Boundary-based multimodal input for geographic planning sketches

Kenneth D. Forbus (forbus@ils.nwu.edu)

Ronald W. Ferguson (ferguson@ils.nwu.edu)

Jeffrey M. Usher (usher@ils.nwu.edu)

Qualitative Reasoning Group
Department of Computer Science
Northwestern University
Evanston, IL USA 60201

Abstract

Interactive sketching is a powerful means of interpersonal communication. Current computer-based sketching systems, while falling short of human-like interaction, successfully exploit the limited gesture and speech recognition technology that is currently available. These sketching systems succeed through calculated bets about the nature of human gesture, language, and spatial interpretation. In this paper, we evaluate such tradeoffs in our own interactive sketching system, nuSketch, which is currently under development. nuSketch uses a technique called *boundary-based multimodal sketching* to build representations of a domain. This technique is well suited to nuSketch's current task: a geographic military planning domain called *course-of-action* (COA) sketches. As part of our analysis of nuSketch, we identify three dimensions of sketching — *visual understanding, language understanding, and conceptual understanding* — that are useful for characterizing the abilities of sketching systems and identifying open problems in this area.

1. Introduction

When interactive communication is important and time-critical, people seldom limit themselves to a single modality. On shared surfaces, such as whiteboards, tablets, and paper, people draw a variety of sketches, charts, and other spatially-rich depictions. These depictions are not static, but are frequently annotated, discussed, and redrawn. Along with drawing, people also point, mark, highlight, and underscore items, using these and other gestures to help disambiguate what they are attempting to convey. Finally, language is used to tie together loose ends, identify objects when drawing skills are insufficient, or provide conceptual information. In this paper, we refer to this communication process as *multimodal sketching*. In multimodal sketching, the drawing carries the spatial aspects of what is to be communicated, while language provides a complementary conceptual channel that guides sketch interpretation.

Our ability to communicate using multimodal sketching is increasingly understood to be a critical skill that we need to embed in software. As shown by looking at a few such systems (Allen et al., 1995; Cohen, 1992; Cohen et al., 1997; Oviatt, 1999), multi-modal sketching is especially crucial in spatial domains. Interfaces for multimodal sketching now form a substantial body of research (Maybury & Whalster, 1998), and build upon existing work on intelligent sketching systems (Hearst, Gross, Landay, & Stahovich, 1998; Landay & Myers, 1995; Pedersen, McCall, Moran, & Halasz, 1993; Saund, 1995).

This paper describes our current progress in developing nuSketch, an architecture for multimodal sketching, and places nuSketch's design within the space of other interactive sketching systems. We begin by introducing nuSketch's current task: a geographic military planning domain called *course-of-action (COA) sketches*. We then discuss three *dimensions* *of multimodal sketching*, and discuss the COA sketch domain in terms of these dimensions. Then, through an input technique called *boundary-based multimodal sketching*, we show how nuSketch allows users to build COA sketches, and motivate the technique in terms of the previously-defined sketching dimensions. We end by discussing nuSketch's architecture.

2. Course-of-Action (COA) Sketches

COA sketches are military planning sketches, designed to relate a set of military units and tasks to a geographical region. A COA sketch, given a rough map of a region's geographic features, depicts that region's military units and shows the units' movements and tasks. The sketch is accompanied by a written description of the intent and plan (Note: collectively, this text and the sketch constitute the whole "Course of Action"—we use the term "COA sketch" to refer to the sketch alone).

COA sketches provide an ideal testing ground for research into multimodal sketching. COA sketches involve both an inherently spatial task and a broad and extensible visual symbology. Military planners use COA sketches in a highly interactive fashion. COA sketches can express the gist of a plan, before many details—such as timing—have been worked out. Traditionally, such sketches are drawn on acetate overlays over maps, or on paper starting with hand-drawn abstractions of critical terrain features. They are frequently redrawn, either to adapt the COA plan, or to make the plan more or less abstract (e.g., to take a division-level plan and adapt it for battalion-level plans).

An example COA sketch, drawn using nuSketch, is shown in Figure 1. Within this sketch, we can look at a few glyph instances to demonstrate the expressiveness of the symbology. The sketch begins with a number of geographic features, including towns, rivers and bridges (as shown by the labels in Figure 1). Military units are shown using a composable set of visual glyph parts that indicate type, echelon level, and specialization (friendly units are rectangles, enemy units are diamonds, units with two "antennae" are battalions, and so forth). Thick boundary lines, marked with X's or hash marks (which indicate their echelon), divide the entire space into *areas of operations*, which are areas of

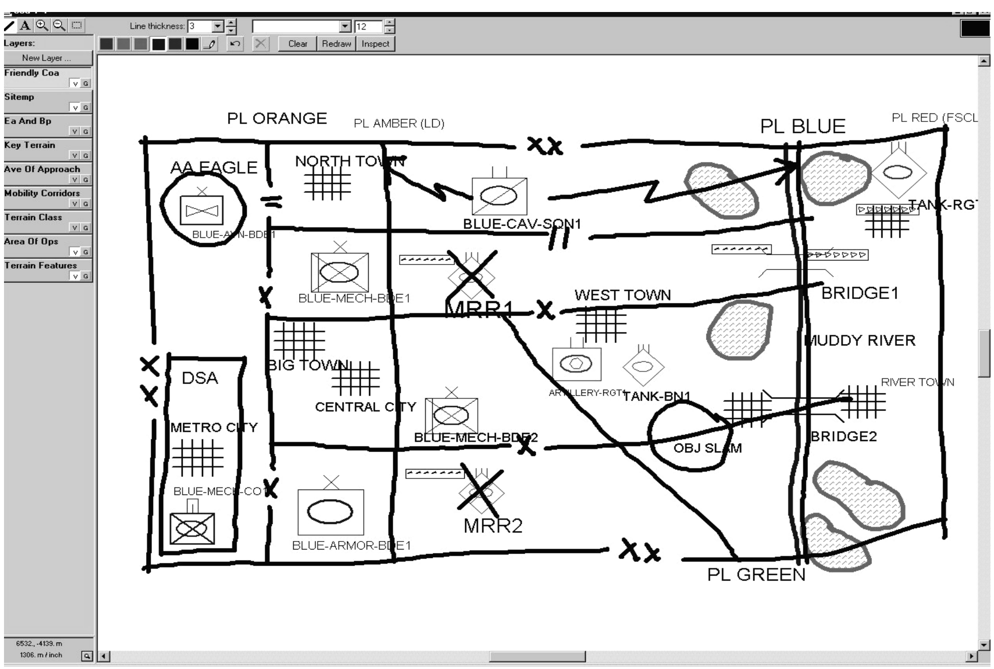

Figure 1: A Course-of-Action sketch drawn using nuSketch

responsibility to which friendly units are assigned. Units are also assigned specific tasks, such as attacks, via drawn glyphs. For example, at the top of the sketch, Blue Calvary Squadron 1 (BLUE-CAV-SQN1) has the responsibility of defending its area of operations, as indicated by the lightning-like arrows leading from it to the east and west.

This symbology is easily drawn. However, it is not clear what level of gesture, speech and glyph recognition is needed for this domain. To consider this, we take a detour to cover the three dimensions of multimodal sketching, which will help us with this analysis.

3. Dimensions of multimodal sketching

To design computer systems that handle multimodal sketching, we must consider what sketch-understanding skills the domain requires. These skills vary depending on the domain and the available computational resources. For this reason, we need a way to characterize the ways in which these skills vary. For this purpose, we characterize these skills along three dimensions: *visual understanding, language understanding, and conceptual understanding*. These dimensions are listed in Figure 2, along with example points along each dimension. We cover each dimension in turn.

Visual understanding. This dimension characterizes how deeply the spatial properties of the ink are understood. The easiest level of visual understanding is recognizing gestures that indicate location (i.e., pointing), which are often used to indicate actions to take with respect to something at that location (e.g., selecting or deleting). Systems can also have users indicate the size of glyphs, using techniques such as selecting an area. Systems that work at this level fall just short of true sketching, since these visual properties do not make use of the kinds of spatial information normally associated with sketching.

The first level of a true sketching system lies beyond just size and location, through systems that allow the user to draw boundaries. Boundary-based objects, such as regional borders, rivers, or highways, allow for more sophistical spatial relations in the sketch.

The next level of visual understanding is the use of a visual symbology (i.e., a collection of glyphs), whose forms represent conceptual elements of a particular domain. Usually the spatial properties of such glyphs also have conceptual meaning. Schematic diagrams in various technical fields and formal visual languages such as COA sketches heavily utilize this kind of symbology. In understanding such a symbology, we make a distinction between shape classification of glyphs (assigning a conceptual type to a particular glyph, such as identifying a particular glyph as a tank), shape characterization (identifying particular characteristics of the represented object based on the shape), and using shape composability (understanding how visual subparts of a particular glyph relate to one another, such as how an echelon marker relates to a military border).

Dimension	**Skill levels along this dimension (from least to most difficult)**
Visual understanding	Glyph location → glyph size → glyph boundary → shape classification → shape characterization → shape composability.
Conceptual understanding	Glyph classification → spatial relations between glyphs → conceptual interpretation.
Language	Glyph naming → template or FSA-based commands → parsed sentences → mixed-initiative dialogs.

Figure 2: Dimensions of multimodal sketching

Language understanding. Just as sketching systems may vary in their visual understanding, systems may have varying levels of understanding of speech and language input. Language provides several services during sketching. It eases vision's task by labeling entities, specifying the type of thing being drawn, or by stating important spatial relationships (indicating when spatial relationships are important as opposed to accidental). It can indicate things that are not in the sketch, or which cannot be drawn. Speech is the most common alternate modality used during sketching because it allows visual attention to remain on the sketch, although short written labels are also used.

Existing multimodal sketching systems typically use off-the-shelf speech recognition systems. These systems are thus limited to finite-state or definite clause grammars (c.f. (Cohen, 1992; Cohen et al., 1989; Maybury & Whalster, 1998)). However, given the differences in complexity between spoken and written text, such grammars are likely to remain sufficient (Allen et al., 1995).

The most important dimension for characterizing language understanding in sketching systems concerns dialogue management (Grosz & Sidner, 1986; Luperfoy, in preparation). Most systems have been command-oriented, with some support for system-initiated clarification questions. We know of no sketching systems that use full mixed-initiative dialogs. We suspect two reasons for this. First, when multimodal interfaces are grafted onto legacy software, the existing output presentation systems are often used. Second, the relatively shallow conceptual understanding used in most systems does not support them doing much on their own, so they are unlikely to need to interject anything.

Conceptual understanding. As a communicative act, sketching requires common ground (Clark, 1996). Thus the depth of representation of what is sketched is probably the single strongest factor determining how flexible communication can be. There must be enough visual and language understanding (and these can be traded off against each other), but it is the degree of shared conceptual knowledge that ultimately limits what can be communicated, no matter what modalities are available. As might be expected, this is the weakest area for current systems.

The simplest level of conceptual understanding for sketching is the ability to handle a fixed collection of types of entities and relationships (Cohen, 1992; Cohen et al., 1989; Gross, 1994; Waibel, Suhm, Vo, & Yang, 1996). It is also the level most commonly used, since it suffices to issue commands to other software systems, the primary purpose of most existing multimodal interfaces. Type information is often used to reduce ambiguity, e.g., if a gesture indicating the argument to a MOVE command might be referring to a tank or a fence, the latter is ruled out.

Moving beyond identifying an intended command and its arguments requires broader and deeper common ground. Domain-specific systems obviously need knowledge about their domain. But there are areas of knowledge that cut across multiple domains of discourse that seem to be necessary to achieve flexible communication via sketching. *Qualitative representations of space* are important in almost every sketching domain. These representations include representations of regions, paths, and relative locations (Cohn, Randell, Cui, & Benett, 1993; Forbus, 1995). *Qualitative representations of shape* provide the ability to abstract away minor differences in order to describe important properties, which facilitates recognition (Egenhofer, 1997).

We claim that qualitative spatial representations are crucial for several reasons. First, they are well-suited for handling the sorts of approximate spatial descriptions provided by hand-drawn figures, layouts, and maps. Second, the level of description they provide is close to the descriptions of continuous properties common in human discourse (Forbus, Nielsen, & Faltings, 1991; Stahovich, Davis, & Shrobe, 1996). The nuSketch COA system, for instance, relies on qualitative representations to understand geographic questions and as part of the encoding of a situation that facilitates retrieval for generating critiques via analogy.

4. Requirements for understanding Course-of-Action sketches

We can use these dimensions of multimodal interaction to help us design a system to understand COA sketches. Here, we consider the kinds of drawing and shape-recognition skills that go into particular classes of glyphs. For our purposes, it is helpful to divide the symbols into three types, depending on the drawing characteristics of the glyph:

Composable shape-based glyphs. These glyphs include symbols for military units, minefields, towns, bridges, and a few other items. For these symbols, the glyph has a specific form that is composed of parts that depict particular characteristics (friendly versus enemy unit, armor type, echelon, and some specializations, such as medical units). These symbols can be identified by their visual form alone, and can also be described using succinct noun phases ("friendly armor battalion"). In most cases, the location of the glyph is the most important characteristic, and the boundary does not represent any useful characteristics of the represented object.

Boundary-based glyphs. These glyphs include military and geographic borders, geographic regions (such as mountain regions or key terrain), and some geographic features such as rivers. In many cases, these glyphs are difficult to recognize without additional information. For example, in COA sketches there are no natural visual differences between the drawing of a river and the drawing of a road. Of course, in lieu of form-based identification, these glyphs can be labeled or named using speech input. However, they require more than a system based on pointing can provide. In addition, while in shape-based glyphs the figure position alone is critical, for boundary-based glyphs the boundary is critical to understanding the intent of the glyph. For example, the “severely restricted terrain” glyph indicates a bounded area that cannot be traversed by a particular class of units, and so characteristics of that boundary (e.g., concavities) can be important to understanding the nature of any related tasks.

Mixed-level glyphs. These glyphs combine aspects of the two previous types, including task glyphs and movement arrows. In these cases, the type of the glyph (i.e., arrow or task shape) indicates one of many task types. However, the spatial characteristics also include elements that are more free-form. A good example is the SCREEN task seen at the top of Figure 1. In this task, the arrow shape (with lightning-like bends) indicates the task type. However, the dimensions of the arrows also show the area being defended.

Drawing in nuSketch attempts to provide a way to handle shape-based, boundary-based, and mixed-level glyphs without interrupting the flow of the user's sketching. At present, it does so without doing glyph recognition. This imposes special constraints on the sketching system. We overcome these constraints, at present, using an input technique called boundary-based multimodal sketching. This technique uses command-level language understanding to compensate for visual understanding that only reaches the level of glyph boundaries.

5. Boundary-based multimodal sketching in nuSketch

The COA symbology is immense. Due to the composability of subtypes (such as parts of a military task force), it contains hundreds of visual forms for military units alone. The size and composability of the symbology makes a menu- or palette-based approach to sketch input impractical. At present, it also appears to block one promising approach taken by other multimodal interfaces, which rely on black-box glyph recognition algorithms (e.g., hidden Markov models or neural nets) which operating on digital ink (Cohen et al., 1997; Gross, 1996). Such an approach requires a large set of examples for each recognized form, and the broad composability of COA glyphs makes that extremely difficult. While a critical subset of the COA symbology is recognized by some systems, most notably QuickSet (Cohen et al., 1997), the sheer variability in the symbology leaves open the possibility that current symbol-recognition technologies will not scale. Because we felt that this form of glyph-recognition was impractical in the short-term, we have taken a different approach.

In general, our approach uses speech recognition to do glyph classification, allowing for a level of visual understanding that is limited to boundary identification. The key idea is that recognition of composable naming conventions in language is a more scalable process than the recognition of composable sketched parts in diagrams. This simplification through language often aids, rather than impedes, the flow of sketching. After all, most people cannot produce drawings of complex objects and relationships that are visually recognizable without breaking the flow of conversation. However, the verbal description that occurs during drawing, punctuated by written labels, does not block flow, and compensates for inaccuracies in drawing. Inaccuracies that remain can be handled by follow-up questions.

The approach we have chosen is called *boundary-based multimodal sketching*. There are three factors that can be used to determine when boundary-based multimodal sketching is a useful technique: (1) Object recognition is difficult, due to limitations in processing power or the breadth of the symbology. (2) Task demands do not allow a palette-based approach. (3) Object boundaries are free-form, and thus boundaries must be sketched.

In boundary-based multimodal sketching, speech recognition is used to indicate the type of object to be added, and the user then either draws the object directly, or indicates the object boundary.

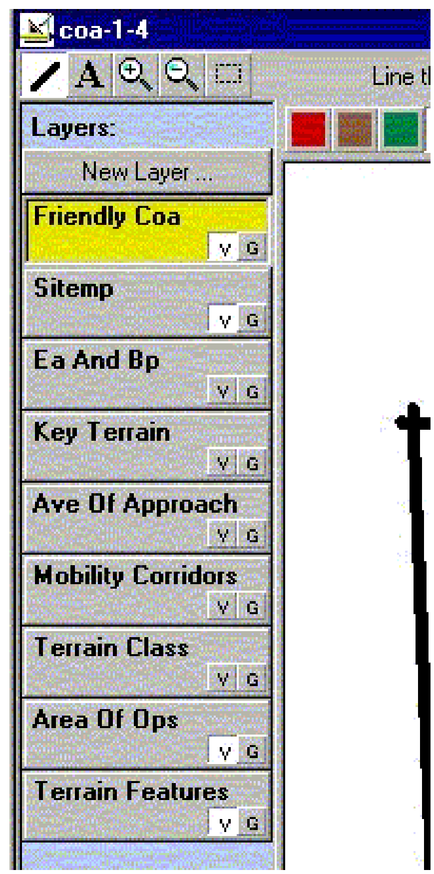

Figure 3: Detail from upper left portion of Figure 1. nuSketch uses a *layers* metaphor to organize drawing layers in the COA domain. The selected layer determines the domain theory used for speech and gesture recognition. Existing layers can be drawn in a given color for emphasis, omitted, or grayed-out as needed to make the existing sketch understandable.

5.1. Layer interface

A *layer* metaphor organizes the interface

(Figure 3). Like acetate layers in real COA sketches, each nuSketch layer corresponds to some category of domain information, such as terrain analysis, enemy disposition, disposition of friendly units, and so forth. Switching between layers is accomplished by clicking on the tabs to the left. Multiple layers can be displayed at once, or hidden or grayed out as convenient.

The choice of active layer determines how user inputs are interpreted. For example, if the *terrain* layer is active, the user can add regions corresponding to different terrain categories (i.e., regions where movement is restricted due to slope, soil type, or vegetation) and man-made features such as cities and towns. The layers thus inform both the user (by indicating only layers of interest) and speech-recognition (by indicating the domain).

5.2. Sketching input

Additions are made via speech command (e.g., “Add severely restricted terrain”) accompanied by a gesture, whose interpretation depends on the command. For adding regions, the curve drawn is taken to be the boundary of the region, so it is closed and filled with the appropriate texture to indicate that the command was understood. For adding standard symbols, such as towns, the user’s gesture indicates a bounding box, and the appropriate glyph is retrieved from the KB, scaled appropriately, and then displayed.

We can describe each type of input using the three glyph categories described above. Examples of each type of input are given in Figures 4, 5, and 6.

For composable shape-based glyphs (Figure 4), the user speaks the object type, and then draws the object bounding box. nuSketch then retrieves a glyph prototype, resizes it to fit the

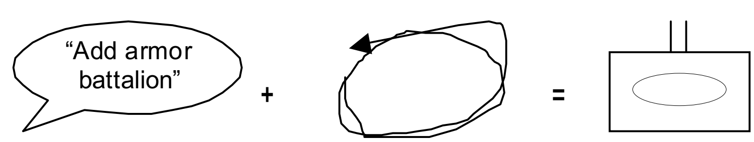

Figure 4: Adding a composable shape-based glyph in nuSketch.

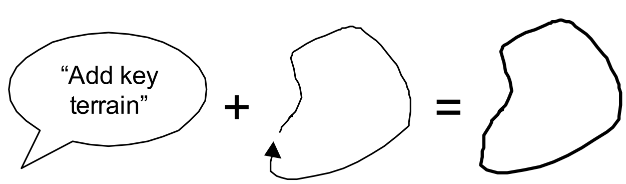

Figure 5: Adding a boundary-based glyph in nuSketch

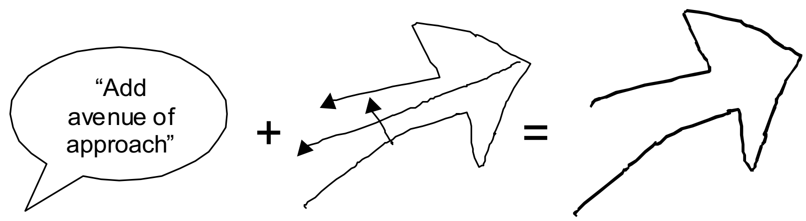

Figure 6: Adding a mixed-level glyph in nuSketch

boundary, and places the new glyph into the sketch. This action also serves as feedback to the user indicating that their command was understood.

For boundary-based glyphs (Figure 5), the user speaks the glyph type ("Add key terrain…") and then draws the object boundary in a single stroke, with no limitations on the nature of that stroke. In the case of closed boundary glyphs, the drawn boundary is then closed by nuSketch and filled with a texture appropriate to the glyph type. Some boundary types, such as military borders, allow multiple strokes, so the user can add additional information such as echelon indicators.

Mixed-level glyphs are the most difficult type of glyph to draw, requiring the greatest amount of learning on the part of the user. These glyphs combine the prior forms with sets of *selection* and *ending* strokes, which select items to which the glyph relates, and also establish spatial properties of the glyph that are difficult to determine from the boundary alone.

In the case of drawing avenues of approach (Figure 6), a mixed-level approach is needed to have the system represent the arrow itself, and to allow other aspects of the arrow (such as breadth and axis) to be determined without object recognition. In this case, the user first draws the arrow (Figure 6), and then performs two ending strokes that give the axis and width of the arrow, respectively.

Although the current system has a single stroke sequence for each glyph type, it would be possible to use more than one stroke sequence, allowing the user to select the most intuitive sequence to draw a particular glyph.

In the COA domain, boundary-based multimodal sketching balances a lower level of visual understanding with particular assumptions concerning language use. To handle the interaction just described requires an engine that flexibly interleaves speech and ink inputs with knowledge of the domain and its symbology. We examine this architecture next.

5.3. Architecture

While we have focused on the COA sketch domain in this paper, nuSketch is designed as a general-purpose multimodal architecture for sketching. NuSketch's architecture (Figure 7) combines speech and ink inputs with user-definable domain knowledge.

The Ink Processor accepts pen input, does simple signal processing, and passes time-stamped data to the Multimodal Parser. The other input to the multimodal parser is from a commercial speech recognizer, which produces time-stamped text strings.

The Multimodal Parser uses grammars that include both linguistic and gesture information, to produce propositions that are interpreted by the Dialog Manager. The Dialog Manager and the KB contents are the only application-specific components of nuSketch.

The Dialog Manager is responsible for interpreting propositions and supplying grammars to the speech recognizer and Multimodal Parser based on context (as determined by its own state and the active layer).

Central to nuSketch is the use of a knowledge-based reasoner, the domain-theory environment (DTE; (Mostek, in preparation)), which provides integrated access to a number of reasoning services, including analogical reasoning and geographic reasoning. The Dialog Manager uses DTE for its reasoning, and as much domain-specific knowledge is stored in the KB as possible. For example, the glyphs corresponding to the visual symbols in a domain are stored as part of the knowledge base, so that the manner in which something is depicted can be reasoned about (e.g., if a glyph is not available for a specific unit type, use a glyph corresponding to a more general type of unit).

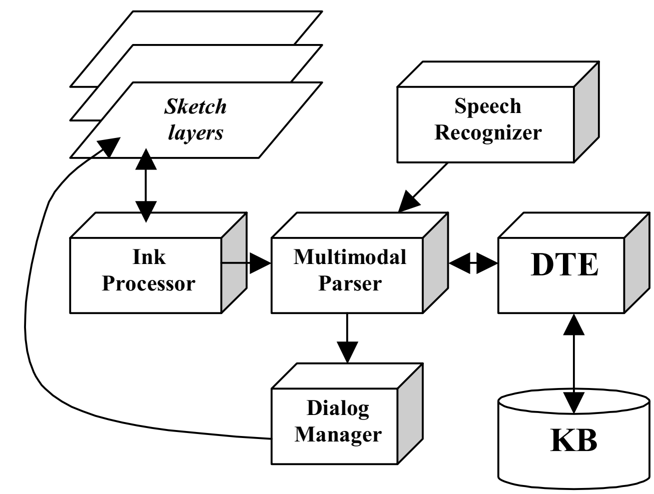

Figure 7: nuSketch architecture

Several aspects of nuSketch are inspired by Quickset (Cohen et al., 1997), a multimodal interface system for setting up military simulations. nuSketch's approach also, like QuickSet's, uses off-the-shelf speech recognition and time-stamping ink and speech signals to facilitate integrating information across modalities. However, Quickset incorporates ink-recognition schemes that nuSketch does not. On the other hand, because Quickset was designed as a means of generating commands for a legacy computer system, it lacks an integrated reasoning system.

The approach in nuSketch is also similar to that used in other gesture-driven sketching systems that do not handle speech input, such as SILK (Landay & Myers, 1995) and Tivoli (Pedersen et al., 1993). These systems also incorporate a flexible but partially-fixed vocabulary of gestures to handle actions such as glyph editing. While these systems do not handle speech input, they handle a broader range of primitive gestures (including squiggles and "pigtails"). The primitive shape recognizers of these systems, at some computational cost, could be usefully incorporated into nuSketch to increase its glyph-recognition ability or to make editing more flexible.

6. Discussion

In this paper, we have described our system, nuSketch, and the tradeoffs it makes between visual, language, and conceptual sketch understanding when drawing COA diagrams. Although nuSketch's visual and language understanding capabilities are minimal, the sketching system is powerful, and has been used to draw several dozen COA diagrams, which have been used in other reasoning systems, such as geographic reasoners (Ferguson, Rasch, Turmel, & Forbus, 2000)[1]. Although nuSketch is still under development, the COA application in nuSketch is currently being evaluated by military personnel at the Battle Command Battle Labs in Fort Leavenworth, Kansas. Initial informal feedback indicates that users appreciate the ability to name a unit type rather than drawing it directly, since some unit types (such as armor units) involve six or more strokes. Such glyphs are more easily named than drawn. Users have also indicated, however, that speech cannot be used in many situations, and so a system that utilizes sketching without speech is clearly needed.

To that end, we are currently looking at ways to expand nuSketch's visual ability, including the incorporation of computer-based methods of perceptual organization, such as (Saund, 1995). As mentioned earlier, several other systems incorporate gesture recognition algorithms that could be used in the existing architecture as an additional method of communication. In addition, nuSketch's multimodal parser can be made more flexible, allowing ambiguous glyph interpretations to be decided based on evidence from multiple modalities (Cohen et al., 1997), or to allow multiple interpretations to be used until a single interpretation is needed (Landay & Myers, 1995).

Our current plan is to expand nuSketch's ability to recognize and categorize individual glyphs based on their qualitative shape characteristics. To make nuSketch more sensitive to qualitative visual structure, we will integrate its recognition routines with a qualitative spatial reasoner, GeoRep (Ferguson & Forbus, 2000). Given the ability to break down the qualitative spatial characteristics of a glyph, we can approach recognition in three stages: (1) validation of glyph classification done by speech-interpretation, and (2) shape-based glyph identification with ambiguous speech input (choosing between multiple interpretations), and (3) identification without speech input. For these stages of development, boundary-based multimodal sketching will serve as a sort of staging area, allowing nuSketch to remain a productive tool for users while we continue to enhance nuSketch's recognition abilities.

Sketching is a powerful human-to-human communication activity, and consequently it provides a potentially powerful metaphor for human-computer interaction. Research on sketching provides an arena for investigating the intersection of conceptual knowledge, visual understanding, and language, making it a valuable area for investigation in order to understand human cognition. This progression suggests that to achieve the kind of flexible interaction that sketching provides in human-to-human communication, multimodal research will rely heavily upon, and even drive, new research in the artificial intelligence field.

7. Acknowledgements

This research was supported by DARPA under the High Performance Knowledge Bases and Command Post of the Future programs. We wish to thank George Lee, Tom Mostek, Bill Turmel and Rob Rasch for programming support and helpful discussions. We also wish to

[1] To get an idea of the relative complexity of the sketches involved, in the 22 sketches in the alpha version of our Tactical Decision Coach, there are around 50 glyphs on average (minimum of 25, maximum of 75).

thank Brian Dennis, Eric Saund, and two anonymous reviewers for thoughtful and useful feedback.

8. References

Allen, J. F., Schubert, L. K., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., Light, M., Martin, N. G., Miller, B. W., Poesio, M., & Traum, D. R. (1995). The TRAINS project: A case study in defining a conversational planning agent. *Journal of Experimental and Theoretical Artificial Intelligence, 7*, 7-48.

Clark, H. (1996). *Using Language*: Cambridge University Press.

Cohen, P. (1992). *The role of natural language in a multimodal interface* (UIST92).

Cohen, P., Dalrymple, M., Moran, D., Pereira, F., Sullivan, J., Gargan, R., Schlossberg, J., & Tyler, S. (1989). Synergistic use of direct manipulation and natural language, *Proceedings of CHI-89* (pp. 227-233).

Cohen, P., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., & Clow, J. (1997). QuickSet: Multimodal interaction for distributed applications, *Proceedings of the Fifth Annual International Multimodal Conference* (pp. 31-40). Seattle.

Cohn, A. G., Randell, D. A., Cui, Z., & Benett, B. (1993). Qualitative spatial reasoning and representation. In N. P. Carrete & M. Singh. (Eds.), *Proc of the IMACS Workshop on Qualitative Reasoning and Decision*

Technologies, QUARDET '93 (pp. 513-522). Barcelona: CIMNE.

Egenhofer, M. (1997). Query processing in spatial-query-by-sketch. *Journal of Visual Languages and Computing, 8*(4), 403-424.

Ferguson, R. W., & Forbus, K. D. (2000). GeoRep: A flexible tool for spatial representation of line drawings, *Proceedings of the 18th National Conference on Artificial Intelligence* . Austin, Texas: AAAI Press.

Ferguson, R. W., Rasch, R. A. J., 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.

Forbus, K. D. (1995). Qualitative spatial reasoning: Framework and frontiers. In J. Glasgow, N. H. Narayanan, & B. Chandrasekaran (Eds.), *Diagrammatic Reasoning: Cognitive and Computational Perspectives* (pp. 183-202). Menlo Park, CA: The AAAI Press/The MIT Press.

Forbus, K. D., Nielsen, P., & Faltings, B. (1991). Qualitative spatial reasoning: The CLOCK project. *Artificial Intelligence, 51*(1-3).

Gross, M. (1994). Recognizing and interpreting diagrams in design. In T. Catarci, M. Costabile, S. Levialdi, & G. Santucci (Eds.), *Advanced Visual Interfaces '94* (pp. 89-94): ACM Press.

Gross, M. D. (1996). The Electronic Cocktail Napkin: A computational environment for working with design diagrams. *Design Studies, 17*(1), 53-69.

Grosz, B. J., & Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics, 12*(3).

Hearst, M., Gross, M. D., Landay, J. A., & Stahovich, T. F. (1998). Trends and controversies: Sketching intelligent systems. *IEEE Intelligent Systems, 13*(3), 10-19.

Landay, J. A., & Myers, B. A. (1995). Interactive Sketching for the Early Stages of User Interface Design, *Proceedings of CHI '95: Human Factors in Computing Systems* (pp. 43-50). Denver, CO.

Luperfoy, S. (Ed.). (in preparation). *Automated Spoken Dialogue Systems*. Cambridge, MA: MIT Press.

Maybury, M., & Whalster. (1998). *Readings in Intelligent User Interfaces*: Morgan Kaufmann.

Mostek, T. (in preparation). The Domain Theory Environment .

Oviatt, S. (1999). Ten myths of multimodal interaction. *Commun. ACM, 42*(11), 74 - 81.

Pedersen, E. R., McCall, K., Moran, T. P., & Halasz, F. G. (1993). Tivoli: An electronic whiteboard for informal workgroup meetings, *Proceedings of the InterCHI93 Conference on Human Factors in Computer Systems* (pp. 391-398). New York: ACM.

Saund, E. (1995). *Perceptual organization in an interactive sketch editing application.* Paper presented at the Fifth International Conference on Computer Vision.

Stahovich, T. F., Davis, R., & Shrobe, H. (1996). Generating multiple new designs from a sketch, *Proceedings of the Thirteenth National Conference on Artificial Intelligence* (pp. 1022-1029).

Waibel, A., Suhm, B., Vo, M., & Yang, J. (1996). Multimodal interfaces for multimedia information agents, *Proceedings of ICASSP 97* .