our implementation uses a grouping KS that treats several enemy units as a group if they seem sufficiently close.

Another type of KS cleans up nodes that accumulate from old time steps. Old nodes can slow down inference without greatly affecting current beliefs. Cleanup KSs can remove nodes that are either older than some cutoff or that do not cause a large drop in information about certain nodes of interest. We define the information value of a node in section 3.2.

The most common type of KS is a *network fragment*, which is a belief network that represents a small fragment of knowledge. An example of a fragment is shown in Figure 2. A node in the fragment *matches* a node on the blackboard when the two nodes have the same type, and their argument lists unify. (Recall that nodes in a KS can have logic variables in their argument lists, and the arguments of a node are distinct from its set of possible outcomes.) By default, the precondition for a fragment KS is that at least one of the fragment nodes has a match on the blackboard; however, the KS designer can designate certain nodes that must be matched, or write an arbitrary precondition.

3.1.1 Posting Network Fragments

Fragments are posted to the blackboard by a process that resembles unification. A fragment can be posted to the blackboard if three conditions hold. First, each of the fragment nodes must match a node on the blackboard; a new node can be created on the blackboard if necessary. Second, a single unifying assignment must unify the argument lists of all the fragment nodes with their corresponding blackboard nodes—this merely ensures that a logic variable like ?U refers to the same thing throughout the fragment. Third, no two fragment nodes can match the same blackboard node.

We can think of fragment matching as a bipartite matching problem, as shown in figure 3. On the left side of the bipartite graph are all the blackboard nodes; on the right are all the fragment nodes. A blackboard node and a fragment node are linked if they have the same node type. Now, any bipartite matching in this graph describes a way the fragment could be posted to a blackboard. A fragment node unifies with its neighbor in the matching. If it has no neighbor, a new node is posted to the blackboard.

Once a fragment has been matched to the blackboard, it can be posted. An example of a fragment posting is given in figure 4. A fragment is posted to the blackboard in three steps. First, new nodes are posted if they are required by the match. Second, for every pair of fragment nodes that are linked, a corresponding edge is added to the blackboard. Now the nodes on the blackboard have both their original parents $\mathbf{V}_{BB}$ and the new parents that were specified by the fragment, $\mathbf{V}_F$. Third, since every node V in the fragment has both a conditional distribution in the fragment, $\mathrm{P}(V \mid \mathbf{V}_F)$, and a one on the blackboard, $\mathrm{P}(V \mid \mathbf{V}_{BB})$, these two distributions are combined to get $\mathrm{P}(V \mid \mathbf{V}_F, \mathbf{V}_{BB})$. Ways this can be done are given in the next section.

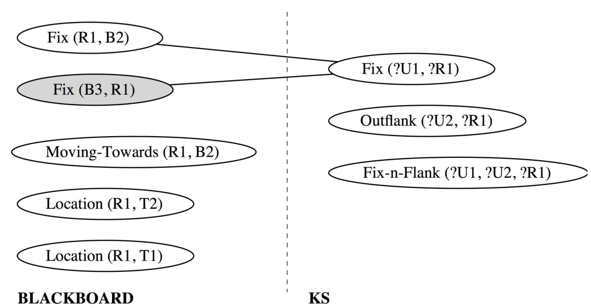

Fig. 3: The bipartite matching problem from matching KS 2 to the blackboard in Figure 1

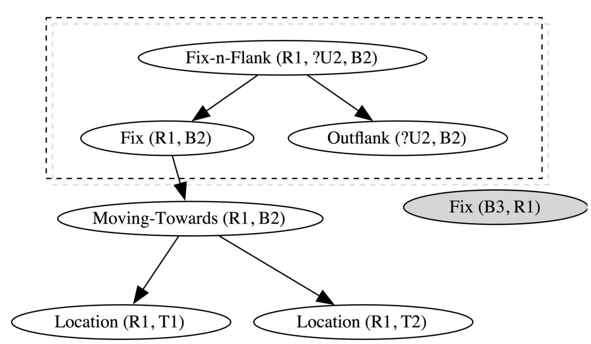

Fig. 4: The blackboard in figure 1 after KS 2 posts

3.1.2 Influence Combination

Since network fragments are themselves belief networks, they specify a complete probabilistic model over their variables. But nodes on the blackboard already have probability distributions. Suppose that some node V has parents

$\mathbf{V}_F$ in the fragment and parents $\mathbf{V}_{BB}$ on the blackboard, so that we have probability distributions $\mathrm{P}(V \mid \mathbf{V}_F)$ and $\mathrm{P}(V \mid \mathbf{V}_{BB})$. When the KS posts, the parents of V will be $\{\mathbf{V}_F \cup \mathbf{V}_{BB}\}$, so we must merge these two models to get a CPT for $\mathrm{P}(V \mid \mathbf{V}_F, \mathbf{V}_{BB})$.

There are several ways to do this. Laskey and Mahoney [7] define several *influence combination methods* to combine conditional probability distributions, one of the principal types being parametric causal models like *noisy-or*. The noisy-or model [10, 11] allows one to compactly specify a conditional distribution when the parents are independent, stochastic causes of the child. The knowledge engineer can specify which combination method should be used for a given node type.

3.2 Control

Many knowledge sources are applicable at any given time, but only a few can be selected to run. This is both because our implementation of AIID runs on a single processor, so only one KS can be run at a time, and because if too many KSs fire, the number of nodes on the blackboard could become too large for probabilistic inference to be tractable. We describe three types of control regimes: a simple one based on KSs' confidence methods, an information-theoretic one based on nodes of interest to the user, and a Bayesian one based on the probability of the blackboard structure given the observed data.

First, KSs can be ordered by confidence. This provides a gross ordering among KSs, but it can be difficult to know in advance, or write a procedure that computes, how useful a KS will be.

Second, certain nodes of the blackboard have more interest to the user. For example, an intelligence analyst may want to know whether two people have communicated, or a military commander may want to know whether a certain attack is a feint. Let $\mathbf{V}$ be a set of these nodes of interest. We can choose to post the knowledge sources that provide the most *information* about $\mathbf{V}$, in the sense of reducing its Shannon entropy. The information gained from firing a KS K is given by $I(K) = H(\mathbf{V}) - H_K(\mathbf{V})$, where $H(\mathbf{V})$ is the entropy of $\mathbf{V}$ before K is posted, and $H_K(\mathbf{V})$ is the entropy afterward. We can compute this directly by temporarily posting K, computing the marginal distribution $\mathrm{P}(\mathbf{V})$, and calculating its entropy.

Since this is probably too expensive to use if many KSs are applicable, we can try to approximate this effect.

We can get a cheap approximation by simply looking at the distance between where the KS will post and $\mathbf{V}$, that is, the length of the shortest undirected path between a node used by the KS and a member of $\mathbf{V}$. Then we prefer the KSs with the shortest distance, on the assumption that nodes that are closer to $\mathbf{V}$ have more influence on its distribution.

Third, we can calculate the value of information and focus attention on information sources and inferences that have high value. Value of information VOI is defined with respect to a utility function, as follows:

$$EU(\alpha|E) = max_\alpha \sum_i U(S_i) Pr(S_i|E, \alpha) \qquad (1)$$

The expected utility (EU) of an action α given evidence E is argmax α of the utility of outcome S_i times the probability of the outcome given the evidence and α.

The value of a new piece of evidence E_j which may take values $e_1, e_2, ...e_n$ is:

$$VI(E_j) = (\sum_k^n Pr(E_j = e_k|E) EU(\alpha_{e_k}|E, E_j = e_k)) - EU(\alpha|E) \qquad (2)$$

This equation defines the value of information E_j as the expected utility of the best action α given E_j minus the expected utility of the best action *without* knowing E_j.

4 Implementation

We have built a prototype of AIID in the domain of military analysis. We simulate military engagements at the battalion level (roughly a thousand troops), using the Capture the Flag simulator [12, 13]. The simulator includes such effects as terrain, fog of war, artillery, combat aviation, and morale.

The data consist of reports about friendly and enemy units, for example, "Unit Red-1 has sound contact with a brigade-sized unit in the east." The prototype does not address the problems of identifying the number and composition of units from individual sightings, which are hard. Rather, our problem is to infer the enemy commander's strategy and, more specifically, the objective of each enemy unit.

The blackboard contains reports about enemy units and hypotheses such as individual unit's objectives, coordinated actions between unit, and theater-wide objectives. We have implemented 30 fragment KSs: for example, one computes the relative combat strength of two opposing units, and others model of military actions such as defeat (i.e., to attack with overwhelming force), outflank, and fix. One procedural KS clusters enemy units by location, and hypothesizes that the units are groups acting in close concert. This kind of geometric reasoning is difficult to implement in a network fragment.

One can see the Bayesian network built by AIID in Figure 5. The image in the top-right of the figure is a screen dump from the Capture the Flag wargaming simulator. One can see three hypothesized groups of units moving from north to south. The associated Bayesian network has four nodes near the bottom of the screen, three of which are hypotheses about the task type of an hypothesized group. Task types are *seize, attrit, fix, penetrate,* and *other*. At the top of the network is a utility node [4] which lays out the utilities of committing the defending forces to axis green, axis blue, or waiting. Below this utility node is a single node that represents a conditional probability distribution over the alternatives that the forces in the north are attacking along axis blue, axis green, or other. The distribution over these alternatives is conditioned on three nodes which represent propositions relevant to the main attack axis. The first of these is that the attack is *determined* along axis blue, green, or other; the second is that the attackers intend to *penetrate* on axis blue, green, or other; and the third proposition looks at whether the attackers are *committing reserves* to axis blue, green, or other.

Each of the evidence nodes in the bottom row of the figure has a value of information, calculated as described earlier. At this state in the simulation, with forces more or less evenly distributed in the north, the most valuable information concern the commitment of reserves.

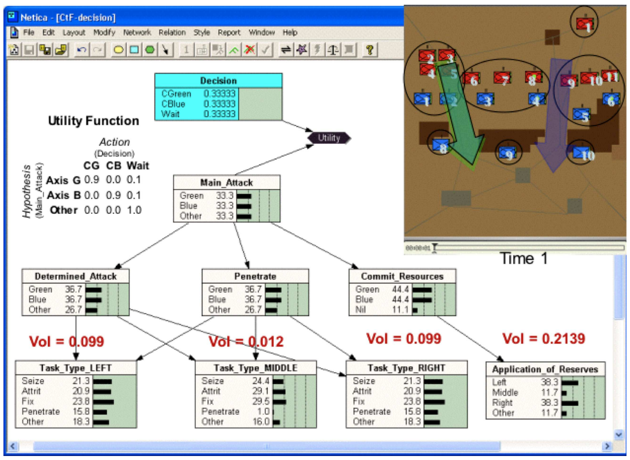

Fig. 5: Network structures at time 1

Some time later the forces in the north have moved further south and are grouped more clearly around axes green and blue (Fig. 6). It is still unclear where the main attack will come, and although AIID is now sure the task type on the left is *attrit*, it is uncertain about almost everything else. The attacker's commitment of reserves remains the most valuable information.

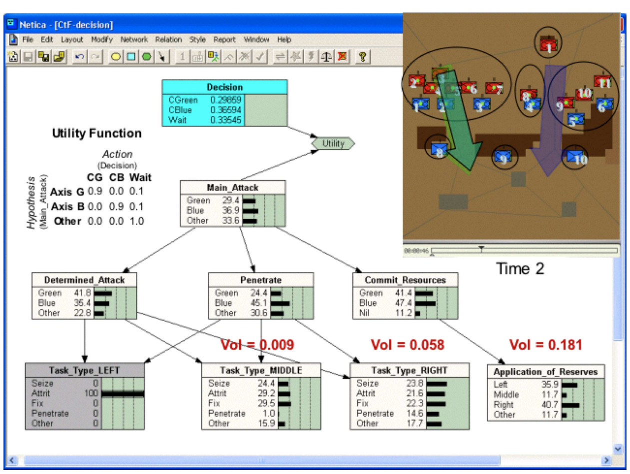

Fig. 6: Network structures at time 2

Later on we see that the network structure has changed: It no longer contains nodes to represent the hypothesis that there is a *middle* group of attacking units; the network now contains nodes only for the left and right groups, corresponding to axes green and blue. It is still unclear which of these axes bears the main attack, and the commitment of resources is still the most valuable information the system could obtain.

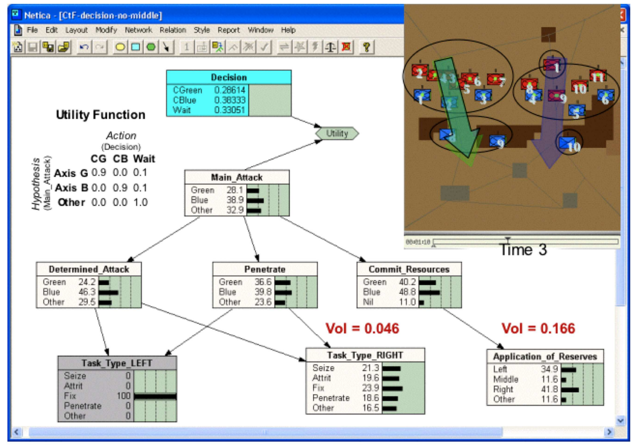

Fig. 7: Network structures at time 3

(The three snapshots are from a real run of AIID and Capture the Flag. However, we lack graphical tools for representing the networks, so we plugged the probabilities and network structures into the Netica tool for visualization purposes.)

5 Conclusion

We have presented an architecture for solving knowledge-intensive problems under uncertainty by incrementally constructing probabilistic models. The architecture synthesizes ideas from the older literature on blackboard systems and the newer literature on construction of belief networks from fragments. Blackboard systems are a method of incrementally building symbolic models, while the network fragment systems incrementally build probabilistic models. The contribution of the current work is, in making the connection between the two literatures, to point out that both have good ideas the other hasn't used: probabilistic models have a principled method of reasoning under uncertainty, while blackboard systems have focused on controlling the search through the space of possible models.

It would be natural to extend the architecture to handle influence diagrams, the largest requirement being influence combination methods for decision and utility nodes. The most important open questions in this architecture are better methods of evidence combination and evaluating different control methods. We would also like to apply this architecture to other domains, both within intelligence analysis and common-sense reasoning.

References

[1] L.D. Erman, F. Hayes-Roth, V.R. Lesser, and D.R. Reddy. The HEARSAY-II speech understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Survey*, 12:213–253, 1980.

[2] H. Penny Nii. Blackboard systems. In Avron Barr, Paul R. Cohen, and Edward A. Feigenbaum, editors, *The Handbook of Artificial Intelligence: Volume IV*, pages 1–82. Addison-Wesley, 1989.

[3] Norman Carver and Victor Lesser. The evolution of blackboard control architectures. In *Expert Systems with Applications 7*, pages 1–30, 1994.

[4] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Addison-Wesley, 2002.

[5] Robert P. Goldman and Eugene Charniak. A language for construction of belief networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):196–208, 1993.

[6] Daphne Koller and Avi Pfeffer. Object-oriented Bayesian networks. *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, pages 302–313, 1997.

[7] Kathryn Blackmond Laskey and Suzanne M. Mahoney. Network fragments: Representing knowledge for constructing probabilistic models. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97)*, San Mateo, CA, 1997. Morgan Kaufmann.

[8] Avi Pfeffer, Daphne Koller, Brian Milch, and Ken T. Takusagawa. SPOOK: A system for probabilistic object-oriented knowledge representation. In *Proceedings of the 14th Annual Conference on Uncertainty in AI (UAI-99)*, pages 541–550. Morgan Kaufmann, 1999.

[9] Kevin P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, U.C. Berkeley, July 2002.

[10] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[11] Sampath Srinivas. A generalization of the noisy-or model. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI-93)*, San Francisco, 1993. Morgan Kaufmann.

[12] Marc Atkin, Gary W. King, David Westbrook, Brent Heeringa, Andrew Hannon, and Paul Cohen. SPT: Hierarchical Agent Control: A framework for defining agent behavior. In *Proceedings of Fifth International Conference on Autonomous Agents*, pages 425–432, 2001.

[13] Marc Atkin, David L. Westbrook, and Paul Cohen. Domain-general simulation and planning with physical schemas. In *Proceedings of the Winter Simulation Conference*, pages 464–470, 2000.