Proceedin gs of Army Science Conference, Dec 2-5, 2002, Orlando, FL

DIAGRAMMATIC REASONING IN SUPPORT OF SITUATION UNDERSTANDING AND PLANNING

B. Chandrasekaran, John R. Josephson, Bonny Banerjee and Unmesh Kurup
Department of Computer & Information Science
The Ohio State University
Columbus, OH 43210

Robert Winkler
U. S. Army Research Lab
2800 Powder Mill Rd
Adelphi, MD 20783

ABSTRACT

Visual representations consisting of terrain maps with an overlay of diagrammatic elements are ubiquitous in Army situation understanding, planning and plan monitoring. The Objective Force requirements of responsiveness, agility and versatility call for digitized graphical decision support interfaces that automate or otherwise help in various reasoning tasks. The main purpose of the paper is to describe the issues involved in building a diagrammatic reasoning system, specifications for a diagrammatic representation formalism, and an architecture for problem solving with diagrams. We have begun implementation of an application that infers maneuvers from data, obtained from an exercise at the National Training Center, about locations and motions of Blue and Red forces. We present algorithms used in the initial stages of an implementation.

1. INTRODUCTION

Visual representations with overlay of diagrammatic elements are ubiquitous in military decision-making. Commanders represent their situation understanding and intended maneuvers and plans, and monitor the action, on maps that contain terrain and other mission-relevant information. In the military context, diagrams are deemed so important that field manuals have standardized the elements of such representations. Diagrams are overlaid on top of terrain maps, and they indicate, using a combination of iconic and spatially veridical elements, information such as movements of units, locations and identities of enemy and friendly units, regions of control, synchronization points, and so on. Figure 1 shows a diagram that might be part of a course of action description. Diagrams abstract away details that are not essential to a reasoning task and highlight those that are, and by means of symbolic elements (such as attached labels and iconic diagrammatic elements), they point to relevant pieces of conceptual information. Diagrams also serve as augmentation of working memory. Commanders combine visually obtained information with other

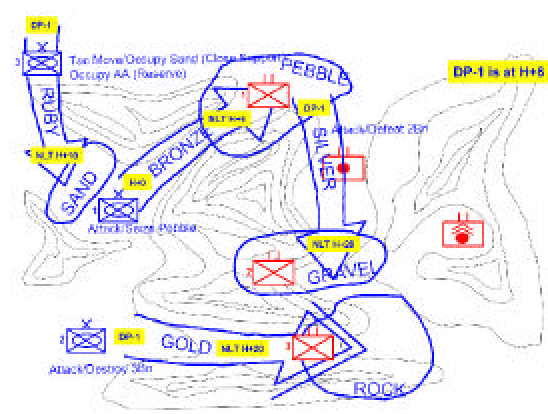

Figure 1. A COA diagram using Army standard symbology.

conceptual knowledge to make a complex series of inferences to help them predict, evaluate and plan. The whole process is so natural that we often fail to appreciate the complexity of the cognitive activities involved. Nevertheless, understanding and formalizing the processes involved in such apparently effortless reasoning is necessary if we wish to provide effective decision support systems for a commander and his staff. Intelligent computer support that, among other things, relieves the commander by providing a degree of automation in basic inferences, alerts him and his staff to rapidly developing potential threats to friendly units and plans, attempts to infer enemy intentions, monitors plans, and presents the results in such a way that salient information visually stands out can be a very valuable technology for Army transformation. Diagrammatic reasoning (Glasgow, et al, 1995; Chandrasekaran, 1997; Chandrasekaran, 2002) is a relatively new area of research in Artificial Intelligence.

In this paper, we discuss the nature of diagrammatic representations, describe an architecture for combining diagrammatic and conceptual inferences to solve problems, and describe some algorithms using these ideas in a military application. The application starts with data about the motions over time of a large number of Blue and Red units taking part in an exercise at the National Training Center. Our long-term goal is to experiment with the diagrammatic reasoning architecture in inferring from the data an account of what is happening on the field – what maneuvers might be under way, and what the goals and plans of the two sides might be.

2. WHAT IS A DIAGRAM?

2.1. Diagrams as Abstractions

Diagrams are used to reason about non-spatial problems as well as spatial problems. In the former, if a suitable mapping exists between certain properties and relations in the domain to two-dimensional spatial properties and relations, then situations in the domain may be represented pictorially. If the domain is a spatial one to start with, such as reasoning about activities on a battlefield, the needed mapping is trivially present. We focus on problems of the latter sort in this paper.

The distinctive properties of diagrammatic representations in spatial domains may be understood by contrasting a map of a region with an aerial photograph of the same region. The map and the photograph agree spatially on several things, e.g., the distances between pairs of spatial locations are supposed to be the same (modulo the scales) in both representations. There are however, several important differences.

1. *Abstracting into (spatial) objects*. The map identifies certain locations, lines and regions as *objects,* i.e., as things that may be named and talked about, such as cities, highways, ponds, and so on. The photograph is silent about these – we may look at the aerial photograph and recognize some part of it as a mountain or a lake, but the ontology of the representation does not commit to objects in the world.
2. *Abstracting away spatial information not relevant to the task*. The map abstracts away much of the spatial information that would be found in the aerial photograph. For example, an aerial photo might show the fine structure of the curves on the edge of a road, the gravel on land surface, or the reflection of sunlight off the water of a lake, but a travel map would not contain this kind of information.
3. *Iconization or attaching symbols to visual objects*. The identity of the objects – either as individuals or as types – is often provided by means of iconic elements associated with the spatial representation of the objects. For example, at the location where a church is located, the map might have a Church icon. The location of the (center of the) icon is intended to be spatially veridical, but the shape of the church icon is not intended to represent the actual shape of the church. Similarly, a curved line on a map might have a specific thickness – the axial part of the curved line might be intended to be representative of the axis of a road, but the thickness might simply be iconic for "toll road." It is also common for alphabetical or symbols to stand for identity or other symbolic features, as in the example where the numeral "26" close to an intersection point on the line for a road might represent the exit number. Some other common techniques include the use of colors and/or stipple patterns. MIL-STD-2525B specifies the icons to be used in diagramming military activities.

It is important to stress the task-dependent nature of the abstractions that distinguish a map from a terrain photograph. The task for which the representation is to be used determines the object types that are relevant and also the spatial properties and relations that need to be retained in the abstraction process. For a given region, the map produced for tourism and automotive travel would be different from the map produced for Army exercises. For Army exercises, a certain topography might be abstracted into a line object for a possible avenue of approach, the front line, regions that are and aren't navigable for various vehicles, and so on. The tourist map might focus on objects corresponding to toll roads, interstates and sites of tourist interest.

2.2. How Diagrams Help

A specific way diagrams help is by virtue of what might be called the "emergent objects and relations" phenomenon in diagrams. If we place two line objects on the diagram (standing, say, for two roads) in such a way that they intersect, the intersection point is a new point object that is present. Similarly, if a line object intersects a region object, several new objects are created: the intersection points, the line segments, and the two new (at least) sub-regions. All these new objects are available for visual perception to pick up. In a similar vein, if object A

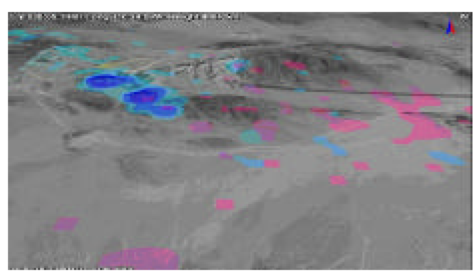

Figure 2. "Blobs" representing groups in an exercise.

is placed to the left of object B in the diagram, and object C is placed to the right of object B, the spatial relation that A is to the left of C doesn't require any inference, visual perception can pick up the information. The capacity of a visual representation in conjunction with visual perception to "see" the emergent objects and relations has been called "free ride" in the diagrammatic reasoning literature, and is one of the reasons why diagrams, when they are applicable, are so attractive as representations. However, this capability is not without its costs. One can err by treating such visually available results as applicable to the general problem at hand, since the diagram may be simply a special case of the general situation. In these cases, usually additional conceptual reasoning is necessary to establish the degree of generality of the information that visual perception gives. These issues are beyond the scope of this paper, but are only mentioned to caution that in general, diagrammatic representations do not stand alone and need to be combined with conceptual reasoning to solve problems.

The task-dependent nature of the abstractions in the diagram or map is relevant for the application domain that we focus on later in the paper. ARL researchers had earlier used the exercise data (Emmerman, et al, 1998) to experiment with a display technique called "blobbing," in which the micro states of individual battlefield entities (e.g. their position and combat power or operational lethality) are aggregated into macro states represented visually as blob-like abstractions. When visualized over a temporally compressed time frame, these blob-like abstractions give a sense of aggregate group motion. Figure 2 illustrates the blobs as seen at a specific instant of time. (In black and white, the blobs may be hard to see – the Blue blobs appear as dark patches in the NW quadrant.) As events happen on the battle field, the temporally compressed fluid-like motions of these blobs give a better sense of the macro events occurring at the higher levels of aggregation captured by the blobs. The blob representation is a kind of abstraction -- they abstract away the details about individuals, but retain information about the spatial extent and density (in terms of any aggregate spatio-temporal metric: number present, operational lethality, combined ammo supply, etc.) of the groups. Thus they can provide some of the elements of a diagrammatic representation of a situation where contiguous spatio-temporal information is important. In Section 4.2, we discuss defensive positions where regions that are occupied are important – for this kind of information, the blob extraction process can be useful.

3. THE ARCHITECTURE

3.1 The Problem Solving Architecture

As a rule, in diagrammatic reasoning, information that can be obtained by visual perception from the diagram (in Psychology, the term "data or stimulus driven information" is used) is combined with other information, most commonly in the form of conceptual information (so-called "symbolic", "linguistic", or "concept or knowledge driven" information – none of these terms is entirely adequate – but think of it as the kind of information that might be communicated by sentences in natural language), to make inferences relevant to the problem-solving goal. For example, a commander might make a visual inference– such as that a friendly unit F is coming within the firing range of enemy unit E— from a situation diagram, and combine this with conceptual knowledge– such as that the lethality of E has been decreased by 90% due to an earlier engagement – to make further inferences relevant to the problem solving goal – such as that the risk of attack on F by E is small. Figure 3 is a schematic of a general architecture that supports this kind of goal-driven, bi-modal inference making.

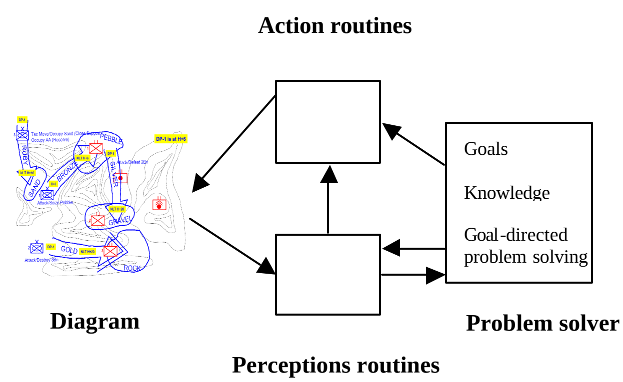

Figure 3. Problem solver working with a diagrammatic representation

The problem solver sets up subgoals, some of which might be achieved by visual perception on the diagram. A set of perceptual routines is available to operate on the diagrammatic representation to obtain the information for the subgoal. Action routines are used to introduce and modify diagrammatic objects to the representation, such as when a commander might propose an avenue of approach by drawing a suitably iconized line on the diagram. So far in our current work we have not looked into the issues surrounding constructing or modifying diagrams during problem solving, so this module will not be discussed further in the paper.

3.2 The Diagrammatic Representation System (DRS)

The diagrammatic representation system (DRS) is intended to be a domain-independent system for representing diagrams in general.

A diagram is pair (*I*, DDS), where:

I is the *image*, defined as a specification of intensity values for points in (the relevant region of) 2-D space; and

DDS is the Diagram Data Structure, which is a list of labels for the diagrammatic objects in the image, with a pointer from object label to the subset of *I* that corresponds to the object. A diagrammatic object can be one of three types: *point*, *line*, and *region*. Point objects only have location (i.e., no spatial extent), line objects only have axial specification (i.e., do not have a thickness), and region objects have location and spatial extent.

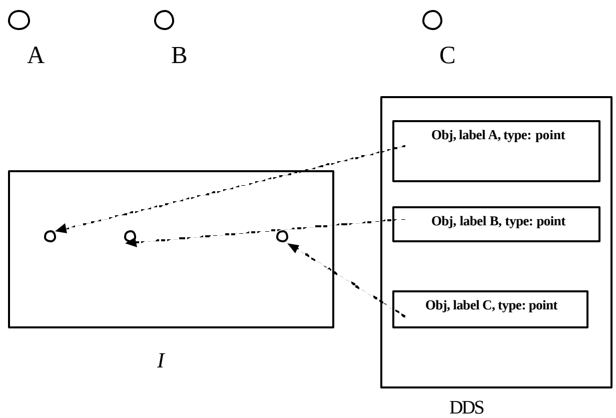

Figure 4. Top, a diagram consisting of 3 labeled circles, standing for points; bottom, the diagram's representation in DRS.

Figure 4 gives an example of the representation in DRS of a simple diagram consisting of three labeled points. At this level of description, there is no commitment at the level of data structures for representing the image, I, in the computer – all that the definition requires is that the intensity values for relevant points in the 2-D space can be obtained from I.

3.3 Perceptual Routines

As mentioned earlier, a power of diagrammatic representations is that they make explicit the emergence or vanishing of new objects and spatial relations in a diagram as the representation is constructed or modified, and that visual perception should be able to access this emergent information. The following is a partial list of domain-independent *perceptual routines* (the term is due to Ullman, 1984) that we are in various stages of implementing.

1. *New object recognition and extraction routines*: Intersection-points between line objects, region when a line closes on itself, new regions when regions intersect, new regions when a line intersects with a region, extracting distinguished points on a line (such as end points) or in a region, extracting distinguished segments of a line (such as those created when two lines intersect), extracting periphery of a region as a closed line. Reverse operations are included – such as when a line is removed, certain region objects will no longer exist and need to be removed. Object property extraction: Length-of (line) (this will return a number), Straight-line (Line) and Closed (Line) will return True or False. Additional property extraction routines can be added as needed. A design desideratum for us is to keep as many of the routines as possible domain-independent.
2. *Relational perception routines*: Inside (I_1,I_2), Outside, Left-of, Right-of, Top-of, Below, Segment-of (Line1, Line2), Subregion-of (Region1, Region2), On (Point, Line), and Closed (Line1).
3. *Translation, rotation and scanning routines may be combined with routines in 1 and 2*. For example, Intersect (Line, Rotate (90deg, Line 2)) will decide if there is an intersection point between Line1 and Line2 if the latter is rotated by 90 degrees. Inside (Region1, translate (5 units to the right, Region2)) will decide if Region2 translated as specified will be within Region1.

3.4 Problem Solver

The problem solver uses domain knowledge (about maneuvers, and plans for example) and problem solving strategies, such as abductive inference (Josephson and Josephson, 1994) and planning, to decompose a problem-solving goal (such as recognizing and characterizing a maneuver) into subgoals, query the diagrammatic system for information relating to the subgoal and make inferences by combining information from diagrammatic and conceptual sources. AI and cognitive science have developed many problem-solving architectures that can be used as the basis for implementation of the problems solver: Soar and Act-R being among the most versatile for our purposes. What problem solving strategies are implemented in the problem solver depends on the problem. We model maneuver recognition, our initial application focus, as an instance of abductive inference, and accordingly, our problem solver incorporate strategies appropriate to it, such as generating elementary hypotheses from local data, maintenance of alternative interpretations at almost all stages of problem solving, seeking consistency of interpretations over larger regions, etc.

4. THE APPLICATION

4.1 Reasoning about Maneuvers

In this kind of AI research, the development of the representations and the inference architecture needs to be driven by an application that exercises the many components of the overall system. As indicated earlier, in the application that drives our research, we start with data about the locations and movements over a number of sampling instants of the individuals and vehicles of blue and red sides participating in an exercise at the National Technical Center. We also have terrain information. For the first set of experiments, we are interested in making hypotheses about the maneuvers that are being undertaken. This task is an intermediate stage in making hypotheses about the plans of the sides.

This application was chosen because it has several attractive properties to drive the basic research on diagrammatic representation and inference: it is knowledge and inference-rich, and involves an intimate integration of diagrammatic and conceptual reasoning. Further, it fits in nicely with earlier research at ARL (Emmerman, et al, 1998) using the same data on presenting battle information to the user in a more comprehensible way than as motions of thousands of points over a terrain. Eventually, our technology could be used in various ways in the military domain. For example, a reasoning system might monitor the actual evolution of a battle with respect to the intended maneuvers and goals, or unanticipated developments with respect to enemy plans, and alert decision makers who might be overloaded with processing incoming information. However, identifying applications for deployment is not the primary aim in choosing this application.

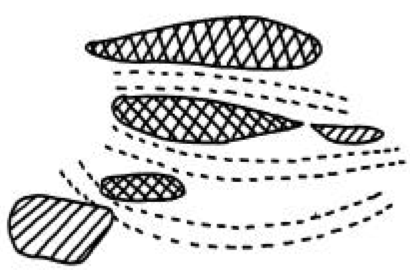

Figure 5. A diagram of the terrain. Double-hashing indicate "no-go" regions, and single-hashed ones are "slow-go." The dotted lines indicate navigable paths. The terrain diagram might also include other elements such as military installations, rivers, etc.

4.2 Building the Diagram

There are several different aspects of the situation and types of information that need to be diagrammed. First is the diagram of the terrain. Diagramming the terrain is similar to constructing a map (see discussion earlier), emphasizing abstractions relevant to military reasoning. This would consist of regions marked off as off limits for various reasons: mountains, not supportive of certain types of vehicles, rivers, etc. The diagram would also mark possible avenues of approach, and friendly and enemy regions and points of interest, such as cities, forts, etc. These are relatively static entities, and such a diagram can be constructed in advance. A terrain diagram corresponding to the terrain in Figure 2 is given in Figure 5.

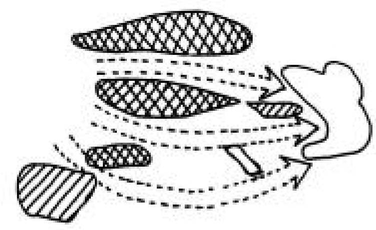

Figure 6. Terrain diagram with an overlay of Red defensive positions (unhatched regions), and avenues of approach (dotted arrows) for Blue towards the Red objective on the right.

A diagram of the action needs to be overlaid on the diagram of the terrain. To diagram the action, it is useful to distinguish between different kinds of activities that take place before, during and after the battle, and the kind of motions that they involve. There is "movement to contact," where a group is moving towards an enemy unit or some objective. There are defensive movements, where groups move to position themselves according to a defensive plan to be followed when contact with the enemy occurs. Then there are motions after contact begins, where the motions are determined by the interactions between the individuals involved in combat. Finally, there are movements associated with post-contact activities, such as retreat, and so on. While all of these involve group motions, and all motions are significant for some class of inferences, their characteristics are rather different, and they are informative for different inferential goals.

To describe the battle at the level of the plans and goals of the two sides, motions corresponding to movement to contact and motions of the defending side are important, but the latter are important only to the degree that they tell what the final defensive positions are. From a diagrammatic point of view, motions to contact are best described as lines of motion of significant groups, whereas defensive positions are best described as regions that block or threaten avenues of approach, and lines that describe defensive perimeters. While the motions during battle may be useful to describe its details, with respect to goals and outcomes, they can be replaced with simpler lines corresponding to any net motion.

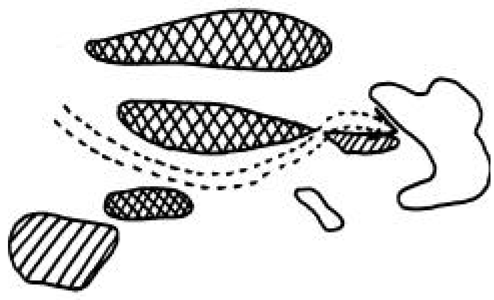

Figure 7. Diagram of Red defensive positions and "movement to contact" (dotted arrow) by Blue overlaid on the terrain map. (Blue motion is simplified version of motion computed by the algorithm described in the text.)

Attached to the various diagrammatic objects (such as lines and regions) will be symbolic abstractions of various kinds – such as identity, size, lethality, etc. – as needed for the inferential goals.

As mentioned, movements of groups to contact can be best represented by line objects. We will shortly describe our current work in automatically constructing such diagrams of motion. Defensive positions can be represented by region objects standing for the spatial extent of the groups. As discussed earlier, blob abstraction algorithms described in (Emmerman, et al, 1998) can be useful for this purpose.

Figure 6 shows the overlay of Red defensive positions (unhatched region objects) on the terrain map. Because of the knowledge of approximate Red positions, the navigable routes in Figure 5 now become potential avenues of approach to objective for Blue forces.

Figure 7 shows the overlay of Blue's movement to contact towards the Red objective on the right. (The avenues of approach in Figure 6 are not shown in this figure.) This diagram of Blue's movement was based on the working of the algorithm described below. In our current implementation, the entire focus so far has been on constructing a diagram of motions of groups, which requires organizing the numerous individual agents on both sides into meaningful groups at different levels of aggregation and representing their motions as lines of motion. An initial version of the diagram construction component has been completed.

Our top-level computational strategy for extracting a diagrammatic representation of the movements of groups at the different levels of organization is as follows (see Figure 8):

1. For each time instant, we generate a set of good hypotheses about meaningful groups at different levels of organization, based on proximity, similarities of identities[1] and velocities of the units.
2. From the grouping at each level, we extract a consistent account of groups and hence draw lines describing the motions of the centroid of each group in order to obtain the desired diagram.

A variety of clustering algorithms are available to generate the grouping hypotheses. In our work, we used a variation of the Kohonen Self-Organizing Maps. Our variation is called Information-Extracting Self-Organizing Neural Network. The details of this clustering algorithm and how it was used to generate grouping hypotheses are omitted here, but can be found in (Banerjee, 2002).

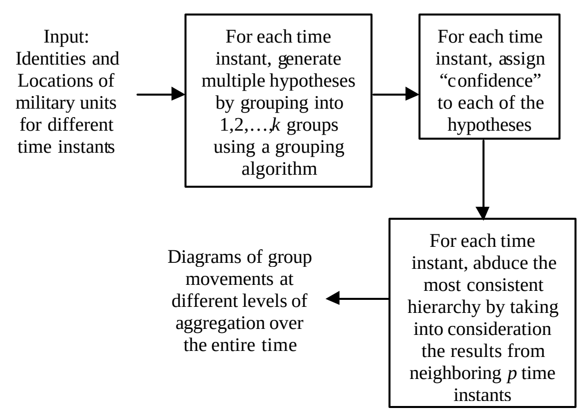

Figure 8. Algorithm for group identification and diagramming group motions.

Grouping should also be consistent over a range of time instants, both prior and after the time instant under consideration. At any given time instant, our algorithm maintains a small list of best grouping hypotheses (rather than pick one best grouping hypothesis) and consistency is imposed over several time intervals by picking hypotheses at each instant that are compatible with the interpretations for the neighboring instants. Additional domain knowledge may be also used to enforce consistency.

[1] The only identity information currently used is about the side, Blue or Red, to which the entity belongs.

The problem is complicated due to many reasons. At a given time instant, the large number of military units (on the order of 10^3) means that clustering with accuracy calls for substantial computational resources. The input data sets are noisy for a variety of reasons: GPS receivers limited to 100m accuracy, cloud cover affecting GPS signals, e.g., or because even when a vehicle is destroyed, its GPS signal continues to arrive. We needed to adopt a variety of strategies to cope with such problems.

An important issue is the local zigzags in motion. For certain purposes, all we might need is a straight line pointing to the objective, while for others, it may be important to retain the zigzags that correspond to terrain. Filtering the directional variations is easy, but deciding what needs to be retained and what needs to be discarded is somewhat dependent on the kinds of additional problem solving that would need to be supported.

Many alternate groupings are possible with the same input data at any given time instant. Additional information in the form of velocity can help filter out some of the ambiguities and thus create more robust group hypotheses. Velocity of each military unit is calculated from the input data (identity and location) given at two consecutive instants of time. However, due to the measurement error inherent to the input data, this instantaneous velocity information receives a much lower weight compared to the identity and proximity information. In our experiments, we found that identity and locality are more important for group hypothesis formation than velocity, which is mostly useful for capturing local changes in direction.

4.3 Problem Solving to Recognize and Characterize Maneuvers

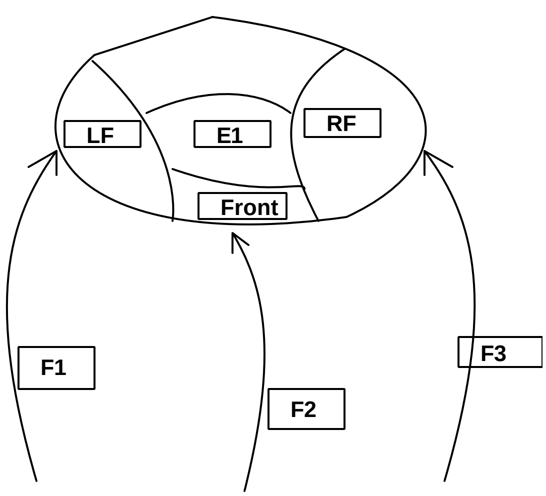

Figure 9. Double-envelopment. LF – Left Flank, RF – Right Flank, E1 – Enemy, F1, F2, F3 - Friendlies

The Problem Solver (PS) has domain-specific knowledge about how to recognize and characterize maneuvers. Part of this knowledge is in the form of a number of maneuver types, and decomposition of each maneuver type into various subtasks, specifically into recognition of various spatial and non-spatial components called features. The underlying problem solving strategy used here is abductive inference. This strategy calls for making hypotheses about presence or absence of maneuver components in local regions, maintaining alternative hypotheses when the local hypothesis is not certain. It then combines local inferences to build a coherent and consistent picture. In this process, it makes use of alternate hypotheses it created during its local inferences. The spatial features are solved by querying the DRS while the non-spatial features are solved by using a combination of inferences and querying a database/user.

Consider the double-envelopment (d-e) maneuver shown in Figure 9. In double-envelopment the attacking side directs its decisive attack on the flanks (typically weaker than the front) of the enemy while keeping the front lines of the enemy occupied by a smaller shaping force. This maneuver can be divided into the following features – **left-flank-attack**, **right-flank-attack** and **frontal-attack**. Each feature can be further divided into sub-features. Figure 10 shows the sub-features associated with **left-flank-attack** along with the functions used by the problem solver to recognize these sub-features. The PS also allows the user to incorporate certainty information into its reasoning. This allows the PS to take into account any doubts about the reliability of the data on which it bases its findings.

Left-flank-attack sub-features

1. Is there a movement of an opposing force towards the left flank of the current force F1?
   - Do we know the left flank of the object?
     Has-property(F1, LF)
   - Is there a movement of an opposing force towards the left flank?
     (Ex)Intersect(F1,x) & type(x)=Opp(type(F1))

2. Does this movement have sufficient lethality?

   Lethality(x) = Lethality-LF

Figure 10. Left-flank-attack subfeatures. Has-property(), lethality, type() and opp() are database/PS calls., intersect is a DRS call

During the recognition process, different maneuver hypotheses might match a given set of features. For instance, a left-flank attack would fit two hypotheses: a single-envelopment a double-envelopment. Also, an evolving motion might partially match a number of different feature hypotheses: a left-flank attack with respect to an objective, a right-flank attack with respect to another objective and so on. The integration stage of the problem solver combines the various local inferences to form a coherent consistent high-level picture of the diagram. This stage also has the responsibility of maintaining consistency in the face of newer external information. For instance, if we are sure (from information from other sources) that a particular maneuver is going on at a specified location, we must be able to incorporate this new information into our reasoning without having to restart the entire reasoning process.

5. CONCLUSION

We have discussed a set of issues related to reasoning with diagrams, a subject of significant importance to providing decision support for commanders planning and carrying our various military engagements. We described a diagram representation system, identified a set of domain-independent perceptual operations, and described a computational architecture for problem solving with diagrams. Problem solving takes place by combining information from the diagram with conceptual knowledge to make appropriate inferences. We are motivating the research and technology development by implementing a system that would construct appropriate diagrams from data about the locations and motions of vehicles and individuals engaged in a military exercise, and use the diagrams to reason infer maneuvers, plans and intentions of the two sides.

There are at least three ways in which research of the type reported could help in building decision support systems. The first way is to help build decision aids that relieve the attentional demands on the decision maker by generating hypotheses about emerging threats and deviations from expected behaviors. The second way is by constructing diagrammatic abstractions that summarize a vast amount of detail, such as suggested in Sec. 4. 2. Third, insights about diagrammatic abstractions that are salient and supportive of specific types of inferences can be provided by the computational and reasoning research of the type discussed.

We described our current, preliminary implementation. Over the next few years, the diagrammatic representation system and the perceptual routines will be built so that they can be used for a variety of problems in which diagrammatic representations can be helpful. With respect to the specific problem of inferring plans, we intend to make progress along many fronts including construction of more complete diagrams that describe defensive as well as offensive aspects using standard MIL-STD-2525B symbology; and a problem solver that is more robust with respect to handling local uncertainty to produce globally more robust solutions.

ACKNOWLEDGEMENTS

The research of the OSU team was supported by participation in the Advanced Decision Architectures Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAD19-01-2-0009. We are also thankful to Col. Jack Gumbert, the Director of Ohio State University Army ROTC program, for his time, and to Jim Walrath for his comments that helped improve the paper.

REFERENCES

Banerjee, Bonny, “Generating Diagrams of Group Motions from Locations and Movements of Individuals,” Master’s Thesis in Electrical Engineering, The Ohio State University, 2002.

Chandrasekaran, B., “Diagrammatic Representation and Reasoning: Some Distinctions”, keynote paper presented at the *AAAI Fall Symposium Series*, Diagrammatic Reasoning, Boston, MA, 1997.

Chandrasekaran, B., "What does it mean for a computer to do diagrammatic reasoning?", keynote talk given at the *Diagrammatic Reasoning 2002 Conference*, Callaway Gardens, GA, April 18, 2002.

Glasgow, Janice, N. Hari Narayanan, B. Chandrasekaran, editors, "Diagrammatic Reasoning: Cognitive and Computational Perspectives," AAAI Press/ The MIT Press, 1995, 780 pages.

Josephson, J. R. and S. G. Josephson, *Abductive Inference: Computation, Philosophy, Technology*, New York, Cambridge University Press, 1994.

Emmerman, P.J., J.D. Walrath, and R.P. Winkler, “Making Complex, Multidimensional Battlefield Information Intuitive,” 21st Army Science Conference, 1998, Norfolk, VA.

Ullman, Shimon, “Visual Routines,” *Cognition*, 18:97--159, 1984.