DESIGN OF A SOFTWARE ENVIRONMENT FOR TACTICAL SITUATION DEVELOPMENT

M. J. Coombs and R. T. Hartley

MCCS-88-144

Computing Research Laboratory
Box 30001
New Mexico State University
Las Cruces, New Mexico 88003

ABSTRACT

This paper concerns the development of a prototype Tactical Situation Development Environment (TSDE) which will aid intelligence analysts to construct and evaluate future situational projections and will support this function by maintaining appropriate models of the current situation. Conventional automated problem solvers have failed in these tasks because they are deterministic, being designed to solve pre-determined problems using pre-defined sets of knowledge and data. They are unable to accommodate the uncertainty and dynamic events characteristic of the realistic battlefield. In contrast, we have employed a general-purpose automated problem solving architecture - Model Generative Reasoning (MGR) - designed for information processing in noisy and ill-specified problem domains. The MGR architecture is currently being developed for a number of military information integration applications, including meteorological data fusion, situation analysis and deception planning.

The Computing Research Laboratory is principally funded by the
New Mexico State Legislature,
administered by its Science and Technology Commission
as part of the Rio Grande Research Corridor.

DESIGN OF A SOFTWARE ENVIRONMENT FOR TACTICAL SITUATION DEVELOPMENT

M. J. Coombs and R. T. Hartley
New Mexico State University
Las Cruces, NM 88003-0001, USA

J. R. Thompson
Science Applications International Corporation
Albuquerque, NM 87106, USA

ABSTRACT

This paper concerns the development of a prototype Tactical Situation Development Environment (TSDE) which will aid intelligence analysts to construct and evaluate future situational projections and will support this function by maintaining appropriate models of the current situation. Conventional automated problem solvers have failed in these tasks because they are deterministic, being designed to solve predetermined problems using pre-defined sets of knowledge and data. They are unable to accommodate the uncertainty and dynamic events characteristic of the realistic battlefield. In contrast, we have employed a general-purpose automated problem solving architecture - Model Generative Reasoning (MGR) - designed for information processing in noisy and ill-specified problem domains. The MGR architecture is currently being developed for a number of military information integration applications, including meteorological data fusion, situation analysis and deception planning.

1. THE SITUATION ANALYSIS PROBLEM

A major part of situation analysis is a data driven process that depends on sensor data for unit and node location and identification. The continuous monitoring of enemy order of battle makes this aspect fairly reliable, being mostly a matter of correlating signatures to equipment and equipment to units through tables of organization and equipment (TO&E). However, analysts tend to become overdependent on sensor data, which commonly leads to operational inflexibility and the susceptibility of intelligence to enemy deception.

Intelligence products with the highest payoff are those that go beyond raw data to predict enemy intentions (Thompson et al., 1988). Models of intentions are valuable because they enable operations to anticipate enemy operations, identify enemy vulnerabilities, and improve performance through added preparation time. They are also difficult to construct. The analyst must integrate into a coherent set of interpretations a complex of diverse, and often dynamic, factors. Moreover, integration has historically been manual, based on sets of map overlays, and must frequently proceed on the

strength of very uncertain data, or even the absence of data expected given some observation. Situation analysts thus have a natural tendency to report uninterpreted facts, to hedge bets by qualifying predictions, or to delay predictions until more evidence is available.

Support for intelligence preparation of the battlefield (IPB) is currently provided in the form of templates. These provide the analyst with prototypical background information on all aspects of the Red and Blue force, including: the organization, equipment and operational postures of units derived from records of enemy doctrine; force capabilities and effectiveness based on studies of tactical and operations norms; time/distance factors in force maneuvers based on training exercises; observables for force elements based on knowledge of sensor system technical characteristics and the signatures of units. Templates represent descriptions of prototypical battlefield elements and are intended to be used as starting points for data interpretation. To them the analyst must add an understanding of the battlefield area in terms of terrain and the effects of weather on friendly and enemy troops, plus any cultural aspects of the area that could impact on operations. Finally, he must integrate battlefield data into this templated picture to form a description of the battlefield situation.

The templating process constitutes a complex hypothesis formulation and testing task (Thompson et al., 1983). Situational hypotheses are formed from multiple template overlays which are adjusted to meet environmental constraints. These compound templates are then tested through the integration with battlefield data. However, there are strong indications that the templating procedure is not adequate for capturing the complex, highly dynamic and variable events characteristic of a battlefield and so may actually add to the analyst's cognitive load without improving performance (Thompson et al., 1986; Coombs et al., 1988b). In particular, it has been found that: (i) parts of a template my conflict with real intelligence data in ways that require the analyst to engage in a lengthy process of resolution, where the conflict is an artifact of the template's status as a prototype (c.f., Brachman, 1985); (ii) the method does not in itself provide support for resolving incoherence between several different templates selected as interpretations of different subsets of fragmentary data, thus adding to, rather than reducing, the analyst's uncertainty; (iii) an analyst may need to characterize a situation which is novel and which has thus not been templated; (iv) templates do not adequately capture the temporal aspects of situation development.

The templating procedure does not in itself reduce either of the main groups of cognitive "pathologies" to which arise as analysts cope with the above weaknesses in the templating method. These include: (i) assimilation pathologies, in which data are over-generalized or over-interpreted in order to fit existing hypotheses, leading to a bias towards confirming expectations, and (ii) accommodation pathologies, in which implausible or over-elaborate hypotheses are manufactured in order to accommodate incoherent subsets of data, leading to a bias towards switching hypotheses upon receipt of minimally conflicting information. These types of biases are observed in a wide range judgment and decision tasks (see Pitz and Sachs, 1984). However, the negative effects may be reduced if the analyst is able to keep multiple situational hypotheses in play during situation development (Thompson et al., 1983). Over-assimilation will then be avoided by retaining alternative interpretive structures for data and overaccommodation will be avoided by forcing the justification of each interpretation in

terms of alternatives.

It is not possible to formulate, test and maintain multiple situational hypotheses of any complexity without automated support. This is particularly true for the analyst working in the field under stress and facing a barrage of changing battlefield data. An automated aid should help the user to: (i) construct and evaluate alternative situation scenarios as competing interpretations of available data; (ii) maintain a set of coherent scenarios in response to changing data; (iii) identify sources of incoherence between the hypotheses and data as they arise; (iv) enable the user to test the effects of tactical assumptions on the set of scenarios in order to develop predictions (i.e., "what if?" games). Moreover, because the automated aid is required to radically enhance analytical skills by changing the user strategies for information integrating, the technology must be capable of being embedded into training systems. Any support system available during training must also be capable of serving as an aid to real situation analysis in the field. The Model Generative Reasoning (MGR) automated problem solving architecture (Coombs and Hartley, 1987, 1988; Fields et al., 1988) developed in the Computing Research Laboratory (CRL) will provide such a technology.

2. TECHNOLOGICAL REQUIREMENTS

The inputs to the situation analysis process are varied in form and can have a high degree of uncertainty. Inputs come from multiple sources both within and from outside the all source production team that integrate intelligence information and produce outputs for users. Inputs are often: (i) incomplete (only a subset of predicted observables are received); (ii) incoherent (items of information relate to situations and plans that are hypothetical alternatives); (iii) overly generalized (information is not sufficiently detailed to permit inferences at the required level of precision); (iv) novel (items can only be explained by some unanticipated event); (v) misleading (the clearest interpretation of information may be incorrect, perhaps due to deception or sensor error); (vi) perishable (information may, or may not, become unreliable over time); (vii) irrelevant (information may, or may not, be relevant to the situation).

Given these multiple sources of uncertainty, intelligence data are usually open to many different interpretations. However, analysts usually work under significant time pressure and suffer from information overload. It is therefore difficult for them to generate multiple interpretations and determine coherence of inputs to these interpretations in order to select the best situational models. As the input problem becomes more severe, the overall picture may be lost as the analyst's time is consumed in handling detail, or the analyst is drawn into the enemy's deception plan because the deceptive inputs are both clear and coherent.

Proposals for providing computer support using AI, including automation of the templating method discussed in the previous section, have proved unable to meet the challenge. In common with other Al-based tactical decision aids TDAs), automated situation analysis aids employ knowledge-based system technology. This technology was developed by AI for application in complex domains where problems are not amenable to algorithmic solution because of the uncertainty or incompleteness of problem solving knowledge or data. The problem solver must therefore rely on the weaker

strategy of edging towards a solution by the application of successive heuristics in response to descriptions of individual problem states (Hayes-Roth et al., 1983). The justification for this approach is clear: a battlefield represents a complex and highly uncertain environment. It is therefore not possible to specify algorithmic solutions, even when problems can be clearly specified in advance. However, the technology has proved to be brittle, i.e., subject to unanticipated failure, in the very environments it was designed to address (Coombs and Alty, 1984).

A knowledge-based program constitutes an executable specification of the problem domain. The developer must first define the scope of the specification in terms of a set of battlefield situations and the set of allowable data input. Then, in the context of a control regime, a set of heuristics are added in order to relate data to situations. The control structure provides the procedural framework within which inferences are generated through the selection of heuristics, and within which arbitration takes place in the event of conflicting selections. Together, the control structure and heuristics define a set of relevance relations between data and the domain knowledge represented in the heuristics.

The knowledge-based approach to TDA design is only effective in domains where relevance relations between knowledge and data can reliably be defined in advance. This is not possible in battlefield situations because data are subject to multiple sources of noise (see above) and the situations to be analyzed may be novel. In such domains, the issues of filtering noise from input data and of developing hypotheses to explain data must be regarded as two aspects of the same problem and must be solved together. It should be noted that in conventional knowledge-based programs, uncertainty is handled by determining the reliability, in the statistical sense, of available solutions as indications of the state of the world. This contrasts with the problem of determining the relevance of available knowledge and data to the problem currently being addressed.

MGR was designed to provide an architecture for automated problem solving in uncertain task environments. The task environment is defined as *uncertain* when it is not possible to anticipate the range or type of data that the system will have available ahead of time. It will therefore not be possible to identify the set of problems that the system will be able to solve, and hence not possible to specify the knowledge the system will need for solution (Fields and Dietrich, 1988). The objective of an MGR system is to form descriptive structures - models -*that* provide coherent interpretations *covers* - of particular world situations. Models are generated, evaluated and refined continuously in response to new input from the system's environment. MGR problem solvers are therefore not goal driven in the traditional sense. They have no fixed expectations that determine the structure of solutions, although deviation from expectations may be a factor in the evaluation of solutions for the purpose of answering queries.

MGR provides a powerful technology for determining relationships between uncertain data and templates through its ability to decompose, reconstruct and refine alternative situation descriptions, represented as MGR models, in order to achieve coherent covers of input. MGR is capable of: (i) automatically generating multiple alternative situation descriptions where information is ambiguous; (ii) maintaining the internal coherence of situation descriptions with new input; (iii) enabling a given information item to be viewed in multiple contexts; (iv) decomposing and recombining knowledge structures (e.g. doctrinal templates) in order to cope with novel conditions; (v)

projecting situations in time. Furthermore, the MGR architecture is intrinsically parallel. Systems developed using MGR are thus ideally suited for implementation on fast parallel hardware.

3. THE MGR ARCHITECTURE

The MGR architecture is shown in Fig. 1, and is described in Fields et al. (1988). MGR is logically a shared-data parallel virtual machine that accepts input from two databases, a fact database F containing input data and a definition database D containing stored knowledge. Models are generated by combining information from D with information from F. Four operators, *specialize (Sp), fragment* (Fr), *merge* (Mr), and *generalize* (Gn), act on the population M of models in an autonomous fashion. The functionality of these operators is specified completely by the architecture. The activity of the operators is governed at a control level above them; control determines when the operators act, but not their functionality. Strategy in MGR thus consists largely of scheduling these four operators. Additional operators *select (SI)* and *evaluate* (Ev) are employed to regulate the flow of facts from F and definitions from D to M, and to evaluate the resulting models with respect to user-specified criteria, respectively. The precise functionality of these operators, unlike that of the graph manipulation operators, is specified as part of each application program.

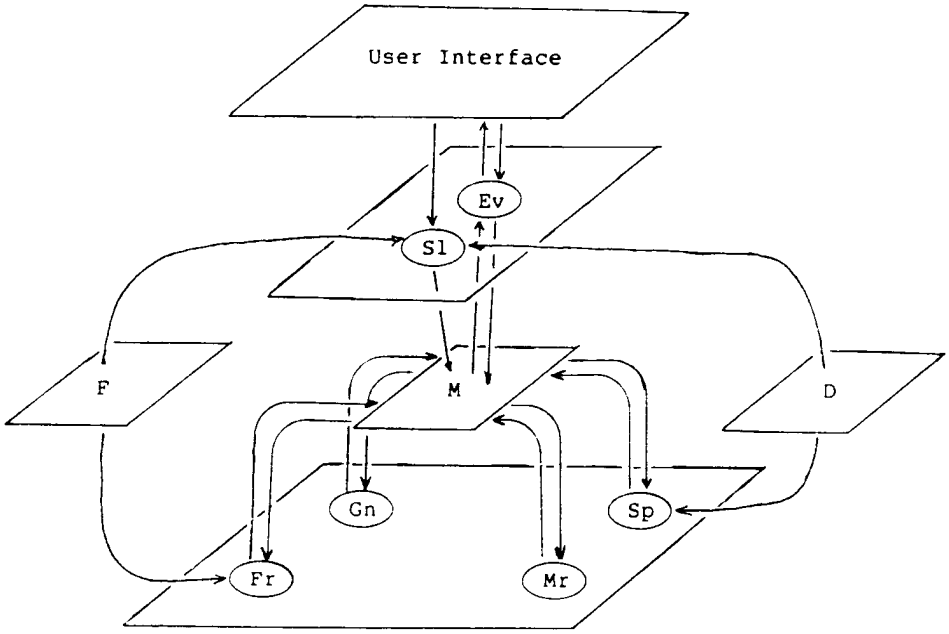

Figure 1. Data-flow diagram of the MGR architecture.

Informally, Sp takes a model m as input, and generates a set of larger, more specialized models by adding definitions. The role of Sp is, therefore, to "glue" knowledge to facts to create models that cover the facts. The number of alternative

models generated by each cycle of specialization may be changed by regulating the selection of the subset D' of definitions used as input to Sp. In the current implementation of Sp, D' is chosen as a minimal subset of D that allows a complete cover of m; Sp thus generates parsimonious models. Fr opposes Sp by breaking models into fragments in a way that preserves the information contained in facts, but may destroy some relational information obtained from definitions. Fr thus removes definitional information, but not factual information; its role is to break apart models that do not cohere with the available facts in order to generate fragments that can be recombined. Like Sp, Fr can be regulated through choice of the input fact set F'; F' is taken to be F in the current implementation. Gn and Mr take subsets of models as input, and generate single models as output which are, respectively, less or more specialized than the models from which they were generated. Gn is capable of generalizing both factual and definitional information; its role is to maintain coherence by removing overspecializations. Mr merges models whenever possible; its role is to generate models that have the greatest possible covering power. All of the operators write over the model population; the set of models available for operating on, therefore, changes after every operator application.

The operators Sp and Mr are based properly on maximal join (Sowa, 1984) and also are subject to the parsimony of minimal cover. The version of generalize used is a hybrid of Gn and Fr, which produces similar results to these, but without the analytic basis of Gn and Fr. Select, SI, is implemented as a user option - any number of facts can be selected for initial specialization. The evaluation function Ev is a simple query match that causes MGR to halt if the query is answered by any model. Queries are graphs; thus query answering is essentially a test for a model containing the query as a subgraph. The current system is implemented in CommonLisp and runs on Symbolics 3600 series Lisp Machines under Genera 7. 1.

4. THE TACTICAL SITUATION DEVELOPMENT ENVIRONMENT (TSDE)

4.1 THE TSDE CONCEPT

The concept of a Tactical Situation Development Environment (TSDE) for analysts engaged in situation development arose from research undertaken into intelligence information integration by (Thompson et al., 1983), in co-operation with USAICS, and by (Thompson et al., 1986), in co-operation with the RADC. As discussed in section 2 above, these studies concluded that the greatest single limitation is that human analysts are only able to consider one hypothesis at a time. The TSDE provides a basic set of data interpretation and situation modeling capabilities to be used as an operational aid to help analysts maintain and evaluate multiple hypotheses.

The TSDE architecture supports two contrasting analytical functions: (i) the maintenance of baseline situation descriptions in response to new intelligence information, and (ii) the projection of selected situation descriptions in order to evaluate hypotheses concerning possible Red intentions. In baseline processing the objective is to generate situation descriptions that are maximally faithful to available intelligence

data. In projection processing the objective is to generate descriptions that are maximally faithful to user assumptions.

The baseline situation development module interfaces to a database of intelligence reports. In the current experimental context, input to the database may either be controlled by the user or a battle simulation of the type that serves the G2 workstation. In the field, it would be continually updated as new information becomes available from individual intelligence disciplines. Data related to a specific mission task are selected from the report database and submitted to MGR for processing as sets of MGR facts F. The data are then interpreted using environmental information and deployment/activity templates, represented as the MGR definitions D, to form a set of alternative descriptions of the current situation, represented as the set of MGR models M. The alternative descriptions are then evaluated as explanations for the data in F. Further development of models is driven either by weaknesses discovered during evaluation, the arrival of new reports, or by user queries.

The projection module enables a user to construct temporal extensions of baseline models in order ask hypothetical questions concerning the development of events and their military impact in terms of threat, risk and uncertainty. Projection will usually focus on specific hypothetical ("what if?") queries which require the introduction of query specific assumptions into models, e.g., the examination of possible avenues of approach may require the addition of assumptions concerning Red's previous use of available avenues. It is unlikely that such information on Red's past actions would have been brought in by data during the construction of baseline descriptions and so it will have to be added in response to the query before predictions of possible Red actions can be generated. Other queries may require the selection of subsets of baseline models, e.g., those coherent with some query assumption, or the extraction of specific features, e.g., the removal of all environmental factors in order to give solutions a doctrinal perspective.

4.2 REPRESENTATION

Intelligence reports, deployment/activity templates and situation descriptions will be represented through the Conceptual Programming (CP) system (Hartley, 1986) as sets of conceptual graphs (Sowa, 1984). CP is being employed for knowledge representation in the MERCURY meteorological data fusion system (Coombs et al., 1988a) and has the capability of mixing the three levels of constraints illustrated in Fig. 2. In contrast to the static prototypical templates currently used for situation development, TSDE will form dynamic structures that integrate the material and functional aspects of military units with environmental factors. These structures are also capable of being animated and so initial states may be projected through time and space to generate a sequence of snapshots representing an evolving event. Any uncertainties in the development of events will become visible as underdetermined inputs to procedural and functional constraints. The example presented later demonstrates these capabilities.

4.3 TSDE PROCESSING MODULES

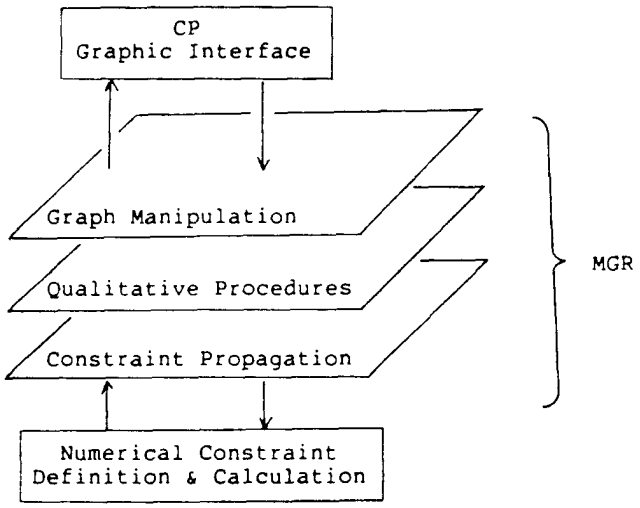

Figure 2. Representation layers in CP.

4.3.1 Baseline module

The objective of the baseline module is to maintain a set of models that give the most complete and detailed description of the current situation that is coherent with available data. Interpretations are generated to the extent of naming Avenues of Approach (AAs) and indicating Named Areas of Interest (NAls). The templates transferred to the set of working definitions **D** thus serve as a set of expectations which are used to augment and amplify relations between data items.

A basic control strategy has been adopted for the baseline module that ensures that baseline models are faithful to current data. This is achieved by generating specialized models which are complex and so are defeasible by new information. This will maximize the likelihood of a model being negated by new input. Following the basic diagnostic strategy reported in (Coombs and Hartley, 1987), this is implemented by giving the operators *specialize* (**Sp**) and *merge* (**Mr**) preference over *fragment (**Fr**)* and *generalize* (**Gn**). **Sp** will generate larger, more specialized models by adding definitions, while Mr will integrate them into maximally related structures. **Fr** is principally invoked when an incoherence or contradiction is found between a new data item and a previously supported model concept and has the effect of decomposing situation descriptions around facts into mutually incoherent sub-models. This will preserve portions of models supported by data and will seed the generation of one or more competing interpretations of the current situation that reflect conflicts within the data. **Gn** is applied when an incoherence is found between unsupported concepts during the application of **M**r. This has the effect of allowing the **Mr** to execute by generalizing away the blocking concepts. Since these concepts have no direct support from data, there is

no reason to maintain their level of precision. Individual baseline models will be evaluated for the degree to which they cover data, their complexity and precision, while the population of models will be monitored for various forms of noise (e.g., redundancy, over generality, perishability) and for the existence of evidence of incoherent subsets of data.

4.3.2 Projection module

Projection is undertaken on available baseline models in order to address specific intelligence queries. Queries typically focus on one of six questions concerning possible future enemy dispositions of actions. These include: who?, what?, where?, when?, with what?, and with what strength? Responses require the generation of hypothesized situation projections that elaborate alternative AAs and NAIs to give Target Areas of Interest (TAls). 'Me set of projections can then be evaluated with reference to the degree of enemy threat, the risk to friendly forces and levels of uncertainty. The projection module will seeks to project baseline models around opposing, although not necessarily incoherent, query assumptions. The projection strategy will seek to maximize the effects of these assumptions and the query will be answered over the resulting set of projected models.

The basic model generation strategy for the projection module varies from that of the baseline module in its emphasis on elaborating differences between models that impact on a query, irrespective of whether they have factual support. Following an initial run of *specialize* (Sp) to fold in user assumptions, *merge* (Mr) and *generalize* (Gr) will only be applied under user direction to support "what if?" experiments; Gn will be used to test the effects of loosening constraints imposed by specialized, but unsupported, sections of model while Gn will be used to test the coherence of model fragments. Currently Evaluation is under user control. The results of each "what-if" experiment are available for inspection.

5. THE BATTLE OF EDGEHILL

5.1 A HISTORICAL ACCOUNT

The battle of Edgehill was the first real encounter in the English Civil War between King Charles 1st's army and the opposing Parliamentarians. On 12th. October 1642, the King was in Shrewsbury ready to march on London (Fig. 3). The Earl of Essex, who led the Parliament's army, was stationed at Worcester. 'Me King started to march to London with some 26 regiments hoping that he could reach London before Essex, or at least encounter the enemy (actually the civil war was called 'this war without an enemy') and beat them in the field. Another factor in the decision was that the King favored a cavalry battle, and the land around Worcester was 'enclosed' i.e. the land was parceled up and divided by thick hedgerows - unsuitable for cavalry. October was almost too late for a serious campaign since the nights get too cold.

Progress was slow on the unmetalled roads, since the cavalry did not want to get separated from the Train of Artillery. After ten days, the King reached Banbury,

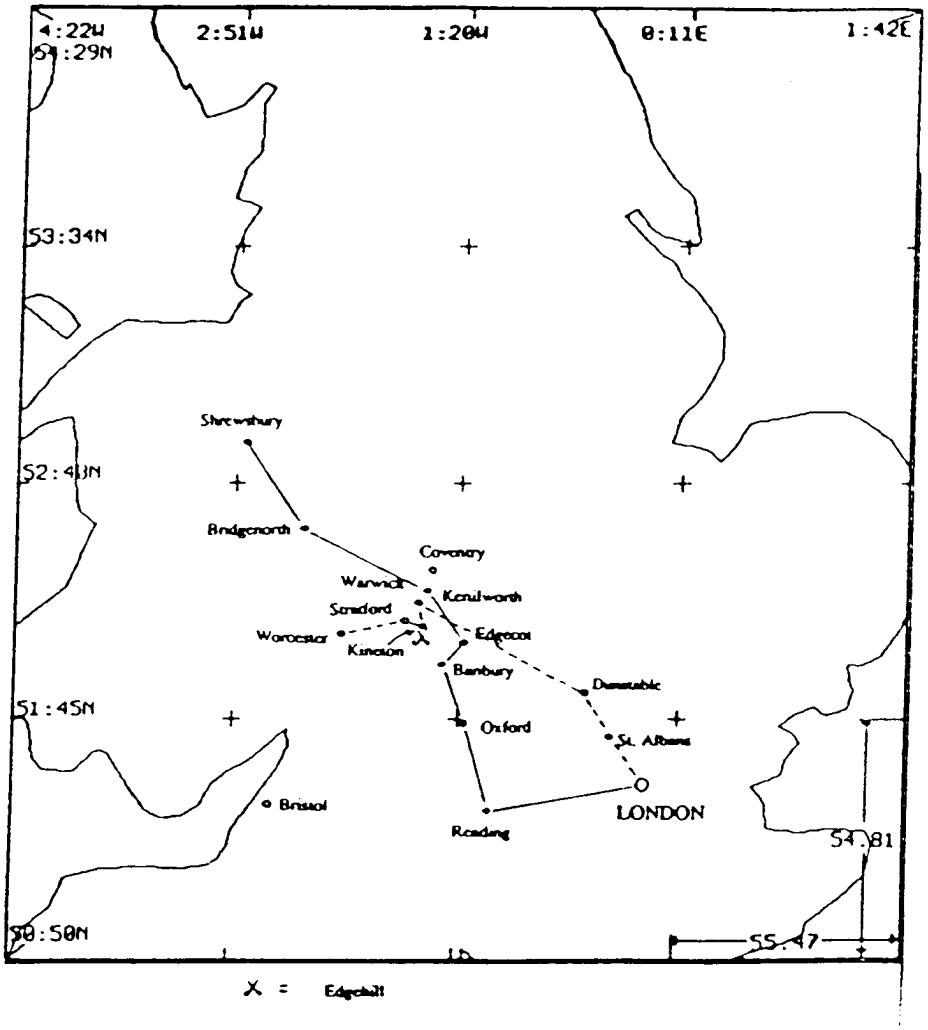

Figure 3. The theatre of war

threading between Parliament garrisons at Coventry and Warwick. Meanwhile Essex started from Worcester on the 19th. having got news of the King's march, and reached Kineton on the 22nd. On the same day, the King sent out Lord Digby to 'find out' the enemy. He returned with no news, but probably missed the arrival of Essex in Kineton.

The Royalists decided to rest at Edgecote and planned to send out a small brigade to take Banbury. The King scattered his army around the surrounding villages for the night, and it was in one such village that the King's quartermasters encountered those of the enemy, and although taken by surprise, they took the Parliamentarians prisoner. Having received this unexpected intelligence, the King sent out a force to confirm their presence in Kineton. The King resolved to rendezvous at Edgehill the following day, while Essex, acting on intelligence about the King's intent to march on Banbury, decided to try to relieve the town. The battle was then inevitable, since Edgehill lies on the path from Kineton to Banbury.

Edgehill was a steep escarpment devoid of trees, and the King assembled his troops on the top. Essex did not seem inclined to attack up the hill, so the King decided to go down and attack him. The King's cavalry, led by his nephew Prince Rupert, routed their opponents on the flanks, chasing them back to Kineton, but the foot soldiers fought to a draw, both sides being exhausted by the end of the day. Essex took his army off to Warwick, leaving the King a tactical victor.

5.2 OUR HYPOTHETICAL ACCOUNT

We have taken the Parliamentarian's point of view and have modified the historical account somewhat in order to display how the decision making might have gone with better intelligence gathering. Our intent is to show capabilities in providing automated help for the four objectives outlined in section 1. They are reproduced here for easy reference:

1. To construct and evaluate alternative scenarios as competing interpretations of available data.
2. maintain coherence in the scenarios in response to changing data.
3. identify sources of incoherence between hypotheses and data.
4. enable the user to test the effects of tactical assumptions to develop predictions

We will show that our example of the battle of Edgehill demonstrates capability in each of these areas.

5.2.1 Phase one: Baseline Models

The generation of the initial set of baseline models fulfills objective 1 above. We also show that by changing the input data only slightly, the nature of the models generated can change radically. This covers the second objective. We will now show, in detail, how these different scenarios are produced by MGR. Throughout the example, we take the Parliamentarian's point of view. Thus the 'Red' army are the Royalists, led by King Charles.

5.2.1.1 Available Intelligence

Initially there are three pieces of intelligence. They are:

INT1:the date - 9/20/1642

INT2:the location of the King's army - at Shrewsbury

INT3:a leading indicator - the fact that the King's army has been seen to be 'victualling' (gathering food and supplies) at a low level of activity. As we shall see, this

is evidence that the King intends to garrison Shrewsbury, possibly because of the lateness of the date in the campaign season, and the coming of winter.

We assume that intelligence has been gathered, filtered for immediate relevance (although MGR is capable of accepting any input) and converted to conceptual graph (CG) form. They are then input to the CP knowledge engineering environment, in the form of CP facts. These facts are shown in CG form in Fig. 4. They are clearly unconnected, in the sense that they have no labels in common. MGR's job, initially, is to connect them by finding a suitable minimal combination of definition graphs that covers all the labels in the fact set. To look at the possibilities of cover, we need to look at the available definitions.

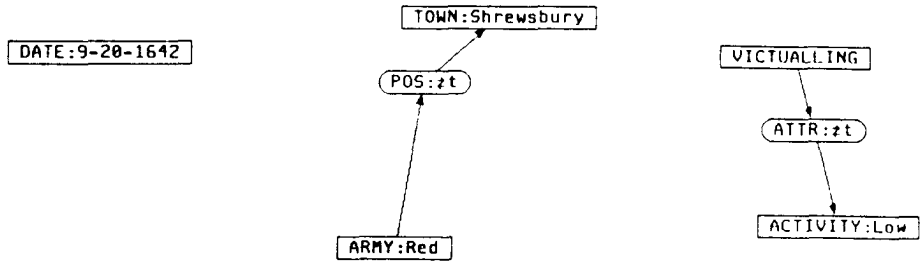

Figure 4. The initial pieces of intelligence

5.2.1.2 Knowledge of Red's Possible Tactics

All definitions are also input into CP and are then available to MGR. MGR makes connections between facts and definitions by matching the labels in the fact set with the contents of all definitions. Where a match is found, a partial cover of that fact with the corresponding definition exists. Total coverage is obtained when all labels in the fact set are covered by at least one definition. Since alternative combinations of definitions may be found to provide complete coverage, then a minimization is carried out, resulting in the minimum sets of definitions that cover the facts. If a particular label in a fact has only a single definition that covers it, then that definition is guaranteed to be necessary for complete cover. When a label has many alternative definitions that cover it, then, in general, this will produce many alternative total covers. The first is what happens initially in our example. The label VICTUALLING is only found in the definitions of the tactical options called GARRISON and PREPARE.

5.2.1.3 The GARRISON option

The graph for the definition of GARRISON is shown in Fig. 5. It contains the following information. An army garrisons in a town at the start of winter, and its leading indicator is the level of victualling activity. A constraint actor, called WINTER?, will fail a model containing it unless it is the start of winter, and the level of victualling activity is low. This is an example of the three layers of knowledge represented in CP. The first layer simply relates terms together, the second, the procedural overlay, relates any states and events together in a temporal fashion; the constraint overlay conditionally relates any instantiations of the term labels (numbers, dates, names etc.). This is a highly simplified account of what it was for an army to garrison, but it illustrates the principles of the representation of such knowledge. If there are multiple accounts of garrisoning that are possible, CP can also represent these for MGR to incorporate into its models.

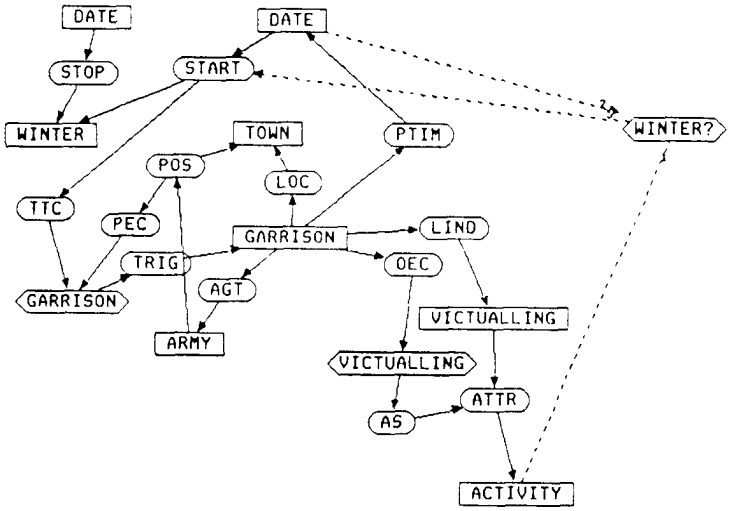

Figure 5. The definition of GARRISON

5.2.1.4 The PREPARE option

The alternative to garrisoning for an army is to prepare to march. Its definition (see Fig. 6) is very similar to garrison, but does not mention winter. A model containing it will only succeed if the level of activity is high. (The distinction between high and low could be made more elaborate, but here only a binary distinction is necessary).

5.2.1.5 The first models.

With the facts INT1-3, three models are produced, two with GARRISON, and one with PREPARE. Only the GARRISON model succeeds, however, because the

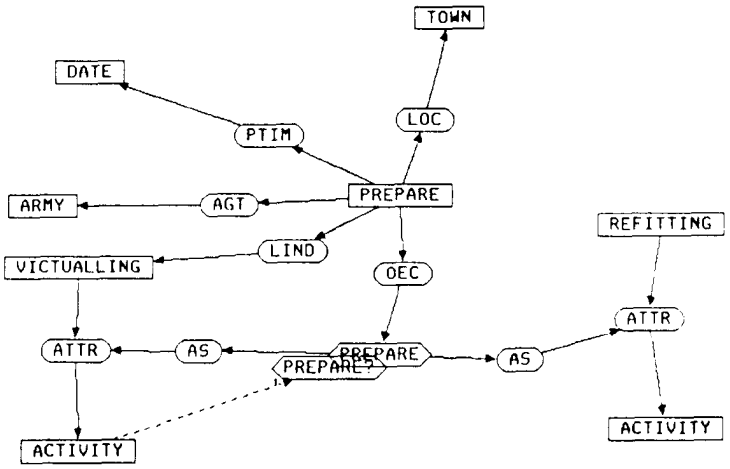

Figure 6. The definition of PREPARE

constraint actor in PREPARE fails, and in the other GARRISON model, the fact's date is identified with the end of winter instead of the start. The successful model is shown in Fig. 7.

At this point we assume that new intelligence reaches us, raising the level of victualling activity to high. This is INT4. The date has also progressed to October 13th. (INT5). By swapping INT4 for INT3, and INT5 for INT1, the the situation becomes inverted. The GARRISON models fail, leaving the PREPARE model (Fig. 8).

There is in both cases a single model that covers all the selected facts. Now assume that intelligence is gained of Red's new location (at Bridgenorth), and of the King's mission. This should give us a way of assembling models that explore the various possibilities of Red's objectives, and their consequences.

5.2.1.6 Red's Mission

From Fig. 9 it is clear that the King plans to take London before November 11 th., and that Rupert, the King's cavalry commander is a major player. Any models that take this into account will need to talk about battle tactics rather than the concerns of feeding the army that were incorporated into the first models. We have assumed three possibilities here. Firstly, there is the possibility of a BLITZ march on London, as fast as possible, presumably traveling without heavy artillery which would slow the march considerably. Secondly, Red could march to a friendly large town (Bristol was the obvious choice) where it could REINFORCE its ranks, then march on London. Lastly, Red could engage the enemy in the FIELD, beat them, and then march on London at its leisure. These three options are represented with temporal and constraint information in

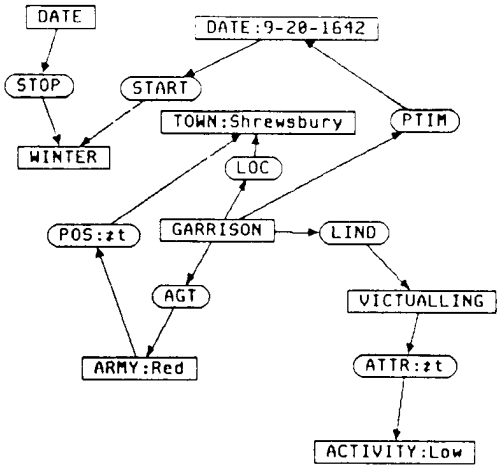

Figure 7. The GARRISON model

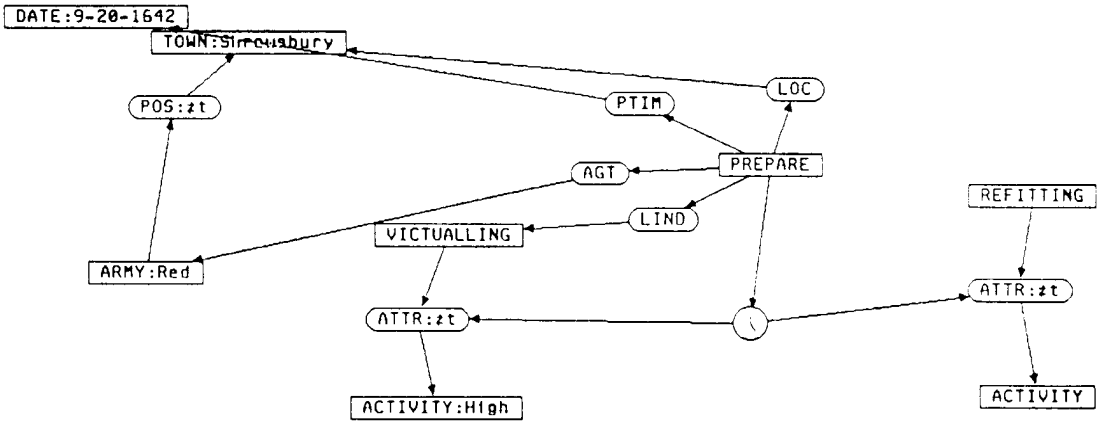

Figure 8. The PREPARE model.

as generic a fashion as possible. Indeed they may well be Blue's possible tactics as well, but we should assume that they are Blue's understanding of Red's preferences, and do not necessarily have any wider significance.

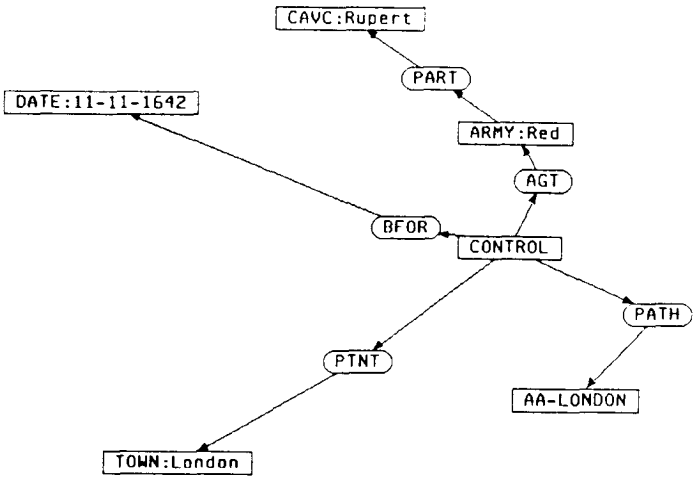

Figure 9. The King's mission.

With the new information, nine models are produced, three for each of the battle tactics combined with three possible avenues of approach (through Bristol, through Oxford, and through St. Albans). Of these nine, only two survive, the others being rejected on constraint satisfaction grounds. For instance, all three avenues of approach through Bristol are rejected because the march would take too long. The two surviving possibilities are to BLITZ through Oxford, and to engage in a FIELD battle. The latter survives because Rupert is a cavalry commander, and would prefer a FIELD battle in open terrain. These models are shown in Figs. 10 and 11.

Because there are two models which cover the latest intelligence, MGR can attempt to merge them in order to combine the previously separate options. In this case, the merge succeeds, resulting in a BLITZ/FIELD model which show how a battle could occur during a fast march on London through Oxford.

5.2.2 Phase two: Projection of Blue's Assumptions

In the second phase of situation analysis, the baseline models are assumed to have been evaluated and accepted as probably representative of Red's intentions. The next phase - projection -tests the coherence of a Blue assumption with any of the baseline models. In the example we have encoded two such assumptions (Fig. 12). Ile intent is to show how the named areas of interest in the baseline models (the towns on the various avenues of approach) become target areas of interest in the projection models.

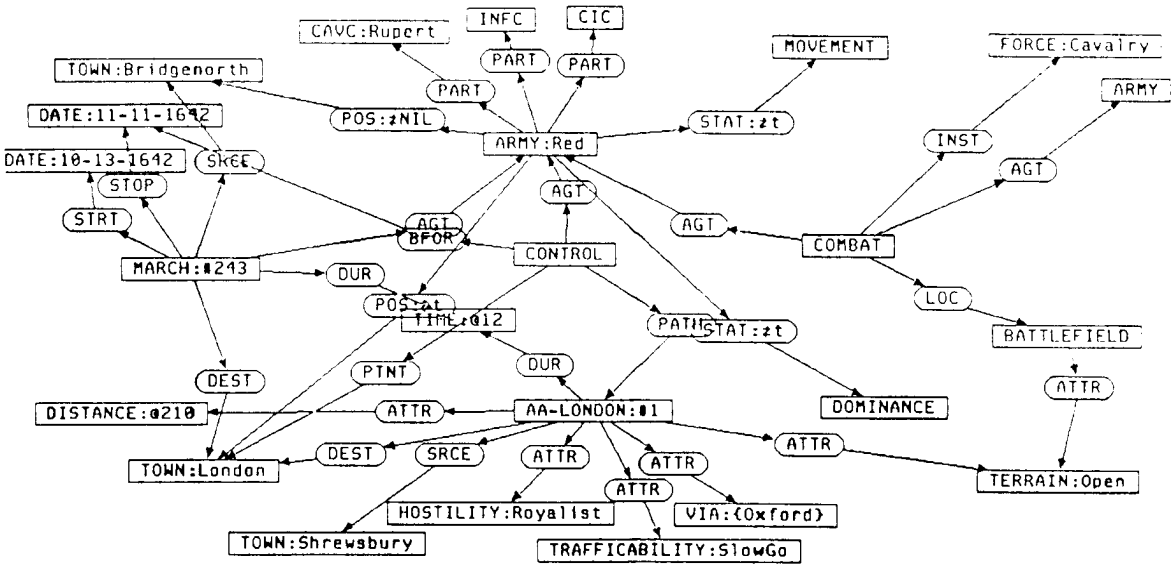

Figure 10. The FIELD model.

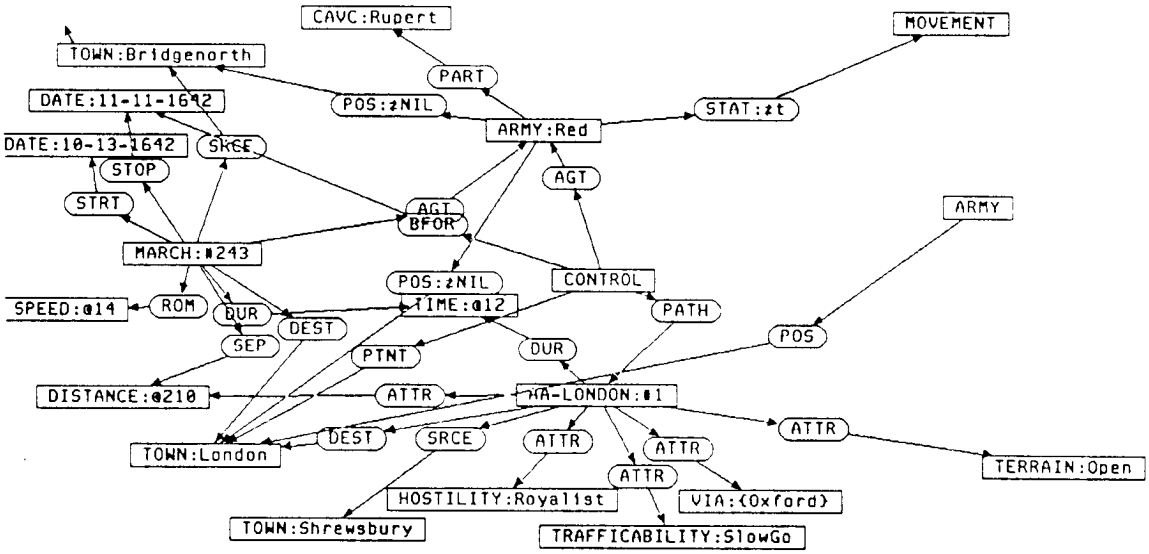

Figure 11. The BLITZ model

in particular, the second assumption leads to the projection of a battle site (Edgehill itself) as a target area.

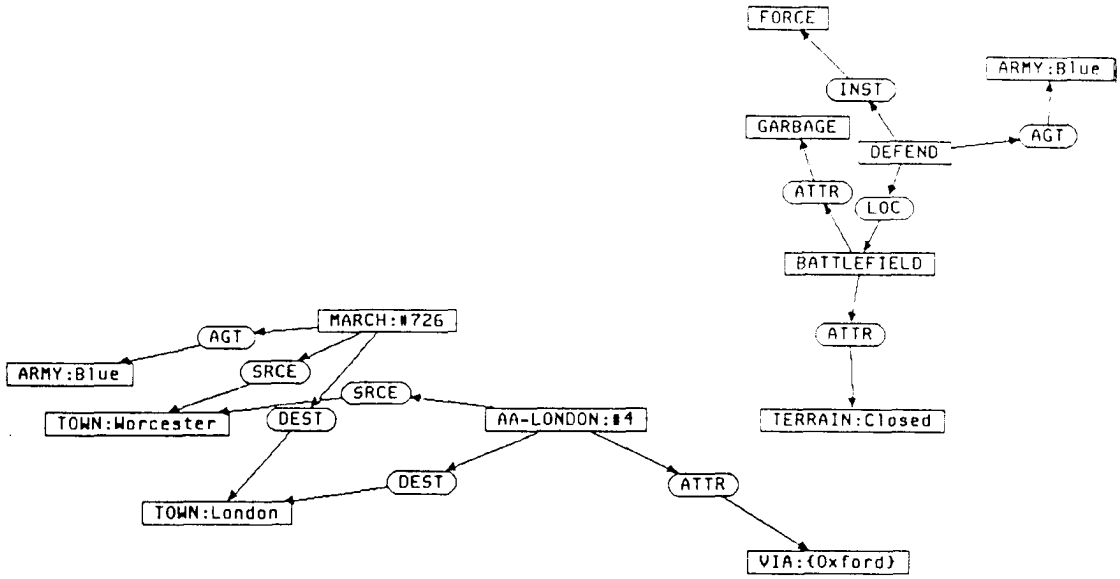

Figure 12. Blue's two assumptions

The first assumption, however, produces no such additional information. The assumption that Blue will march from Worcester to London via Oxford does not clash with the models of Red's likely choices. The interpretation of this result is that the assumption made is too weak.

5.2.2.1 The Choice of a Battle Site

The second assumption is purposely stronger. It states the desired intention of Blue to defend any battlefield chosen by Red with a particular type of force, and that the battlefield should be in Closed terrain (i.e. unsuitable for cavalry), and be flat land. This assumption is coherent with the BLITZ model, producing little of interest, but is incoherent with the FIELD model. This incoherence triggers MGR into generalizing the assumption, since it is assumed to be too strong. The generalization, when respecialized with the intelligence facts that the model was built on, produces a model with the definition of a any battle joined in which, as its constraint overlay, chooses a battle site by correlating the set of towns on the avenue of approach, and the type of terrain. In this case, the terrain is forced back to 'Open', and the appropriate battleground is found to be Edgehill (Fig. 13).

5.2.3 Summary of the Example

We have shown how MGR can partially automate the production of baseline models and then then allow projections using these models together with assumptions

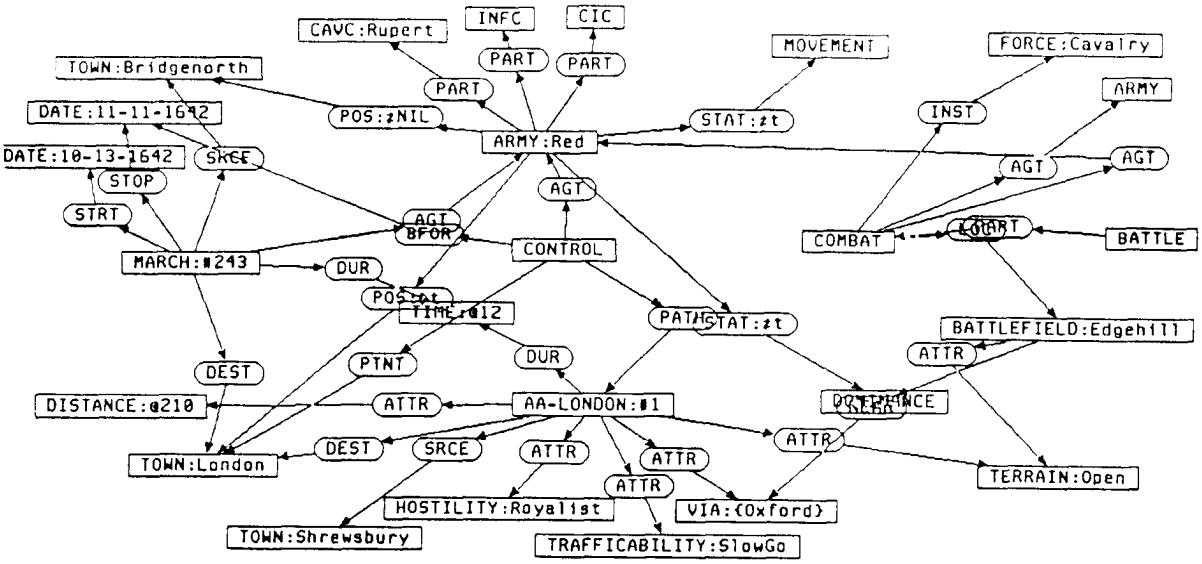

Figure 13. The final model

made by the user. The baseline models assess the feasibility of Red's alternative tactics, given the amount and type of intelligence available. The projection models allow evaluation of Blue assumptions as coherent with baseline models, indicating a weak assumption, or incoherent, indicating an over-strong assumption. In the latter case, MGR will attempt a generalization possibly resulting in a new model.

REFERENCES

Brachman, R. J., 1985: "1 lied about the trees", or, Defaults and Definitions in Knowledge Representation, AI Magazine, Fall 1985, 80-93.

Coombs, M. J., and J. L. Alty, 1984: Expert Systems: An Alternative Paradigm, International Journal of Man-Machine Studies, 20, 21-44.

Coombs, M. J., and R. T. Hartley, 1987: The MGR Algorithm and its Application to the Generation of Explanations for Novel Events, International Journal of ManMachine Studies, 27, 679-708.

Coombs, M. J., and R. T. Hartley, 1988: Explaining Novel Events in Process Control through Model Generative Reasoning, International Journal of Expert Systems, 1, 89-109.

Coombs, M. J., C. A. Fields, and G. McWilliams, 1988a: Artificial Intelligence Methods for Optimizing the Use of Meteorological Databases: Recommendations for Implementing the MERCURY System, ASL Report CR-88-0034-2, US Army Atmospheric Sciences Laboratory, White Sands Missile Range, NM 88002-5501.

Coombs, M. J., J. R. Thompson, R. T. Hartley and T. Fichtl, 1988b: A Software Environment for Tactical Situation Analysis, White Paper, Computing Research Laboratory, New Mexico State University, Las Cruces, NM 88003-0001.

Fields, C. and E. Dietrich, 1988: Engineering Artificial Intelligence Applications in Unstructured Task Environments: Some Methodological Issues. In D. Partridge (Ed.), *Artificial Intelligence and Software Engineering*, Erlbaum, Hillsdale, NJ.

Fields, C., M. J. Coombs, and R. T. Hartley, 1988: MGR: An Architecture for Problem Solving in Unstructured Task Environments, Proceedings of the Third International Symposium on Methodologies for Intelligent Systems, Elsevier, Amsterdam, 44-49.

Hayes-Roth, F., D. Waterman and D. Lenat (Eds.), 1983: Building Expert Systems, Addison Wesley, Reading, MA.

Pitz, G. F., and N. J. Sachs, 1984: Judgement and Decision: Theory and Application, Annual Review of Psychology, 35, 139-163.

Sowa, J. F., 1984, Conceptual Structures, Addison Wesley, Reading, MA.

Thompson, J. R., R. Hopf-Weichel, and R. E. Geiselman, 1983: The Cognitive Basis for Intelligence Analysis, OSD Report No. R83-039C, LOGICON Inc., Woodland Hills, CA 91367.

Thompson, J. R., R. Trout, and B. Landee-Thompson, 1986: Artificial Intelligence Applications for Sensor Data Fusion, Report A002, Perceptronics, Woodland Hills, CA 91367.

Thompson, J. R., B. Landee-Thompson, T. C. Fichtl, and L. Adelman, 1988: Measurement and Evaluation of Military Intelligence (MI) Unit Information Processing Performance: Year I Technical Report, Report No. 88/6504, Science Applications International Corporation, Alexandria, VA 22333-5600.