Network Fragments for Knowledge-Based Construction of Belief Networks

Kathryn Blackmond Laskey
Department of Systems Engineering and C^{3}I Center
George Mason University
Fairfax, VA 22030
klaskey@gmu.edu

Suzanne M. Mahoney
Information Extraction and Transport, Inc.
1730 N. Lynn Street, Suite 502
Arlington, VA 22209
suzanne@iet.com

Abstract

This paper describes an example problem of knowledge-based belief network construction (Goldman, Breese and Wellman, 1992; Goldman and Charniak, 1993; Laskey, 1990). Knowledge-based model construction requires declarative representations for encoding modular, abstract, repeatable domain relationships, and procedures for instantiating and combining these knowledge elements to form models for particular problem instances (Egar and Musen, 1993; Regan and Holzman, 1992). Recent work in knowledge representation (Laskey and Mahoney, 1997; Mahoney and Laskey, 1996; Koller and Pfeffer, 1997) represents domain relationships as fragments of belief networks. The object-oriented framework is natural for this purpose, with its ability to represent abstract types with associated structure and methods, inheritance, and encapsulation. The example problem is drawn from the domain of military situation assessment. Although highly simplified, the example problem illustrates many of the issues that must be tackled to apply the technology to more complex problems. We discuss roles for mixed-initiative reasoning in knowledge-based model construction for military situation assessment.

1 MILITARY SITUATION ASSESSMENT

Information is a force multiplier: a military force that knows the enemy's capability and intentions can outfight a larger and better-equipped force. For example, the surprise "Hail Mary defense" of the Persian Gulf conflict depended on the U.N. forces denying the Iraquis information on U.N. troop movements. Situation assessment is the process of inferring the type, location, and activity patterns of forces in a military situation. Situation assessment involves the incorporation of uncertain evidence from a diverse variety of sources. These include photographs, radar scans, and other forms of imagery (IMINT); electronics intelligence (ELINT) derived from characteristics (e.g., wavelength) of emissions generated by enemy equipment; signals intelligence (SIGINT) derived from the characteristics of signals sent by the enemy; and reports from human informants (HUMINT).

These sources must be combined to form a model of the situation. The purpose of the technology described in this paper is to construct a situation-specific Bayesian network (Mahoney, 1998) to model a military situation. The situation-specific network is constructed to respond to queries about *focus* variables using contextual knowledge and available evidence. The situation-specific network is constructed from a knowledge base of generic knowledge about military entities, their behavior, their interrelationships, and the reports from which their presence can be inferred. The situation-specific network represents knowledge about which military units are present, where they are located, what activities they are engaged in, and what they are likely to do next. Domain knowledge is represented as network fragment classes. Instances of these fragment classes represent particular entities in the situation. For example, a generic Scud launcher battery fragment is instantiated to reason about a particular Scud launcher that has been observed. Fragment objects have *identifying attributes* which are bound to particular values when an instance is created. Fragments also have associated *influence combination methods* which specify how to construct the local distribution for a node from information contained in multiple fragments (Laskey and Mahoney, 1997).

The task involves a number of subtasks, described below. The terminology is borrowed from the multiple hypothesis tracking community, because the problem we are addressing can usefully be viewed as a generalization of multiple hypothesis tracking.

1. *Data association.* Given a new report and a set of hypothesized entities (military units), data association identifies which of the entities might have given rise to the report. In single-hypothesis systems, data association picks a single, best-fitting entity and assumes the report came from that entity. In multiple-hypothesis systems, data association identifies a set of objects that could have given rise to the report.
2. *Hypothesis management.* Hypothesis management is the problem of postulating new entities and pruning existing hypotheses. If none of the existing entities provides a very good "match" for a report, a new entity may be created to explain the report. Alternatively, new evidence may so weaken support for an existing

hypothesis that the system may no longer need to maintain it as a possibility.

3. *Inference and projection.* Once reports have been assigned to entities, they are processed to infer the state of the entities and to project their state forward to the time of the next set of reports. For traditional tracking, this usually involves updating dynamic information such as position and velocity. In our models the relevant state information concerns activity, activity transitions, location, status, hierarchical organization, etc.

The models we are developing are intended to provide support for intelligence analysts, not to completely automate the military situation assessment process. Construction of a situation specific network will be a mixed initiative task. Identification of relevant fragments is driven by observed evidence and by user queries which identify variables of interest. We anticipate that on problems of any complexity there will be situations in which incorporating all relevant fragments will be infeasible due to computational considerations. The user may provide guidance about which hypotheses to focus on and which to prune. Another important issue is communication of model results to the user. The purpose of this paper is to put forward an interesting real-world problem for which mixed initiative reasoning is required, and to stimulate discussion about the role and mechanisms of mixed-initiative reasoning in this problem.

2. INFERENCE ABOUT LOCATION

Spatial and temporal reasoning are key issues in military situation assessment. The example presented in this paper is simplified to ignor the time dimension, but does involve reasoning about the location of military entities. This section provides a brief overview of our framework for reasoning about location.

Reports are associated with a spatial region within which the entity giving rise to the report might be located. Typically there is also likelihood information, with for example areas at the edge of the region having less likelihood than areas near the center of the region. Information about the probable location of hypothesized entities must be represented so that entities can be associated with new sensor reports. These new reports can then be used to update and refine information about location.

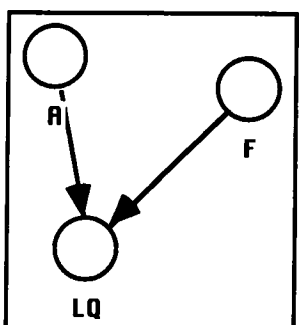


Figure 1: Simplified Top-Level Abstracted Network Fragment

Each of our models has a variable called Location Quality. This variable is conditioned on a particular unit type and a particular location. Its identifying attributes are an entity tag and a location. For example, the location quality for a hypothesized Scud missile unit SCUD <E123> located at <L1> is [SCUD <E123> Location Quality <L1>]. Under the hypothesis that <E123> is an SA6, its location quality would be represented by the random variable [SA6 <E123> Location Quality <L1>]. In simplified form, the network fragment for location quality (LQ)in all our models can be summarized by Figure 1, taken from (Laskey, 1995).[1]

In this fragment, the node A stands for the unit's type-specific activity and F stands for spatial/terrain/location (STL) features for the specific location. Examples of STL features include accessibility, terrain characteristics such as slope and roughness, distance from related units, etc. Location quality models can be quite complex, but this is not the subject of the present paper. Both activity and location quality are conditioned on a particular unit type. Thus, there is a location quality variable for each possible type/activity/location combination for each hypothesized entity. These variables are designated [UNIT-TYPE <Entity> Location Quality <Location>] (e.g., [SA6-Battery <E123> Location Quality <L1>]).

In our models, the location quality variables have two states: "Good (likely to be here)" and "Bad (not likely to be here)." The semantics for location quality is (Laskey 1995, 1996):

- Different type/activity/location combinations for each entity are exclusive. That is, the entity must be exactly one type, performing exactly one activity, at exactly one location.
- The location of an entity is conditionally indpendent of its type and activity given its type/activity specific location quality. That is, the state of the location quality variable (good vs. bad) carries all the information about unit type and activity necessary to infer its location.
- Units are not located at bad locations. All good locations are equally likely.

This semantics implies a method for combining fragments such as Figure 1 into a network fragment to be used for inferring location. Consider an entity, designated <E>, whose unit type, activity, and/or location is unknown. According to the way we have constructed our models, there will be a location-specific location quality random variable for each unique unit type and location combination. Let (T1, L1), ... (Tn, Ln) denote the type and location hypotheses under consideration, assumed exclusive and exhaustive.[2] In our standard naming conventions, the location quality variable corresponding to (Ti, Li) is designated [Ti <E> Location Quality <Li>]. For brevity, we abbreviate these random variables as LQi. As currently constituted, our models represent the probability distribution of LQi, which depends on the unit's activity and STL features as shown abstractly in Figure 1.

---

[1] The networks in the body of this paper were generated using Ergo for convenience in cutting and pasting. This necessitated creating abbreviated names in place of our standard naming conventions.

[2] Note that it may be the case that Ti=Tj or Li=Lj for some i≠j, as long as both type and location are not the same.

Location inference requires combining the separate fragments from Figure 1 into a combined fragment relating the locations to each other. This requires a fragment expressing the location of a unit given the location qualities for each location/type combination. Figure 2 shows the structure of this fragment. In the figure, our standard designation for location random variable `[Location <E>]` is abbreviated Loc.

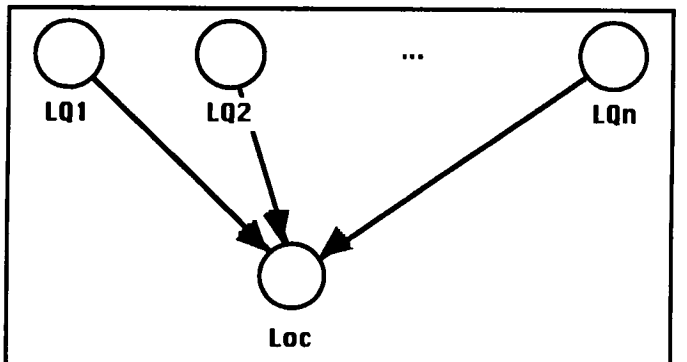


Figure 2: Network Fragment for Location

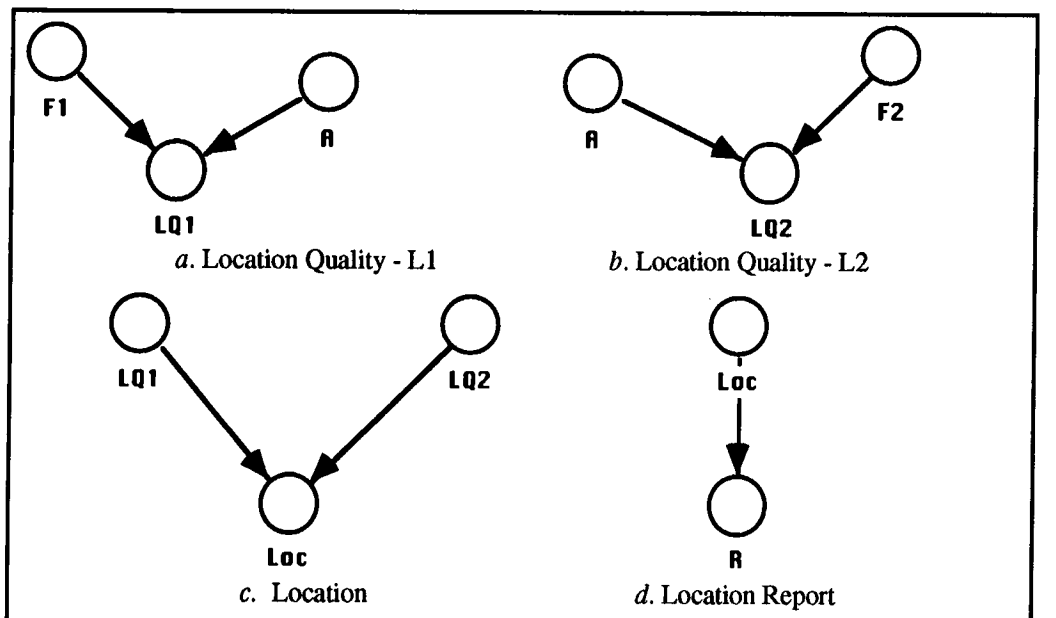
The states of the `[Location <E>]` variable are the unique locations among the Li, along with the special state *not_here*. The conditional probability distributions for the Loc random variable express the knowledge that the different locations are exhaustive and that units may be placed only at good locations. This is implemented by defining an influence combination method for the random variable `[Location <E>]` given its parents `[Ti <E> Location Quality <Li>]`, i=1,...n.

- The probability of state L of `[Location <E>]` depends only on the states of parent variables `[Ti <E> Location Quality <Li>]` for which Li=L. The probability is α if any such variable has value *good* and 0 otherwise. This implements the constraint that units may be located only at locations with good location quality, while assigning all states meeting the constraint equal likelihoods.
- The special state *not_here* is assigned the remaining probability $1\text{-}k\alpha$, where k is the number of states of `[Location <E>]` at which it is possible to locate the entity, given the states of its parent variables.

To illustrate, consider the example of a hypothesized SA6 battery designated <E>. A set of network fragments for reasoning about the location and activity of this unit is shown in Figure 3. The activity of this unit would be designated by the random variable `[SA6-Battery <E> Activity]`, abbreviated in the figure as A. For simplicity of exposition we assume there are only two hypothesized locations, designated L1 and L2. The location quality variables `[SA6-Battery <E> Location Quality <Li>]` are abbreviated LQi. Figures 3a and 3b represent knowledge about location quality given activity and terrain features. The variables F1 and F2 represent STL features of L1 and L2, respectively, that are relevant to location quality.

Our fragments knowledge base contains knowledge relating terrain features to location quality. Figure 3c relates location quality to location. Its conditional probability table is constructed as described above. Note that the specific value of α is irrelevant to the computation. What matters is that all possible states are assigned the same likelihood. The final fragment for location inference is a report which establishes the set of possible locations for an entity and assigns a likelihood to each. The Report node has two states, designated "Yes" and "No." It is the likelihoods of the "Yes" state that affect the inference, and again it is only the likelihood ratios that affect inference.

These fragments are combined using simple combination to form a Bayesian network for reasoning about the location of the entity. In general, there will be very many possible locations for a given entity. Typically, locations would be represented by points on a suitably fine grid. All points within a sensor error ellipse would be considered possible locations for a given entity. As the number of possible locations increases, the conditional probability tables become very large and standard Bayesian network methods become intractable. Efficient implementation of location inference is a key issue not addressed in this paper.



a. Location Quality - L1

b. Location Quality - L2

c. Location

d. Location Report

Figure 3: Fragments for Inferring Location

3. EXAMPLE INFERENCE PROBLEM

We illustrate our network construction approach with an example. The example is simplified but captures important features of the problem. Section 4 below abstracts generic knowldege structures and a generic network construction methodology from the example in this section.

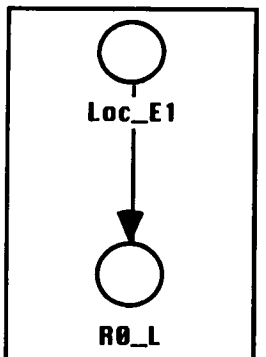


Figure 4: Report Fragment for R0

Assume that an imagery report, assigned report designator R0, is received. The report indicates that a unit of unknown type is located within a given error ellipse. The system is tasked with constructing a probability model for reasoning about the unit's type, location and activity.

The first step in the process is data association. The job of data association is to determine whether any already hypothesized entities could have given rise to the report. We assume in our example that none is found. A new unit designator, E1, is therefore created to refer to the entity giving rise to this report. The possible locations of the unit are indicated by the error ellipse for R0. We assume there are three possible locations, designated L1, L2, and L3. A fragment relating location to report is created as described above in Section 2 and shown in Figure 4. The variable Loc_E1 abbreviates our standard notation `[Location <E1>]`. The report variable is abbreviated in the figure as R0_L, to designate that part of the information content of R0 relating to location. Our standard designation for this report would be `[IMINT <R0> Location]`.

Next, the system constructs the rest of the fragments to be combined to form the inference network for <E1>. To do this, the unit-type hypotheses consistent with the report must be considered. The system analyzes the imagery report to determine the possible unit types that could give rise to the report. For the purposes of our example we make the simplified assumption that there are two possible unit types, an SA6 battery and a Scud battery. To represent the unit type hypotheses, the system creates a random variable `[Unit-Type <E1>]`, which we abbreviate as UT_E1, having states Scud and SA6. It also creates activity random variables `[SA6 Activity <E1>]` and `[Scud Activity <E1>]`, abbreviated SA_A_E1 and Sc_A_E1. Location quality variables `[UNIT-TYPE <E1> Location Quality <Li>]`, abbreviated SA_LQ_E1_Li and Sc_LQ_E1_Li, are created for each of the unit type / location hypotheses. Then the system constructs the location fragment shown in Figure 5.

Now comes the model for reasoning about activity and location quality. The generic fragment shown in Figure 5 is retrieved and instantiated for each of the unit-type / location combinations. Here we must address another issue of semantics, that of the meaning of, for example, the `[SA6-Battery Activity <E1>]` random variable under the hypothesis that E1 is a Scud battery. We can model this with *context-specific variables* -- that is, random variables that have meaning only within a particular context. The random variable `[SA6-Battery Activity <E]` is defined to be the activity of <E123> conditional on its being an SA6 battery. When the unit is not an SA6 battery this variable has no meaning. We modeled this by augmenting the state space of our context-specific variables with the value NA, which is defined to have probability 1 in contexts in which the variable has no meaning. The conditional distribution for the location quality variable in the generic network of Figure 5 is defined as in our current model when the state is one of the standard activity states (e.g., move, hide, operate) for the random variable. Conditional on NA, location quality is bad with probability 1.

The network of Figure 6 differs from the networks of Figures 1 and 3 in that the STL features (represented in Figure 1 by the placeholder node F) have been omitted. For purposes of this example, assume that STL features have been marginalized into the distribution for the location quality variable. Explicitly including STL features in the model would be straightforward, but the example networks would be more complex, obscuring essential features of the reasoning process.

Finally, fragment instances relating unit type to activity are constructed for each of the Scud and SA6 activity variables, using the generic fragment shown in Figure 7. Unit types are *a priori* equally likely. Of course, information on the relative frequencies of different types of units can be used if available. If the report itself contains unit type information (such as features on the image indicating a Scud as distinct from an SA6), this would be modeled by adding a second report variable R0_UT. Multi-component reports are discussed in more detail below.

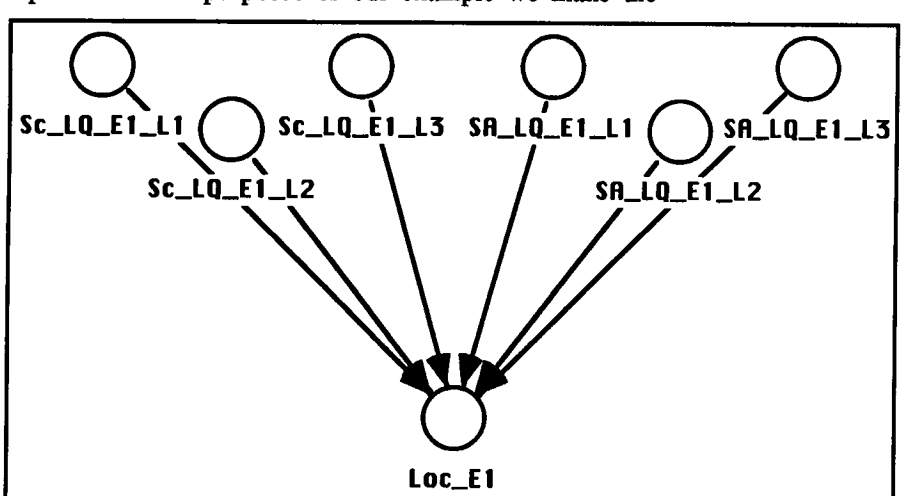
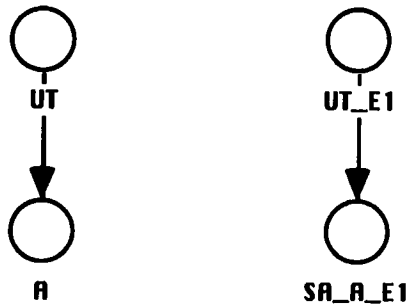


Figure 5: Location Fragment for E1

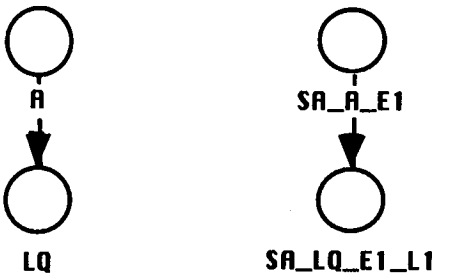


a. Generic Fragment b. Instance: SA6 Battery

Figure 6: Fragments Relating Activity to Location Quality

Given that the unit type is consistent with the activity variable (i.e., Scud activities given that unit type is Scud), the probability distribution on activities is taken from our existing knowledge base. Given a unit type that does not match the activity variable (e.g., SA6 activities given that unit type is Scud), the activity is NA with probability 1.

The fragments of Figures 5 and 6, together with the six different instances of the fragment of Figure 5 and two instances of Figure 7, are combined to give the network of Figure 8. We condition on the R0_L=Yes and propagate



a. Generic Fragment b. Instance: SA6 at L1

Figure 7: Unit Type / Activity Fragment

beliefs to see the results displayed in the figure. The report likelihood favored L1 and L3 over L2. These locations were more conducive to Scuds than to SA6's. Therefore, the final belief is about 60:40 for a Scud. This is reflected in the length of the appropriate belief bars in the figure, and the posterior beliefs for locations L1, L2, and L3 are about 48:04:48.

To summarize, report processing and network construction follows the following steps:

1. *Data association.* Determine whether there are any existing units consistent with the report. There are none.
2. *Hypothesis generation.* Postulate new entity E1 for unit giving rise to the report. This step is necessary given that no existing unit could have given rise to the report.
3. *Network fragment retrieval and construction.* Retrieve generic fragments and construct fragment instances according to standard fragment construction operators. The following fragment instances were constructed:
   - Report fragment (Figure 4);
   - Location fragment (Figure 5);
   - Six activity/location quality fragments (Figure 6);
   - Two unit type/activity fragments (Figure 7).
4. *Fragment combination.* Combine all fragments into a single belief network (Laskey and Mahoney, 1997).
5. *Evidence processing.* Declare evidence R0="Yes."

The result is a belief network with posterior beliefs representing current knowledge about the situation given report R0.

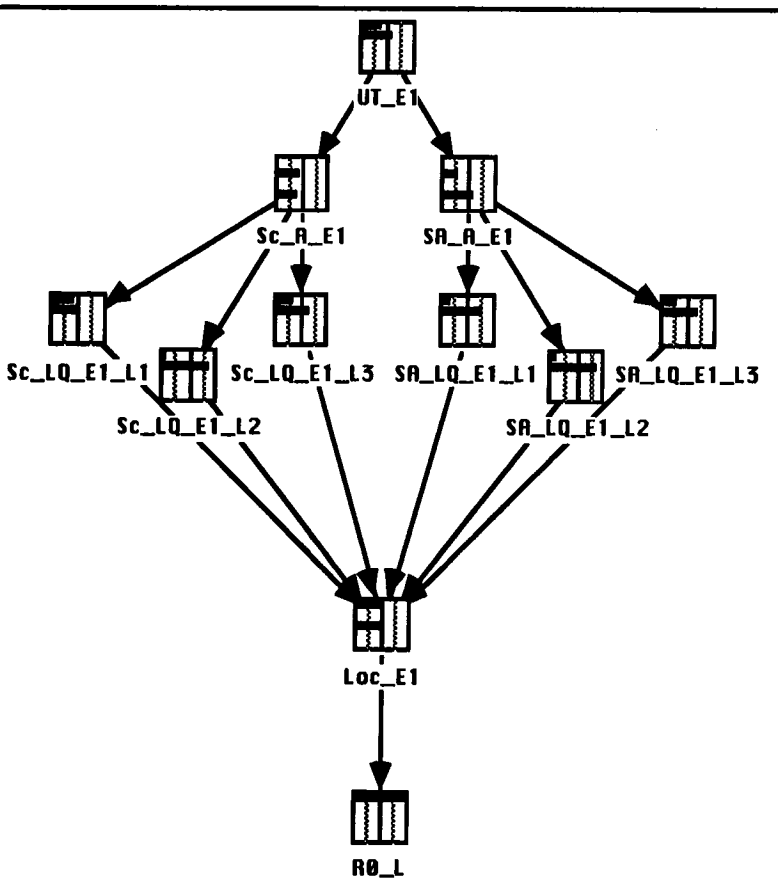
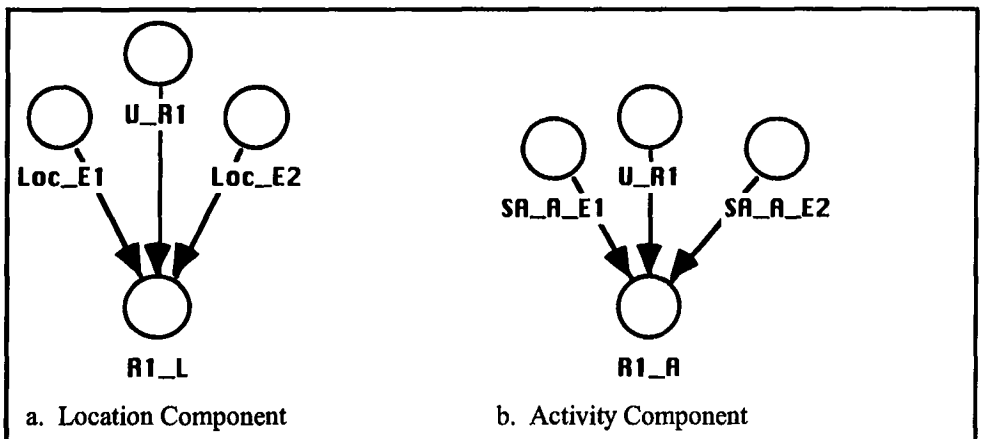


Figure 8: Belief Network for Processing Imagery Report R0



a. Location Component b. Activity Component

Figure 9: Report Fragments for R1

3.2 SIGINT Report: SA6 Battery Operating

The example continues with receipt of report R1, a radar report indicating a Straight Flush radar in operating mode. Locations consistent with this report are the same as for R0. However, this report favors L2 over L1 and L3 by a ratio of 9:1.

Now we have to consider the problem of reasoning about whether R0 and R1 refer to the same unit. On the one hand, the locations are consistent. On the other hand, the locations favored most by R0 favor the Scud hypothesis we know that R1 came from an SA6.

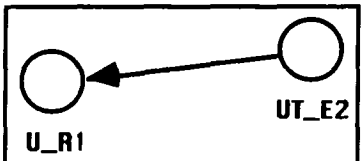


Figure 10: Unit Type Fragment

Our first step is to create a random variable to reason about whether E1 or some new unit, which we will designate E2, gave rise to report R1. We call this node `[Unit for <R1>]`, or U_R1 for short. We construct the fragment of Figure 9a to represent the report location likelihood conditional on the location of the unit giving rise to the report. The influence combination function for this fragment is the "select one" function: The report likelihood depends on the values of Loc_E1 if the report came from E1 and on Loc_E2 if the report came from E2. The same generic structure is used to create a fragment for the activity component of report R1. This report is assumed to be consistent only with an SA6 in operating mode. Therefore, the report likelihood is 1 given that the SA6 activity of the unit giving rise to the report is operating and zero otherwise.[3]

The entity E2 is defined as the entity hypothesized to explain report R1. It therefore does not exist (i.e, has type NA) if R1 was generated by some other entity (in this case, the only candidate is E1). This knowledge is encoded by the "existence" fragment shown in Figure 10. This fragment declares that U_R1 is E2 with probability 1 if E2 is of any type other than NA. When UT_E2=NA then U_R1 places probability zero on E2 and equal likelihood on all other unit hypotheses being considered for the report (in this case only E1). The variable UT_E2 is assigned the distribution of .9 for NA and .1 for SA6, reflecting a bias that observations should be assigned to existing units when they can be explained by existing units.

The remainder of the network for reasoning about R1 is constructed exactly as for R0, except that only one activity variable and only one set of location quality variables is required because only one unit type is hypothesized.

There is one additional component we need to add, which is the constraint that two different units may not be placed at the same location. This is represented by the node Ex_E1_E2, which represents the statement that locations for the two entities must be exclusive. This report variable is given likelihood 1 when its parent variables are in different states and 0 when they are in the same state.

The resulting belief network is displayed in Figure 11. The reports are processed by declaring state "Yes" on variables R0_Loc, R1_Loc, R1_A, and Ex_E1_E2. Note that there is a very high belief (about .996) that E2 has type NA; in other words, that both reports refer to the same entity. This is because of the inherent bias to explain reports using existing entities, and the fact that report R1 was consistent with the existing entity.

3.3 SIGINT Report: Scud Launch

Finally, suppose a third report is received indicating a Scud launch. Again, the report contains both location information (with the same three possible locations) and activity information (a Scud battery in Operate mode, where Launch is considered to be a sub-category of Operate and we are representing activity only at the higher level of granularity where the states are Move, Operate, Hide, and NA). The steps in augmenting the network to process this report are exactly in parallel to the steps described in Section 3.2 above. The resulting network is shown in Figure 11.

From the network of Figure 12 we see that E1 is believed to be a Scud by ratio 83:17, E2 is believed to exist and be an SA6 by ratio 83:17, and E3 is believed to be NA by ratio 83:17. Thus, the most likely hypothesis is that E1 is a Scud, located at either L1 (probablity .57) or L3 (probability .43 given E1 is a Scud), E2 is an SA6 located at L2 (probability .91 given E1 is a Scud), and E3 is NA (probability greater than .999 given E1 is a Scud). The next most likely hypothesis is that E1 is an SA6, located at L1, L2, or L3

[3] Note that declaring the SA6 activity of E1 to be Operate has the effect of declaring E1 to be an SA6 battery, because the SA6 activity of E1 is NA with probability 1 under the Scud hypothesis.

with probabilities .17, .58, and .25, respectively, E3 is a Scud located at L1 or L3 with probabilities .55 and .44, respectively, and E2 is NA with probability .999. All other hypotheses have very little probability.

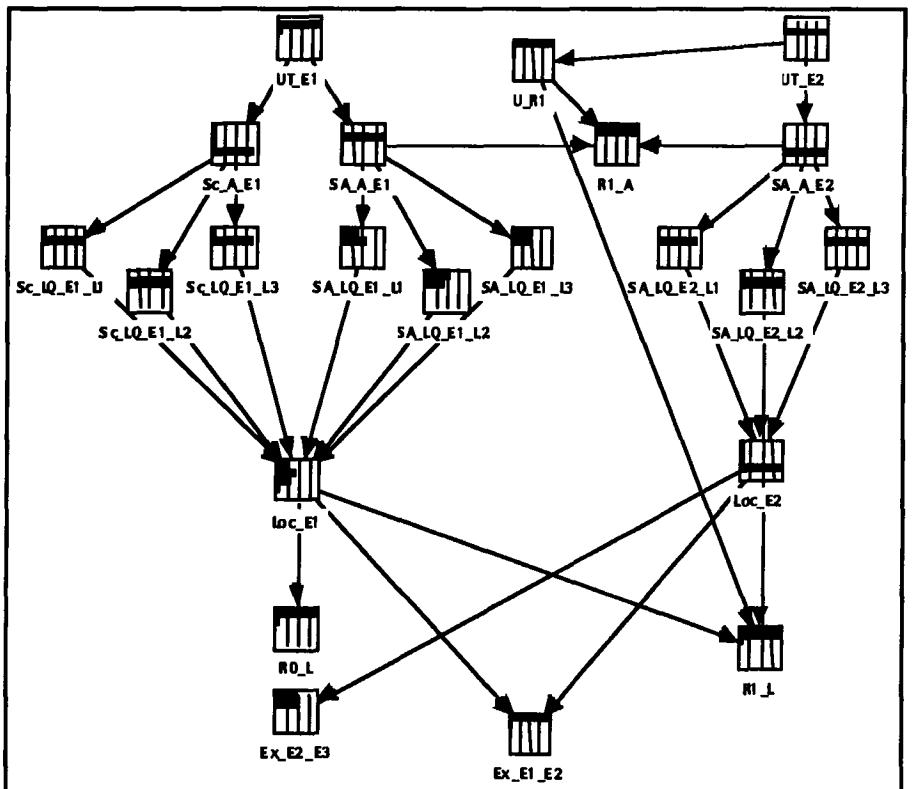


Figure 11: Belief Network for Processing Imagery Report R0 and SIGINT Report R1

3.4 Summary: Steps in Report Processing and Network Construction

The steps given in Section 3.1 are now seen to be the standard steps in processing a report, augmented by some additional hypothesis and state space management steps.

1. *Data association.* Determine whether there are any existing units consistent with the report.
2. *Hypothesis generation.* If no units match unit type and location constraints, postulate a new entity giving rise to the report. If the report can be matched with an existing unit, determine whether it to postulate a new entity or whether simply to match the report only with existing entities. (In the example of this section we always postulated a new entity. We might, for example, not have postulated a new entity for either E2 or E3, sticking with a two-entity model.
3. *State space management and hypothesis pruning.* Determine the possible unit types for each of the hypotheses under consideration. This could involve adding states to a node. For example, we might have processed report R2 by adding the state Scud to entity E2 rather than creating a new entity E3. This could also involve pruning states (if SA6 became unlikely enough for E1 we might prune it) or even pruning whole networks (if state NA became sufficiently likely we might remove the entire network for Entity E3).
4. *Network fragment retrieval and construction.* Retrieve generic fragments and construct fragment instances according to standard fragment construction operators. The following fragment instances are constructed (examples from the report in parentheses):
   - Location report fragment (Figures 4 and 9a);
   - Location fragment (Figure 5);
   - Activity report fragment (Figure 9b)
   - Activity/location quality fragments (Figure 6);
   - Unit type/activity fragments (Figure 7);
   - Unit type fragment (Figure 10; needed only if there are multiple report hypotheses having different unit type implications).
5. *Fragment combination.* Combine all fragments into a single network.
6. *Evidence processing.* Declare evidence "Yes" on all report nodes (alternatively, process all report nodes as soft evidence likelihood multipliers).

4 KNOWLEDGE STRUCTURES

The example presented in this report can be used to define generic network fragment objects to be instantiated and combined to form a situation-specific network (Mahoney, 1998) for reasoning about unit type and location given reports. The following generic categories of network fragments are defined:

1. *Reported location fragments.* Examples are given in Figures 3d, 4 and 9a. A report fragment associates the location component of a report with a random variable representing the locations consistent with the report. The report likelihood incorporates information about differential likelihoods of locations within the range of the sensor in queston.

   Focus variable: `[SOURCE-TAG <R> Location]`

   Parents: `[Location <E>]`

   for all hypothesized entities

   `[Unit for <R>]`

   selection variable

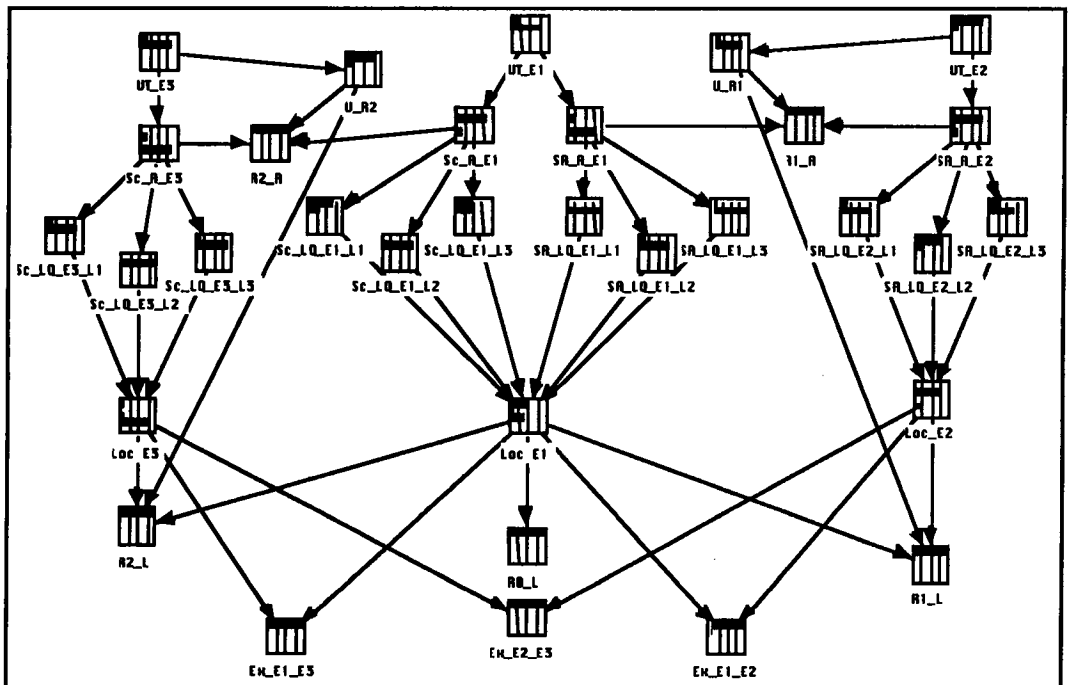


Figure 13: Belief Network for Processing Imagery Report R0, SIGINT Report R1, and Ballistics Report R2

The distribution for the focus variable is defined as follows. First, the system creates an instance of the generic fragment [Location <Ei>] → [SOURCE-TAG <R> Location] for each unit Ei that might have given rise to the report. Distributions for these fragments are computed using a report likelihood method associated with the source type.[4] These fragment distributions are combined using the *pick one* influence combination method. The pick-one method combines a set of distributions over a common focus variable. A *selection variable* is defined whose state space is the set of parents to be combined (i.e., the state space of [Unit for <R>] is the set of Ei). The pick-one combination method uses the state of the selection variable to select the associated distribution. Thus, a pick-one distribution is independent of all variables except the one corresponding to the state of the selection variable.

2. *Location fragment*. An example is given in Figures 2, 3c, and 5. This fragment relates a location quality variable for each possible location/unit type combination to the location random variable.

   Focus variable: [Location <E>]

   Parents: [UNIT-TYPE <E>
   Location Quality <Li>]
   one for each location & type

   The semantics of this variable is that it represents a constraint that reports may be generated only from units at locations with "good" location quality, and all "good" locations are equally likely to give rise to a report. Thus, the probability that location quality is "good" can be thought of as being proportional to the *a priori* probability that a reported entity will be located at that point. This semantics gives rise to a natural way to do knowledge engineering: probability ratios of "goodness" for different STL feature/activity combinations can be thought of as ratios of the probability that a unit would be located at this point.

   A method for constructing the distribution for this variable is described above in Section 2 and illustrated in Table 1.

3. *Reported activity fragments.* An example is given in Figure 9b. This fragment relates an activity report to the activity of the associated unit. Its distribution encodes the likelihood of the report given different activities. This variable is included only when a report gives information about activity. It does not appear for R0 because the imagery report contained no activity information.

   Focus variable: [SOURCE-TAG <R> Activity]

   Parent: [UNIT-TYPE Activity <E>]
   [Unit for <R>]
   selection variable

   This fragment is defined in the same way as the location report fragment. Instances of the generic fragment [UNIT-TYPE Activity <E>]→ [SOURCE-TAG <R> Activity] are created for each unit Ei that might have given rise to the report and combined using pick-one influence combination.

4. *Activity/location quality fragments*. Examples are given in Figures 3a, 3b, and 6.

   Focus variable: [UNIT-TYPE <E>
   Location Quality <L>]

   Parent: [UNIT-TYPE Activity <E>]
   [STL-FEATURES <Li>]

[4] For example, the likelihood might be defined as an error ellipse, with points in the center of the ellipse having higher likelihood than points at the edge.

Generic activity/location quality fragments already exist in our MOB fragment database for the MOB models we elicited. They encode knowledge provided by our MOB experts. They model the factors associated with good locations for different unit type/activity combinations.

5. *Unit type/activity fragments.* An example is given in Figure 7. This fragment relates a unit type specific activity variable to a parent variable which collects all unit type hypotheses under consideration for the entity. One such fragment instance is created for each unit type.

   Focus variable: [UNIT-TYPE Activity <E>]

   Parent: [Unit-Type <E>]

   The distribution for the focus variable is constructed as follows. Our MOB model for each given UNIT-TYPE (e.g., SA6 Battery, Scud Battery, MRL battery, SA6 regiment, etc.) contains the generic random variable [UNIT-TYPE Activity <E>] as a root node with a defined prior distribution. This distribution is used for the conditional distribution of [UNIT-TYPE Activity <E>] given [Unit-Type <E>]=UNIT-TYPE. The state NA of [UNIT-TYPE Activity <E>] has probability 1 given [Unit-Type <E>]≠UNIT-TYPE.

6. *Existence fragment.* An example is given in Figure 10. This fragment encodes the constraint that a unit has type NA if and only if the report it was hypothesized to explain was in fact generated by a different entity. When no unit other than the one hypothesized can explain a report, the state NA is not considered for the unit type and no existence fragment is necessary (this was the case for R0).

   Focus variable: [Unit for <R>]
   Parents: [Unit-Type <E>]
   where <E> was hypothesized to explain <R>

[Unit for <R>] has value <E> with probability 1 when [Unit-Type <E>] has any value other than NA and 0 when [Unit-Type <E>] has value NA. In the latter case, all units consistent with the report are assigned equal likelihood.

5. DISCUSSION

This paper describes a generic knowledge representation for constructing situation specific networks for reasoning about unit type, activity and location. Belief network construction algorithms operate on a knowledge base of network fragment objects, which are used to construct fragment instances that are incrementally added to the situation-specific network as report processing proceeds.

We can see that the network of Figure 8 is inferentially complete with respect to the additional elements that were added with the processing of additional reports. That is, the networks of Figures 11 and 12 would give the same inference results as the smaller network of Figure 8. Thus, we can think of the generic fragments in our knowledge base as implicitly representing a much larger network, most of which is irrelevant to reasoning about the problem at hand. Only those parts of the network needed to process existing reports need be brought into the model workspace at any given time. As new reports are added, the relevant network fragments are constructed from the knowledge base and added to the situation-specific network. A formal definition of situation-specific networks and a proof of inferential completeness will be available in a forthcoming paper.

The example presented in this paper illustrates the knowledge representation framework and some of the knowledge structures we have developed for the task of constructing situation-specific networks for the military situation assessement problem. We described the sequence of steps by which a situation-specific network for this specific example is constructed. We are currently developing search methods and network construction algorithms for building situation-specific networks. Our methods will build on earlier work in the area (Goldman and Charniak, 1993; Egar and Musen, 1993; Regan and Holzman, 1992). The human analyst will participate in network construction in the following roles:

- The *focus* variables, that is, the variables that are the object of the query, are provided by the user. Our observations of human intelligence analysts have made it clear that the query is a major determinant of how intelligence information is processed by human analysts. The query determines the focus of attention, the search strategy, and the hypotheses considered by the analyst. Similarly, the focus variables are used to direct search in our construction algorithms.
- In hypothesis management, the user may provide guidance about which hypotheses to prune and which to explore in more depth.
- The user may provide direction about the appropriate level of granularity.
- A major issue for the development of systems to support human analysts is the interface for communicating results to users and providing human inputs to the system.

Acknowledgments

The research reported in this paper was sponsored by DARPA and the U.S. Army Topographic Engineering Center under contract DACA76-93-0025 to Information Extraction and Transport, Inc. Many thanks to Jim Myers for his patient and painstaking work implementing the example and debugging the knowledge structures. Thanks are due also to Bruce d'Ambrosio for helpful feedback.

References

Egar J. and Musen M., Graph Grammar Assistance for Automated Generation of Influence Diagrams. *Uncertainty in Artificial Intelligence: Proceedings of the Ninth Conference,* 1993.

Goldman, R.P. and Charniak, E. (1993) A Language for Construction of Belief Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(3),* 196-207.

Koller, D. and Pfeffer, A. Object-Oriented Bayesian Networks. *Uncertainty in Artificial Intelligence: Proceedings of the Thirteenth Conference*, 1997.

Laskey, K.B. A Probabilistic Reasoning Environment. *Uncertainty in Artiticial Intelligence: Proceedings of the Sixth Conference*, 1990.

Laskey, K.B. and Mahoney, S.M. Network Fragments: Prepresenting Knowledge for Constructing Probabilistic Models. *Uncertainty in Artificial Intelligence: Proceedings of the Thirteenth Conference*, San Mateo, CA: Morgan Kaufmann, 1997.

Laskey, K.B. *Network Fragments: Prepresenting Knowledge for Constructing Probabilistic Models*. Rosslyn, VA: IET, Inc., 1996.

Laskey, K.B. *Knowledge Elicitation and Modeling for Spatial/Terrain/Location Reasoning*. Memorandum to IET, Inc., Fairfax, VA: George Mason University. September, 1995.

Mahoney, S.M. *Situation Specific Networks*. Rosslyn, VA: IET, Inc., 1998.

Mahoney, S.M. and Laskey, K.B. Network Engineering for Complex Belief Networks. *Uncertainty in Artificial Intelligence: Proceedings of the Twelfth Conference*, 1996.

Wellman, M.P., Breese, J.S. and Goldman, R.P. (1992) From Knowledge Bases to Decision Models, *Knowledge Engineering Review*, 7(1), 35-53.

Regan, P. and Holzman, S., R&D Analyst: An Interactive Approach to Normative Decision System Model Construction. *Uncertainty in Artificial Intelligence: Proceedings of the Eighth Conference*, 1992.