A Bayesian Methodology for Effects Based Planning

'Bert Bullen
Advanced Information Processing Department – BAE SYSTEMS Advanced Technology Centre
Bldg 20R, PO Box 5 FPC 267
Filton, BRISTOL BS34 7QW
UK

bert.bullen@baesystems.com

ABSTRACT

We present a methodology that employs Bayesian Networks in aiding Effects Based Planning. The network models the probabilistic dependencies between elements of the domain of interest in which a mission is being planned, the actions taken on those elements and the effects resulting from those actions. The model can be interrogated from the perspective of the actions, the elements of the domain or, recursively, the effects.

1 INTRODUCTION

This paper illustrates the use of Bayesian Networks (BNs) in aiding Effects Based Planning (EBP). The benefit of using this methodology is demonstrated using the Bayesian EBP Reasoning Tool (BERT), an embryonic realization of this method. Sections 2 and 3 summarize the need for improved tools to assist in situation assessment and how the chosen example could benefit EBP, Section 4 further describes BNs in this context, including the defining of variables and states, the encoding of probabilistic dependencies and the elicitation of the expert knowledge required to construct the model including a simplified method thus reducing one of the main difficulties in building the network. Section 5 introduces the model and Section 6 describes BERT, a basic realisation of the technique. Finally Section 7 summarizes the process and describes potential developments in the adaptation of this technique.

2 SITUATION ASSESSMENT

In general, assessing the situation in the battlefield has become easier and more difficult by turns. On the one hand the mechanisms for obtaining data and information have improved with the enhanced sophistication of surveillance and reconnaissance techniques. The disadvantage of this is that the quantity of raw data has increased enormously; further, the less intelligent and judicious the collection of data, the greater this problem becomes. Hence, the ability to structure and store the data in some coherent form (a model) which can be inspected, actively interrogated and communicated between elements of a command and control structure could greatly assist in both the assessment of the situation and the making of any resultant decisions. We believe that Bayesian Networks, which encode simple probabilistic relationships between domain elements, are a straightforward, transparent (to the user) and intuitive method of capturing the knowledge which makes sense of the data and therefore can provide useful information to decision-makers in the field of operation. The subject of Effects Based Planning is a particular instantiation of this process.

Bullen, 'B. (2006) A Bayesian Methodology for Effects Based Planning. In *Information Fusion for Command Support* (pp. 5-1 – 5-8). Meeting Proceedings RTO-MP-IST-055, Paper 5. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int/abstracts.asp.

3 EFFECTS BASED PLANNING

Effects Based Planning (EBP) is currently finding favour with some military planners at some or all levels of mission planning, i.e., in the UK parlance, strategic, operational and tactical levels. The idea that EBP is purely 'planning to achieve an effect' is naive in the extreme and most military planners would argue that previous approaches, including the currently used manoeuvrist approach are driven by the attainment of mission goals through achieving desirable effects to the friendly forces and undesirable ones for the enemy. What is 'new' under the banner of EBP is that non-kinetic effects are given far more consideration and that an holistic view of a mission is taken involving not just military but other involved or affected agencies. The recursive process of *'planning backwards from victory'*[1] is, in the current warfighting climate, made immensely complicated by: the number of actors who are considered to have an effect on, or who are affected by, the mission; the amount of information potentially available to the planners; and the labyrinthine network of possible 1st, 2nd order and tertiary effects which may be considered in the planning process. All of this provides a ripe environment to potentially show advantage of data and information fusion techniques to aid in the planning procedure.

4 BAYESIAN NETWORKS

Bayesian Networks (BNs) are a means of modelling the probabilistic dependencies between variables within a domain. They are used (for instance) in the aggregation of data and information in order to achieve situational awareness [2], for modelling the cause and effect in systems for medical/fault diagnosis and in other areas. The resulting model allows dynamic interaction of the user such that they can instantiate variables whose states are known or hypothesised and the (possibly unknown) variables of interest will react according to the probabilistic dependencies encoded in the model.

4.1 Variables and states

Two of the main constitutes of a Bayesian Network graph are variables and states. Careful preparation is required in order to correctly capture the elements of a domain and the states in which they exist at any stage in the mission.

Variables[1] in a BN of this type fall into three categories

- **Nodes** are entities in the domain of interest; that is the elements that constitute the environment in which the mission or campaign is taking place. They can be physical entities such as components in a country's infrastructure or geographical features, people such as politicians, military leaders or other key players or conceptual entities such as tribes, government departments or religions.
- **Actions** are taken on the nodes which result in their state being altered
- **Effects** result from the changing state of the nodes that in turn result from actions being executed.

Variables are (initially) given very simple (usually binary) states: Nodes could be *effective* or *ineffective* indicated their ability to perform their normal function. Actions could be *fully taken* or *not taken*. Effects could be *fully achieved* or *not achieved*.

4.2 Probabilistic dependencies

The key to the correct operation of a BN is the way in which probabilistic relationships are encoded. In this situation one expects that an action being taken affects the state of a node which, in turn, results in a effect being achieved to a greater or lesser extent. The fact that these relationships are encoded as

[1] Variables are often referred to as the 'nodes' in a BN. As 'nodes' have a very specific meaning in this paper, the term 'variables will be exclusively used in this paper.

conditional probabilities allows for some variability and uncertainty in the result that can allow the user to make decisions based on his judgement superimposed on the output of the model. All probabilities in this paper exist in the interval [0 ..1].

4.3 Knowledge Elicitation

We have stated the BN model comprises variables, states and probabilistic dependencies. To create a model which accurately affects the domain relies on correctly identifying the variables and states and accurately assigning the probabilistic weightings. In a military planning environment, this has to be done by eliciting expert knowledge[2] that can be an extremely difficult and time-consuming process. It is imperative, therefore, that complexity of the model be constrained to a simple representation of the key features of the domain which remains a transparent encapsulation of the knowledge and experience of the expert.

The most difficult aspect in creating a BN is assigning the values that populate the Conditional Probability Tables (CPTs); that is, the effect on the probability distribution over states of a variable given the combinations of states of those that have influence on it (its 'parents'). For the purposes of this methodology, given binary states of each variable, a simple weighting system is employed on the links between variables.

Figure 1 illustrates a simple example. The CPT for the node has to contain the probabilities of it being *effective* (P(**Node**=*effective*)) and *ineffective* (which is 1-P(**node**=*effective*)) for all combinations of the states of the **Actions** which have influence upon its state.

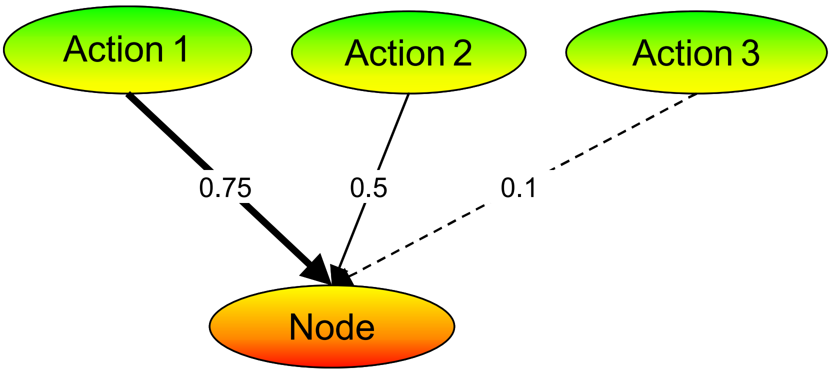

Figure 1: Simplified Influence Weighting

These weightings result in the following CPT:

[2] BNs can be learned from data but this is not appropriate in this context as data is extremely sparse

	Action 1	Action 2	Action 3	P(**Node**=*effective*)
1	*Not taken*	*Not taken*	*Not taken*	0.9
2	*Not taken*	*Not taken*	*Fully taken*	0.8
3	*Not taken*	*Fully taken*	*Not taken*	0.4
4	*Not taken*	*Fully taken*	*Fully taken*	0.3
5	*Fully taken*	*Not taken*	*Not taken*	0.15
6	*Fully taken*	*Not taken*	*Fully taken*	0.1
7	*Fully taken*	*Fully taken*	*Not taken*	0.1
8	*Fully taken*	*Fully taken*	*Fully taken*	0.1

Table 1: Example Conditional Probability Table

In this example all actions have the effect of reducing the probability that the node is effective and, from expert knowledge, the table could be filled in longhand as it is shown and, with a three parent binary example this is perfectly feasible. However the length of the table scales as the product of the parent states and can quickly grow to unmanageable size; for instance with all binary nodes tables will have 2^P (where P is the number of parents) entries to complete. Entries in CPTs have to be correct in two ways; the probabilities have to be of the correct magnitude for the state of the parents and they have to be in the correct relative order as all the combinations of the parents are considered. Of course the one follows from the other but it is easy, when completing a long table, to lose the anchoring which preserves the relative ordering.

This example, however, is contrived to illustrate a simpler method of capturing the probabilities that ensures the correct relative ordering of the probabilities assuming some level of individual influence of the parents. Initially an upper and lower bound is set (in this case 0.9 and 0.1 respectively; these rows are highlighted in black) and then the weights for each individual relationship are determined. For instance the change in P(**Node**=*effective*) between lines 1 and 2 of the table is the result of 'flipping' the value of **Action 3** from *Not taken* to *Fully taken* and has the effect of reducing P(**Node**=*effective*) by 0.1. This, then, is the weight of influence of **Action 3** on the **Node** as illustrated on the link between them in Figure 1. Similarly the change between lines 1 and 3 results from flipping **Action 2** (resulting in a 0.5 drop in P(**Node**=*effective*)) and between lines 1 and 5 a 0.75 difference resulting from taking **Action 1**. This operation produces a set of 'waypoints' in the relative ordering of the conditional probabilities; these rows are highlighted in grey. The remaining values are just a matter of interpolation; in this case the weights are simply summed, up to the lower threshold of P(**Node**=*effective*)) but parameters can be added to account for the reduced cumulative effect of taking more than one action simultaneously, which may range anywhere between the effect of taking the 'strongest' action only (i.e. secondary actions have no additional influence) and the full additive effect of taking all actions. It is important to note that this shortcut will ensure the correct relative ordering of the probability list but will not guarantee that their absolute magnitude is correct, especially in the case of the cumulative effect of taking more than one action. Expanding the number of states of a parent merely means that the weights of influence of the interim states are interpolations between its most benign and active states but this adds another layer of complexity to the operation. This is a similar construction to the 'Noisy OR' method of assigning conditional probabilities [3]. It has some resonance with the weights in a neural network but the model retains the transparency that is one of the strengths of a BN from a non-expert perspective.

In terms of knowledge elicitation, then, the expert does not need to concern himself with hideous CPTs but can merely assign weights to the probabilistic dependencies (these can be as simple as 'weak' 'medium' and 'strong'); a simple script with a few parameters trivially converts this to the relevant CPT. Weights lie in the same interval as the probabilities being used and do not need to sum to one (or the maximum assigned probability).

5 THE MODEL

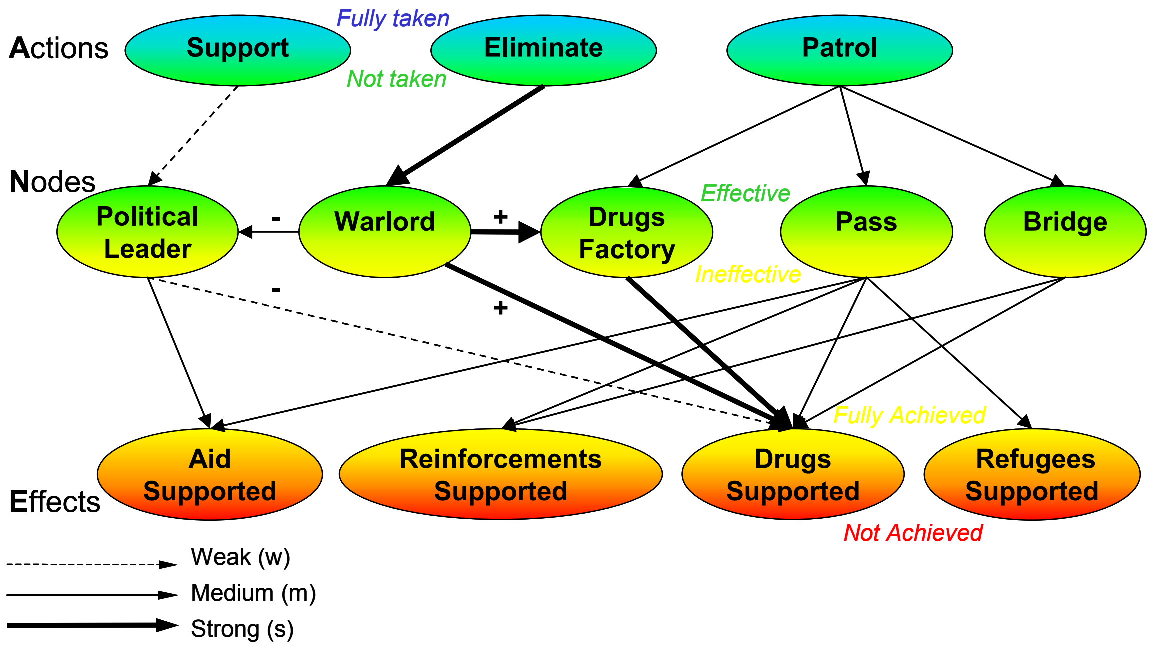

Figure 2: Simple Domain Model

The model (as illustrated in Figure 2) has a simple 3-layer topology. Once the model is set it can be interrogated in various ways: In a top-down sense, one or more actions can be instantiated and the change in the effectiveness of nodes and the level of achievement of effects could be observed. This is probably not very useful as this requires the user to anticipate the effect of interesting actions, which implies that they have worked out the effect *a priori*. Of perhaps more use is instantiating the states of nodes to investigate a) the resultant effects and b) perhaps the most suitable action. This is the equivalent of asking: "What if we disabled this political leader?" Of most interest from an EBP viewpoint is to instantiate the level of an effect and see a) what nodes have to be affected and b) what is the most appropriate action. Furthermore, second order effects resulting from the change in state of common nodes could be observed and this could lead to constraining the space of actions. For instance the user could instantiate **Drugs Supported**=*Not achieved* and **Aid Supported**=*Fully achieved*. This is equivalent to asking: "How do I reduce drugs but not affect aid and from the model, it can be seen that there are many nodes that can be affected resulting in the former while those which will adversely affect the latter can be forced to be discounted.

6 BERT

This methodology has been realized in an embryonic tool for the purposes of demonstration. The Bayesian EBP Reasoning Tool provides a simple GUI with underlying model that demonstrates the concept. The three elements of the GUI are described below. In the following sections, only the **Node** and **Effect** layers are considered and the example is that given in Figure 2.

6.1 Set-up Window

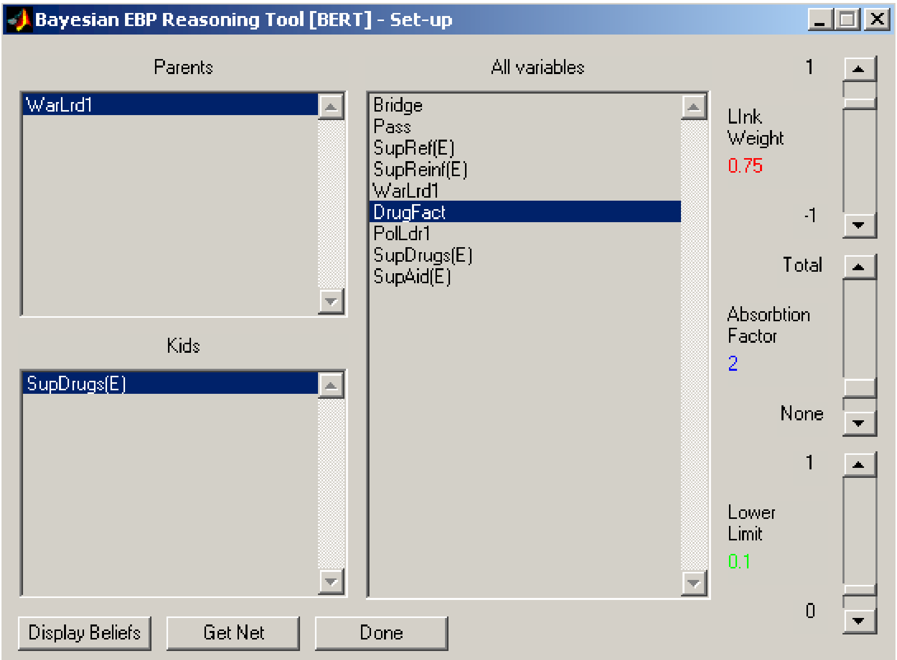

Figure 3: BERT Set-Up Window

For an existing BN, that is one whose structure has been defined, the set-up window will allow interrogation of the network in a dialogue sense and will enable the parameters, as described above to be adjusted. In Figure 3 the variables in the *All Variables* window can be individually highlighted; the highlighted variable is the 'subject variable' for the purposes of all the other windows. The variables that influence the subject variable and those that they influence are shown in the *Parents* and *Kids* windows respectively. In addition three parameters can be set/adjusted in the slider bars on the right: The upper slider sets the link weight between the highlighted variable and its (highlighted) child (as described above). The centre slider sets an *Absorption Factor* which moderates the effect of cumulative actions. The bottom slider sets a *Lower Limit* below which the probability of the subject variable will not fall.

6.2 State and Evidence Windows

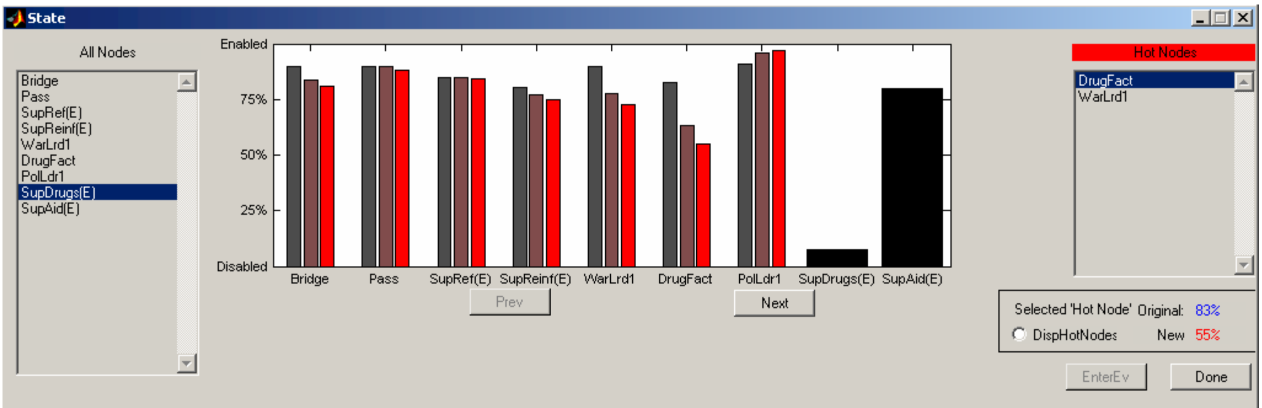

Figure 4: BERT State Window

The state window illustrates the state (probability) of variables within the model. The variables are listed in a listbox on the left, with the subject variable highlighted as in the set-up window. The bar display contains a 3-bar display for each variable; the three bars are, in order, the original state of the variable, the state immediately before the last change in the model and the current state. This allows a 1½D display indicating the trend in the state probability values. To enter evidence on (instantiate) a highlighted variable, the slider bar in the evidence window is used; this changes the bar display in the state window to black. Highlighting a different variable in either the set-up or state window allows evidence to be entered on that variable.

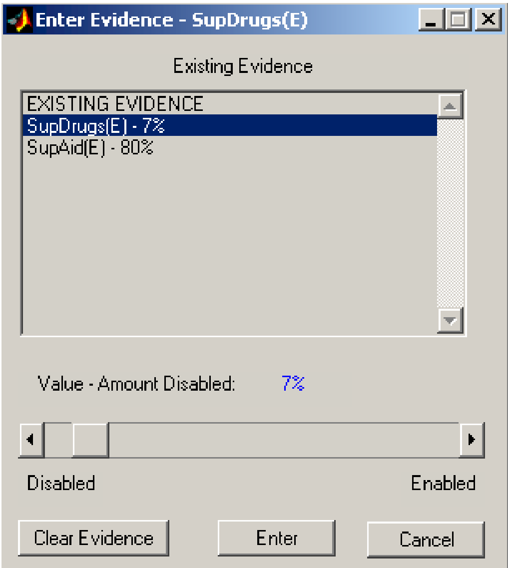

Figure 5: BERT Evidence Window

In the example illustrated, from the model in Figure 2, **SupDrugs** has been progressively reduced to 7% and **SupAid** held at 80%. This is the equivalent of making the enquiry of the model: "I want to reduce drugs but not affect aid". This results in a large change in **WarLrd 1** and **DrugFact** as can be seen by their bar graphs indicating that these are the preferred elements in the domain to act on, rather than, for instance the pass and the bridge that would have adversely affected **SupAid**.

7 SUMMARY AND FURTHER WORK

We present a Bayesian methodology for modelling the dependencies within domain, or part thereof, for the purposes of Effects Based Planning, along with a simplified knowledge acquisition method for building the model from expert knowledge. The methodology is built around an effects-nodes-actions architecture and is simply illustrated by the use of the experimental tool BERT.

However, this demonstration only concerns the methodology in its most basic form and there are several areas which provide for fruitful research:

7.1 State Representation

The state representation, particularly of the nodes, in this demonstration is oversimplified and does not contain the richness to correctly describe the state of the domain at any time-step. For instance, the state of the pass in this toy model is *Effective* or *Ineffective* but this does not sufficiently capture the true state of the pass in the context in which an enquiry is made. If a pass is patrolled by friendly forces, this is intended to make it *Ineffective* for the purposes of drug trafficking and enemy reinforcement and *Effective* for the purposes of aid and friendly reinforcement. Some further though needs to be given to the appropriate state representation.

7.2 The Dimensions of War

However the 'Dimensions of War' are presently described, it cannot be denied that enquiries to a model if this type are made in the context of different levels within these dimensions. For example, the dependencies between nodes at the political and military and economic etc levels may be fundamentally different. This adds yet another layer of complexity to the model building and the conflict of constraining the model to transparently represent the domain but capture enough of its complexity to be an accurate representation needs to be addressed.

7.3 A Dynamic Model

It is extremely naive to pre-suppose that the domain elements exist in isolation at some point in time, independent of their state and the state of associated elements at previous time points. Any model needs to be constructed in such a fashion that these cross-time dependencies are accounted for. Dynamic Bayesian Networks allow for this dependency structure but quickly become complex and computationally intensive to solve. Research is needed in this area to investigate the incorporation of this dimension into the model such that its simplicity and transparency is preserved however the advantages of being able to capture momentum and inertia in the behaviour of the model are undeniable.

7.4 Loops

Bayesian Networks cannot handle loops (or two-way probabilistic dependencies). A directed cyclic set of dependencies merely results in rapidly escalating recurrent probabilities. As an instance a tribal leader is dependent on his tribe for support but has influence over them while in power. Addressing this two-way dependency in more than the most naive way (i.e. assuming some net single dependency) requires careful thought and perhaps some new representation. The dynamic model, referred to in Para 7.3 would allow for cycling dependencies through time where a node influences its own state at some future point but again, further research is needed in this area.

8 REFERENCES

[1] D. Macgregor, Colonel (US, ret'd). 2003. Transformation Under Fire, Praeger

[2] P. Bladon *et al*. 2005. High Level Fusion using Bayesian Networks: Applications in Command and Control, in proceedings NATO IST-055

[3] J. Pearl. 1986. Fusion, propagation, and structuring in belief networks. Artificial Intelligence, 29(3):24–288.