

Sketching for Military Courses of Action Diagrams

Kenneth D. Forbus

Jeffrey Usher

Vernell Chapman

Qualitative Reasoning Group

Northwestern University

1890 Maple Avenue,

Evanston, IL, 60201 USA

847-491-7699

forbus@northwestern.edu

ABSTRACT

A serious barrier to the digitalization of the US military is that commanders find traditional mouse/menu, CAD-style interfaces unnatural. Military commanders develop and communicate battle plans by sketching *courses of action (COAs)*. This paper describes *nuSketch Battlespace*, the latest version in an evolving line of sketching interfaces that commanders find natural, yet supports significant increased automation. We describe techniques that should be applicable to any specialized sketching domain: *glyph bars and compositional symbols* to tractably handle the large number of entities that military domains use, specialized *glyph types and gestures* to keep drawing tractable and natural, *qualitative spatial reasoning* to provide sketch-based visual reasoning, and *comic graphs* to describe multiple states and plans. Experiments, both completed and in progress, are described to provide evidence as to the utility of the system.

Categories & Subject Descriptors: H.5.2 User Interfaces – Interaction styles, I.2.4 Knowledge Representation Formalisms and Methods, I.2.10 Vision and Scene Understanding – perceptual reasoning,

General Terms: Algorithms, Human Factors, Design

Keywords: Sketch understanding; multimodal interfaces; nuSketch; qualitative reasoning; analogy; spatial reasoning

INTRODUCTION

Sketching provides a natural means of interaction for many spatially-oriented tasks. One task where sketching is used extensively is when military planners are formulating battle plans, called *Courses of Action (COAs)*. This paper describes a system we have built, *nuSketch Battlespace* (nSB), which provides a sketching interface for creating COAs. It is based on the *nuSketch* architecture for sketching outlined in [14,18], but represents a generational advance over the system described in [9]. We start by outlining the problem and our approach to it, contrasting our

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IUI'03, January 12–15, 2003, Miami, Florida, USA.

Copyright 2003 ACM 1-58113-586-6/03/0001...\$5.00.

understanding-based approach with the more traditional recognition-based approach for multimodal interfaces. Next we describe the engineering techniques that enable us to avoid using recognition technologies. Then we discuss two of the more powerful features of nuSketch Battlespace: Our spatial reasoning system and the *comic graph* visual representation. Experiments with military users are then summarized, and implications and future work are discussed.

The problem

A COA consists of a sketch and a textual statement. The sketch conveys a number of crucial properties of the situation and the plan. First, it includes a depiction of what terrain features are considered important. (Sometimes COAs are drawn on acetate overlays on maps, sometimes the basic terrain description itself is simply sketched.) The results of analyzing terrain, such as possible paths for movement (*mobility corridors, avenues of approach*) and good locations for different kinds of operations are identified. The disposition of troops and equipment, both for friendly (Blue) forces and what is known about the enemy (Red) forces is shown by means of *unit symbols*, a vocabulary of graphical symbols defined as part of US military doctrine. This graphical vocabulary also includes symbols for tasks, such as destroy, defend, attack, and so on. The COA sketch indicates a commander's plan in terms of the tasks that their units are assigned to do. The COA statement provides a narrative, describing why units are being assigned the tasks that they are ("Alpha will defend the Toofar Bridge in order to prevent Red from moving reinforcements across it") and timing information that would be difficult to express in the sketch.

Currently COAs tend to be created on pencil and paper, or on acetate overlays on maps with grease pencils, post-its, and pushpins. For larger echelons in unhurried situations, PowerPoint slides are sometimes generated later for communication. There has been no shortage of attempts to make computer software to speed the process of COA generation, but on the whole these systems have not been accepted by military users (cf. [23]). In all of our discussions with military personnel, they cite as a major problem the awkwardness of mice and menus for what is more naturally done by sketching.

Systems like QuickSet [2] and Rasa [23,24] provide strong evidence that multimodal interfaces could provide more

acceptable interfaces. QuickSet was used to describe the layout of forces in setting up simulated exercises, and was shown to be faster than using traditional CAD-style interfaces. Rasa was used to help command post staff track the positions of units (friend, foe, and neutral) based on field reports, and Marines preferred it to their traditional purely paper-based system. However, both simulation setup and unit tracking are simpler tasks than COA generation, which involves representing and reasoning about qualitative properties of terrain, hypotheses about enemy intent, and planning. Can multimodal technologies scale to the more complex problem of COA creation? Also, both of these systems involve the overhead inherent in today's recognition technologies: Significant investments in data collection are needed to train statistical recognizers for glyphs and for speech, users must be trained to use specific grammars and vocabulary (e.g., choosing a list of names that phrase lines can be drawn from in advance¹). Today's speech systems have serious problems in noisy environments, especially when operators are under stress. In our conversations with many active-duty military personnel, we hear repeatedly that if a system requires speech recognition, they simply will not use it. Can we find ways to provide the naturalness of sketching without speech? Our experience with nuSketch Battlespace indicates that the answer to both questions is yes.

THE nuSketch APPROACH

Most multimodal interfaces (cf. [1,2,19,22,23,26,27]) focus on recognition. Typically they combine input from several noisy channels (e.g., speech and gesture), using task context and mutual constraints between channels to disambiguate the input. For example, someone drawing a unit symbol for an armor battalion, while saying "armor battalion", might lead a system like QuickSet to create a new entity, coded as an armor battalion, in a database. This is clearly an important approach, and has been demonstrated to provide more natural and usable interfaces to a variety of legacy computer systems [1,2,23]. Progress in this approach includes improving recognition techniques for each modality and finding better ways to combine cross-modal information.

The nuSketch approach [14,18] is very different. Our focus is on visual and conceptual understanding of the user's input. We engineer around the recognition issues, since they are not our primary concern. Instead, we concentrate on enabling users to specify the spatial and conceptual aspects of some situation, in sufficient detail to support subsequent reasoning by AI systems on this input. Progress in our approach includes improving the visual processing and reasoning techniques and supporting richer reasoning about what is sketched.

¹ In [24] training times of 15 minutes were obtained, but by the system designers specifying the names of all the entities used (which means they would be in the grammars) rather than using names generated spontaneously generated by the operators.

These two approaches are of course complementary, and in the long run it would be useful to have the best of each in a combined system. However, we have found that it is possible to do surprisingly well with sketch-based interfaces that engineer around the need for recognition. The next section describes how we do this.

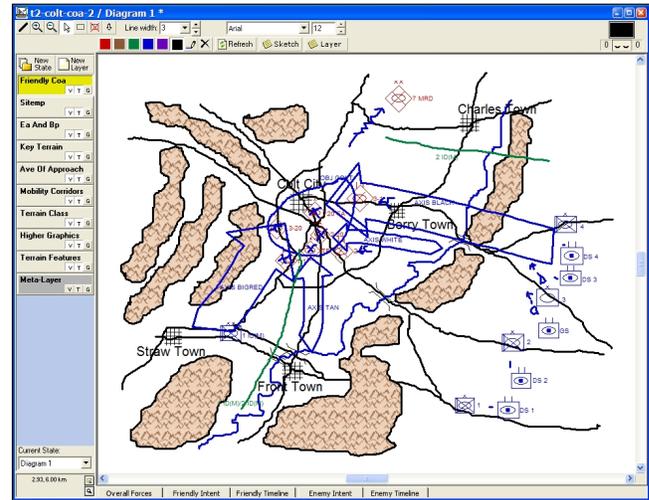


Figure 1: nuSketch Battlespace interface

INTERFACE OVERVIEW

Figure 1 shows the nuSketch Battlespace (nSB) interface. Many of the elements are standard for drawing systems (e.g., widgets for pen operations, fonts, etc.) and need no further comment. The crucial aspects that make the basic interface work are *layers* to provide a functional decomposition of the elements of a sketch, *glyph bars* for specifying complex entities, *gestures* that enable glyphs to easily and robustly be drawn, and *intent dialogs* and *timelines* to express a significant portion of the information in a COA narrative. We describe each in turn.

Layers

COA sketches are often very complex, and involve a wide range of types of entities. The use of layers in the nuSketch interface (Figure 2) provides a means of managing this complexity. The metaphor derives from the use of acetate overlays on top of paper maps that are commonly used by military personnel. Each layer contains a specific type of information: Friendly COA describes the friendly units and their tasks, Sitemp describes the enemy (Red) units and their tasks, Terrain Features describe the geography of the situation, and the other layers describe the results of particular spatial analyses. (nSB also sometimes produces new



Figure 2

layers that summarize its response to a query.) Only one layer can be active at a time, and the glyph bar is updated to only include the types of entities which that layer is concerned with. Clutter can be reduced by toggling the visibility of a layer, making it either invisible or graying it out, so that spatial boundaries are apparent but not too distracting. The chosen layer also controls the grammar used by the multimodal parser, which is used to both process input from the glyph bar and (optionally) spoken input².

Avoiding the need for recognition in glyphs

Glyphs in nuSketch systems have two parts. The *ink* is the time-stamped collection of ink strokes that comprise the base-level visual representation of the glyph. The *content* of the glyph is an entity in an underlying knowledge representation system that denotes the conceptual entity which the glyph refers to. Our interface uses this distinction to simplify entering glyphs by using different mechanisms for specifying the content and specifying the spatial aspects. Specifying the conceptual content of a glyph is handled by the *glyph bar*, while the spatial aspects are specified via gestures. We describe each in turn.

Glyph bars



Figure 3

Glyph bars (Figure 3) are a standard interface metaphor, but we use a system of modifiers to keep it tractable even with a very large vocabulary of symbols. The idea is to decompose symbol vocabularies into a set of distinct dimensions, which can then be dynamically composed as needed. For example, in nSB there are (conceptually) 294 distinct friendly unit symbols and 273 distinct enemy unit symbols. However, these decompose into three dimensions: the type of unit (e.g., armor, infantry, etc., 14 friendly and 13 enemy), the echelon (e.g., corps to squad, 7 in all), and strength (regular, plus, minus, or a percentage). Our glyph bar specifies these dimensions separately. Templates stored in the knowledge base for each dimension are retrieved and dynamically combined to form whatever unit symbol is needed.

Modifiers are also used to specify the parts of complex entities. Tasks, for example, have a number of roles such as the actor, the location, and so on. Widgets are added to the

glyph bar whenever a glyph with parts is chosen. They include combo boxes and type-in boxes for simple choices (e.g., echelon), with drag and drop supported for richer choices also (e.g., the actor of a task). This simple system enables users to quickly and unambiguously specify roles.

One unanticipated advantage of this approach is that we discovered that almost all of our military experts hated drawing unit symbols. They strongly preferred having a neat symbol drawn where they wanted it. Those who had tried ink recognition systems particularly appreciated never having to redraw a symbol because the computer “didn’t get it”.

Gestures

Multimodal interfaces often use pen-up or time-out constraints to mark the end of a glyph, because they have to decide when to pass strokes on to a recognizer. This can be a good interface design choice for stereotyped graphical symbols. Unfortunately, many visual symbols are not stereotyped; their spatial positions and extent are a crucial part of their meaning. Examples include the position of a road, a ridge line, or a path to be taken through complex terrain. Such glyphs are extremely common in map-based applications. Pen-up and time-out constraints are also problematic when the user is participating in conversations with other people, not just focusing their attention on the software. Our solution is to rely instead on manual segmentation. That is, we use a Draw button that lets users indicate when they are starting to draw a glyph. There are two categories of glyphs where pen-up constraints are used to end glyphs, but in general we require the user to press the Draw button (relabelled dynamically as Finish) again to indicate when to stop considering strokes as part of the glyph.

Types of glyphs

For purposes of drawing, glyphs can be categorized according to the visual implications of their ink. There are five types of glyphs, each with a specific type of gesture needed to draw them, in nSB: *location*, *line*, *region*, *path*, and *symbol*. We describe each in turn. In all cases, the start of the gesture is marked by pressing the Draw button, and most gestures require pressing the Draw button again to indicate when they are finished.

Location glyphs: The only visual property that matters in a location glyph is the centroid of its bounding box. Military units are an example of location glyphs: Their position matters, but the size at which they are drawn says nothing about their strength, real footprint on the ground, etc. Since such glyphs are drawn via templates, a gesture consisting of a single ink stroke is used to indicate where and how large they should be. (Users can of course move, resize, and rotate glyphs after they are drawn if desired.) Most other template-based glyphs (e.g., bridges, towns) are drawn as if they were location glyphs, although their size is considered significant in subsequent visual computations.

Line glyphs: Line glyphs represent one-dimensional entities whose width of their content, while important, is not tied to

² We have left the speech interface hooks in the system, to be ready for improved technology when it is available.

the width of their ink. Roads and rivers are examples of line glyphs; while their width is significant, on most sketches it would be demanding too much of the user to draw their width explicitly. The gesture for drawing line glyphs is to simply draw the line. Optionally, the line can be drawn as a number of distinct, disconnected segments, with gaps filled in via straight line segments. Our users found this ability to tacitly express a straight line very useful, since few of them are artists but prefer their diagrams neat.

Region glyphs: Both location and boundary are significant for region glyphs. Examples of region glyphs include terrain types (e.g., mountains, lakes, desert...) and designated areas (e.g., objective areas, battle positions, engagement areas, ...). The gesture for drawing a region glyph is to draw the outline, working around the outline in sequence. Multiple strokes can be used, with straight lines being used to fill in gaps.

Path glyphs: Paths differ from line glyphs in that their width is considered to be significant³, and they have a designated start and end. This information is used for queries in the spatial reasoner, since what is ahead or behind on a path can be of considerable importance in this domain. Path glyphs are drawn with two strokes. The first stroke is the medial axis – it can be as convoluted as necessary, and even self-intersecting, but it must be drawn as one stroke. The second stroke is the transverse axis, specifying the width of the path. Based on this information, nSB uses a constraint-based drawing routine to generate the appropriate path symbol, according to the type of path. (For example, main attacks use a double-headed arrow, while supporting-attacks use a single-headed arrow.) We used to require users to draw the outline of the arrow themselves, but this was intensely unpopular compared to them specifying only what was necessary and having our code fill in the details.

Symbolic glyphs: Symbolic glyphs don't have any particular spatial consequences deriving from their ink. They mainly are used to serve as a visual referent for abstract entities. Military tasks are an example. In some cases there are spatial implications intended by the person drawing it that would be missed with this interpretation, i.e., a defend task is often drawn around the place being defended. Unfortunately, the use of such conventions is far from uniform across the pool of experts we have worked with. Consequently, the most robust approach we have found is, as noted earlier, to use the glyph bar to specify the participants in a task, and not draw other spatial implications from the ink used to depict it. Symbolic glyphs can be drawn with whatever ink strokes the user desires⁴.

³ Often widths are specified to prevent moving units from being hit by friendly artillery and air strikes.

⁴ While there are specific visual symbols specified by doctrine for each task, we do not try to force users to use the “right” symbol. We view this as an opportunity to gather data on what spatial implications different users

Entering Other Kinds Of COA Information

In the military, courses of action are generally specified through a combination of a sketch and a COA statement, a structured natural language narrative that expresses the intent for each task, sequencing, and other aspects which are hard to convey in the sketch. In response to user feedback, we have provided facilities in nSB that provide some of this functionality.



Figure 4

The *Intent Dialogs* (Figure 4) enable the purpose of each task (both friendly and enemy) to be expressed. Intent is important in military tasks because it tells those doing it why you want it done. If, during execution, they decide that the task they were ordered to do won't accomplish that purpose, or that there is a better way, they will not do the specific task they were ordered to do but instead do something that better accomplishes the intent. Thus intent statements are a crucial part of a task specification. In orders, the general form is “<task specification> in order to <intent for that task>” After much consultation with military officers, we found that the following generic template successfully captures a surprisingly wide range of intents:

“<modal> <actor> <operation> <object>”

where <modal> = enable, prevent, maintain

<actor> = a unit or side (e.g. Alpha Brigade, Red)

<operation> = a list of event types, e.g., destroying, attacking, controlling, ...

and <object> = another COA entity, e.g., Red, Alpha Brigade, Foo Bridge, etc. The intent dialog enables such statements to be made for each friendly and enemy task⁵. Multiple <actor>s <object>s can be specified to handle conjunctions.

might communicate through the way they draw their glyphs.

⁵ These must be symmetric to support war-gaming. The same is true of the timeline. The astute reader will notice that the red and blue units are slightly different; this is a doctrinal distinction made by the US military, whose generic Red side is based on a Soviet model.

The *Timeline* (Figure 5) enables temporal constraints to be stated between friendly tasks and between enemy tasks. Constraints that cross sides cannot be stated, since typically one is not privy to the other side's planned tasks⁶. One task can

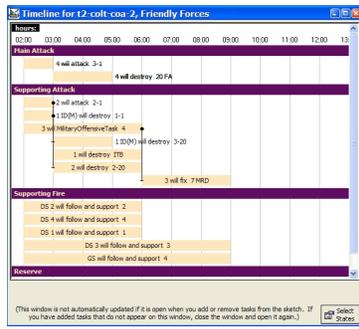


Figure 5

be constrained to start or end relative to the start or end of another task, or at some absolute time point. Estimates of durations can also be expressed.

SPATIAL REPRESENTATIONS AND REASONING

nSB is designed to be an interface to battlespace reasoning systems, built both by us and by others. Spatial reasoning is a crucial component in most battlespace reasoners [16]. Consequently, we incorporate a suite of visual computations in nSB that use sketched input to provide a combination of domain-specific and domain-independent qualitative spatial reasoning [10,17].

The nuSketch architecture currently uses two visual processors for spatial reasoning. The *ink processor* carries out basic operations when a glyph is created or updated. For example, the ink processor computes a bounding box, axes, and area of all glyphs when they are first created, and updates this information if the glyph is resized or rotated. Qualitative topological relationships (using the RCC8 vocabulary [3]) are automatically computed between a new glyph and every other glyph on its layer. The *vector processor* carries out more sophisticated spatial analyses, such as those involving



Figure 6

position-finding and path-finding. It maintains a set of *Voronoi diagrams* [6] for specific types of glyphs (e.g., terrain, terrain+friendly units, etc.) that are used in a variety of on-demand queries. Both processors are threaded, to take advantage of idle time and keep responsiveness high. Two “eyes” on the interface (Figure 6) let users know when spatial reasoning is occurring, and the number of events in the queue for each processor is also shown, as a form of progress indication.

Most of the spatial reasoning facilities are accessed on demand, by other reasoning systems⁷. All conclusions are justified through a logic-based truth maintenance system [12], to facilitate explanation generation and to retract conclusions appropriately when the diagram is updated. Most queries either confirm a relationship, when none of their arguments

⁶ This does limit our ability to express conditionals, e.g. “don’t fire until you see the whites of their eyes”.

⁷ The built-in knowledge inspector also has an ASK window for developers, but other reasoners provide their own interfaces for Q/A.

are (logical) variables, or find the set of entities that satisfy the relationships, when one of the arguments is a variable. The types of queries supported are:

Location: Asking whether an entity is at or inside a region, based on RCC8 relationships.

Positional: There are two kinds of positional relations. *Compass-based* positional relations are the standard northOf⁸, southOf, etc. There are two versions of compass positional relations, one based on centroids and the other which takes relative sizes of glyphs into consideration. The latter is closer to intuitive psychological judgments, but, like them, is not always defined for every pair of glyphs, e.g., if they overlap or one surrounds the other. Centroid-based relations are always defined, except in the extremely rare case where both centroids are identical. *Path-based* positional relations concern whether an entity is on a path (e.g., axis of advance, avenue of approach) and its relative position along the path (e.g., ahead or behind).

Preposition-like: These are close analogs to what are intuitively treated as spatial prepositions in natural languages, specifically near, adjacent, and between. We use Voronoi diagrams to compute these, based on [6]. How close these are to human intuitions is still an open question; it is known in other domains that function as well as geometry is important to accurately model human use of spatial prepositions [4,8]. So far these approximations have been reasonable.

Position-finding: Terrain analysis often involves identifying places that satisfy specific functional constraints. For example, a hiding place for an ambush must not be visible from any enemy vantage point (cf. Figure 7). We provide a query for

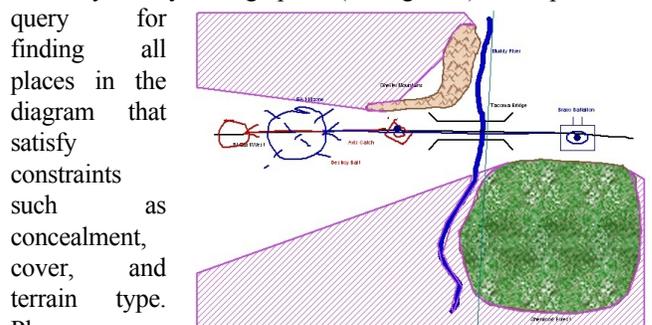


Figure 7

finding all places in the diagram that satisfy constraints such as concealment, cover, and terrain type. Places are constructed by polygon set operations over the sketch, using a simple domain theory about the cover, concealment, and trafficability properties of different terrain types for categories of units.

Path-finding: Many queries involve finding paths (e.g., which of these two units could reach Objective Slam sooner?). Following [5], we use a path-planner based on quad trees, using constraints specified as part of the query (i.e., speed, stealth) to construct a constraint diagram that

⁸ We use the DARPA subset of Cyc KB contents plus our own domain theories in our knowledge base.

divides the sketch into regions of constant cost, a dynamic query-specific qualitative representation.

COMIC GRAPHS

Plans are often complicated, involving sequences of states and conditionals. Military planning is often done under great uncertainty, making it necessary to take into account alternate hypotheses about what is happening and why. Both of these factors suggest that a sketching tool to support military planning should enable users to construct and relate descriptions of multiple states. nSB uses *comic graphs* for

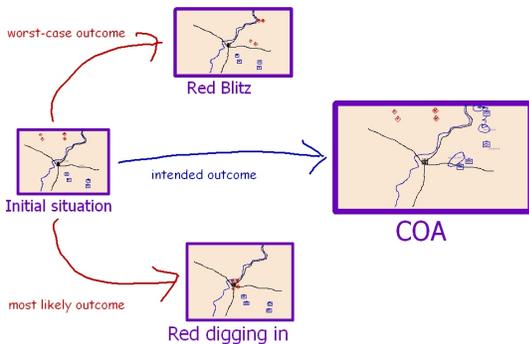


Figure 8

this purpose. A sketch can have multiple subsketches, each corresponding to a qualitatively distinct state of affairs (cf. Figure 8). If everything were certain, one could view an unfolding sequence of states almost like a comic strip, with each panel (subsketch) leading to the next. Unfortunately, our knowledge of the world, and of the future, are only partial. Thus we must introduce alternative states corresponding to different interpretations of observations, and different outcomes of events. We believe that this branching structure provides a valuable alternative to animation in visualizing complex plans and their outcomes, although experiments to prove this point are work for the future.

In qualitative physics, an exhaustive set of such states would be an *action-augmented envisionment* [10], but since comic graphs are user-generated we require neither completeness in the contents of a state nor for the set of states. While we expect future battlespace reasoners to produce comic graphs as one form of output, our users report it is already valuable in organizing their own thoughts. We plan to mine the corpus of sketches we are gathering for heuristics to guide development of automatic generation and presentation techniques.

Comic graphs are implemented using the *metallayer* mechanism introduced in sKEA [18]. That is, a sketch consists of a set of subsketches, each of which represents a particular state of affairs. Each subsketch appears as a glyph in a special layer, the metallayer. States can be given classifications from a small, intuitive collection (e.g., observed, hypothesized, intended, etc.). Relationships

between states are indicated by drawing arrows, labeled with the semantics of the relationship (e.g., hypothesis, intended next state, etc.). Comparison is an important operation on sketches [20]. States can also be compared to each other by analogy, via a drag and drop interface that invokes our analogy software [7,13] to compare them (Figure 9). These comparisons can be used to reflect on alternate choices, and work is in progress to hypothesize enemy intent based on historical precedents.

USER EXPERIMENTS AND FEEDBACK



Figure 9

nSB has benefited from substantial formative feedback from experts in three different venues. We discuss each in turn.

Integrated Course of Action Critiquing and Elaboration System Experiment: This experiment was conducted at the Battle Command Battle Laboratory at Ft. Leavenworth in FY 2000. An early version of nuSketch Battlespace (COA Creator) was combined with three other modules to create a crude prototype end-to-end system that started with a sketch and generated a synchronization matrix (a Gantt-chart style representation used by the military for detailed battle plans). The other modules were: (1) an Active-Templates style NL system to provide COA statement information, from AlphaTech, (2) a fusion system that combined COA Creator output with the statement information from Teknowledge, and (3) the CADET system from BBN to generate detailed plans and schedules from this output.

As reported in [25], this system enabled active-duty officers to generate COAs three to five times faster than by hand, with the same quality of plan produced. Four hours were needed to train officers to use this crude prototype⁹; it was estimated that with professional software integration the training time would be closer to one hour, due to the naturalness of the sketching system. This is an important datum, since the US Army's experience with digital media has generally been dismal¹⁰.

⁹ The officers had to be taught to transfer files from one component to another, and about situations that would lead to crashing, for example.

¹⁰ An anonymous opposition-force commander at the National Training Center claims that "Digital technology is a force multiplier ... for the enemy" <http://dtsn.darpa.mil/ixo/cpof%2Easp>

Greybeard usage in DARPA's Command Post of the Future Program. We have been fortunate to have a number of retired military officers testing our software, which has been a valuable long-term source of formative feedback. In our experience, we can have generals doing analogies between battlespace states within an hour of sitting down with the software for the first time.

Interface component in DARPA's Rapid Knowledge Formation Program. nSB was adopted by both RKF teams as part of the interfaces for their integrated systems. The purpose of these systems is to enable experts to extend and maintain knowledge bases with minimal intervention by AI experts. The KRAKEN system, by Cycorp and its collaborators, relies on natural language dialogue to interact with experts. The SHAKEN system, by SRI and its collaborators, relies on concept maps to interact with experts. Both teams are using nSB as part of their interface for this domain, enabling domain experts to combine sketching with their other modalities of communication. nSB provides a KQML server that enables external systems to access sketch information, perform spatial reasoning, and control nSB's interface to set up context for questions. Furthermore, the UMass group is also using output from nSB to set up situations for their Abstract Force Simulation system [21], which provides COA critiques using qualitative summaries compiled from Monte Carlo simulation. In this year's evaluation, supervised by an independent evaluation contractor during the first two weeks of October, retired military personnel used nSB to enter COAs as part of the knowledge capture process. Only 1-2 hours of training via teleconference was required for them to achieve reasonable fluency. The combined systems were successfully used by the experts to add and test new knowledge about COA critiquing.

All three of these experiences suggest that we have succeeded in making an interface that is natural for the intended user population. The ICCES experiment suggests that the representations we produce are useful for subsequent reasoning, and our experience in the RKF experiments, where two radically different AI systems successfully used it as an integral component, lends strong additional evidence in this regard.

DISCUSSION

We believe that nuSketch Battlespace provides a solid demonstration of the utility of the nuSketch approach to multimodal interfaces. By focusing on understanding rather than recognition, we have created an interface that users find natural and that enables them to work more efficiently. Clever interface design, backed by careful design of visual processing and reasoning, enables military users to carry out sophisticated analyses and generate plans. Much remains to be done, of course, but the basic mechanics of nuSketch Battlespace appear to be a stable platform for future development of more sophisticated battlespace reasoners and visualization systems.

We believe that these techniques can be applied to any domain-specific sketching system. Moreover, the major differences between this domain-specific system and our open-domain sketching system sKEA [14] are (1) the use of a domain-specific glyph bar instead of allowing arbitrary KB collections and (2) some domain-specific spatial reasoning. This suggests that one could have the best of both worlds, by combining domain-specific glyph bars for areas of frequent use, and rely on more general mechanisms for extensibility.

Most of our planned extensions concern embedding more sophisticated battlespace reasoning within nSB. First, we plan on using our MAC/FAC model of similarity-based retrieval [15] as part of an enemy intent recognition system. We believe that such a system could be a valuable adjunct in war-gaming, given the medium of comic graphs for communicating its results. Second, we are exploring collaborations with the US military to use a version of nSB for training, extended with built-in coaching and critiquing functionality. This will enable us to greatly extend our case library, which will raise some interesting interface issues for browsing and maintenance of a semantically rich graphical case library. Finally, we are exploring the use of nSB as an interface for computer wargames, where players would issue commands by assigning tasks to (computer controlled) subordinates. This brings up interesting issues concerning real-time operation and updates, as well as the potential to significantly increase the number of users of multimodal interfaces.

ACKNOWLEDGMENTS

This research was supported by the DARPA Command Post of the Future and Rapid Knowledge Formation programs. We thank Thomas Hinrichs and our military colleagues for valuable feedback.

REFERENCES

1. Alvarado, Christine and Davis, Randall (2001). Resolving ambiguities to create a natural sketch based interface. *Proceedings of IJCAI-2001*, August 2001.
2. Cohen, P. R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. (1997). QuickSet: Multimodal interaction for distributed applications, *Proceedings of the Fifth Annual International Multimodal Conference (Multimedia '97)*, (Seattle, WA, November 1997), ACM Press, pp 31-40.
3. Cohn, A. (1996) Calculi for Qualitative Spatial Reasoning. In *Artificial Intelligence and Symbolic Mathematical Computation*, LNCS 1138, eds: J Calmet, J A Campbell, J Pfalzgraf, Springer Verlag, 124-143, 1996.
4. Coventry, K. 1998. Spatial prepositions, functional relations, and lexical specification. In Olivier, P. and Gapp, K.P. (Eds) 1998. *Representation and Processing of Spatial Expressions*. LEA Press.
5. Davis, I. 2000. Warp Speed: Path Planning for Star Trek: Armada, *AI and Interactive Entertainment: Papers*

- from the 2002 AAAI Spring Symp., AAAI Press, Menlo Park, Calif.
6. Edwards, G. and Moulin, B. 1998. Toward the simulation of spatial mental images using the Voronoi model. In Olivier, P. and Gapp, K.P. (Eds) 1998. *Representation and Processing of Spatial Expressions*. LEA Press.
 7. Falkenhainer, B., Forbus, K., Gentner, D. (1989) The Structure-Mapping Engine: Algorithm and examples. *Artificial Intelligence*, 41, pp 1-63.
 8. Feist, M. and Gentner, D. 1998. On Plates, Bowls and Dishes: Factors in the Use of English IN and ON. *Proceedings of the 20th annual meeting of the Cognitive Science Society*
 9. Ferguson, R.W., Rasch, R.A., Turmel, W., & Forbus, K.D. (2000) Qualitative Spatial Interpretation of Course-of-Action Diagrams. *Proceedings of the 14th International Workshop on Qualitative Reasoning*. Morelia, Mexico. June, 2000.
 10. Forbus, K. 1989. Introducing actions into qualitative simulation", Proceedings of IJCAI-89, August.
 11. Forbus, K. 1995. Qualitative Spatial Reasoning: Framework and Frontiers. In Glasgow, J., Narayanan, N., and Chandrasekaran, B. *Diagrammatic Reasoning: Cognitive and Computational Perspectives*. MIT Press, pp. 183-202.
 12. Forbus, K., and de Kleer, J. 1993. *Building Problem Solvers*, MIT Press.
 13. Forbus, K., Ferguson, R. and Gentner, D. (1994) Incremental structure-mapping. *Proceedings of the Cognitive Science Society*, August.
 14. Forbus, K., Ferguson, R. and Usher, J. 2001. Towards a computational model of sketching. *IUI'01*, January 14-17, 2001, Santa Fe, New Mexico
 15. Forbus, K., Gentner, D. and Law, K. 1995. MAC/FAC: A model of Similarity-based Retrieval. *Cognitive Science*, 19(2), April-June, pp 141-205.
 16. Forbus, K., Mahoney, J.V., and Dill, K. 2001. How qualitative spatial reasoning can improve strategy game AIs: A preliminary report. 15th International workshop on Qualitative Reasoning (QR01), San Antonio, Texas, May.
 17. Forbus, K., Nielsen, P. and Faltings, B. "Qualitative Spatial Reasoning: The CLOCK Project", *Artificial Intelligence*, 51 (1-3), October, 1991.
 18. Forbus, K. and Usher, J. 2002. Sketching for knowledge capture: A progress report. *IUI'02*, January 13-16, 2002, San Francisco, California.
 19. Gross, M. (1996) The Electronic Cocktail Napkin - computer support for working with diagrams. *Design Studies*. 17(1), 53-70.
 20. Gross, M. and Do, E. (1995) Drawing Analogies - Supporting Creative Architectural Design with Visual References. in *3d International Conference on Computational Models of Creative Design*, M-L Maher and J. Gero (eds), Sydney: University of Sydney, 37-58.
 21. King, Gary W., Brent Heeringa, Joe Catalano, David L. Westbrook, and Paul Cohen. Models of Defeat. *Proceedings of the Second International Conference on Knowledge Systems for Colation Operations 2002*. pp. 85-90.
 22. Landay, J. and Myers, B. 1996. Sketching storyboards to illustrate interface behaviors. CHI'96 Conference Companion: Human Factors in Computing Systems, Vancouver, Canada.
 23. McGee, D. R., Cohen, P. R. (2001) "Creating tangible interfaces by augmenting physical objects with multimodal language," in the Proceedings of the International Conference on Intelligent User Interfaces (IUI 2001), ACM Press: Santa Fe, NM, Jan. 14-17, pp. 113-119.
 24. McGee, D.R., Cohen, P.R., Wesson, M., and Horman, S. 2002. Comparing paper and tangible multimodal tools. *Proceedings of the Conference on Human Factors in Computing Systems (CHI'02)*, ACM Press, Minneapolis, MI, April 20-25 2002
 25. Rasch, R., Kott, Al, and Forbus, K. 2002. AI on the Battlefield: An experimental exploration. *Proceedings of the 14th Innovative Applications of Artificial Intelligence Conference*, July, Edmonton, Canada.
 26. Stahovich, T. F., Davis, R., and Shrobe, H., "Generating Multiple New Designs from a Sketch," in Proceedings Thirteenth National Conference on Artificial Intelligence, AAAI-96, pp. 1022-29, 1996.
 27. Waibel, A., Suhm, B., Vo, M. and Yang, J. 1996. Multimodal interfaces for multimedia information agents. *Proc. of ICASSP 97*