

Sketching for knowledge capture: A progress report

Kenneth D. Forbus

Qualitative Reasoning Group
Northwestern University
1890 Maple Avenue
Evanston, IL 60201 USA
+1 847 491 7699
forbus@northwestern.edu

Jeffrey Usher

Qualitative Reasoning Group
Northwestern University
1890 Maple Avenue
Evanston, IL 60201 USA
+1 847 491 7699
usher@northwestern.edu

ABSTRACT

Many concepts and situations are best explained by sketching. This paper describes our work on sKEA, the *sketching Knowledge Entry Associate*, a system designed for knowledge capture via sketching. We discuss the key ideas of sKEA: *blob semantics* for glyphs to sidestep recognition for visual symbols, *qualitative spatial reasoning* to provide richer visual and conceptual understanding of what is being communicated, *arrows* to express domain relationships, *layers* to express within-sketch segmentation (including a *meta-layer* to express subsketch relationships themselves via sketching), and *analogical comparison* to explore similarities and differences between sketched concepts. Experiences with sKEA to date and future plans are also discussed.

Keywords

Artificial Intelligence Sketching, sketch understanding, qualitative modeling, knowledge acquisition, analogy, diagrammatic reasoning, spatial reasoning.

INTRODUCTION

Sketching is often used when explaining new ideas. The combination of drawing and talking in sketching is a natural means of expression. When interpreted by another participant on the basis of their background and with misunderstandings clarified interactively, sketching provides a rapid means of communicating many complex ideas. Making software that can participate in sketching is a difficult challenge: Ideally, the software needs the full range of human visual, linguistic, and conceptual abilities. Fortunately, by providing some capabilities for drawing and for communicating conceptual material, one can get much of the power of sketching with less than human capabilities. Our *sketching Knowledge Entry Associate* (sKEA) provides a sketch-based interface for knowledge capture. Users describe cases in terms of annotated collections of ink (*glyphs*), where the vocabulary of annotations is drawn

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'02, January 13-16, 2002, San Francisco, California, USA.
Copyright 2002 ACM 1-58113-459-2/02/0001...\$0.00.

from a large knowledge base. The cases they produce can in turn be added to the knowledge base. (sKEA can produce flat files in KIF, MELD, and CML formats.) Unlike traditional multimodal interfaces, which optimize interaction naturalness at the cost of tightly restricted domains, sKEA can be used in any domain (subject only to knowledge base limitations), at the cost of reduced interaction naturalness.

This paper starts by motivating our approach. Then we discuss the key ideas of sKEA: *glyph bars* and *blob semantics* to sidestep the need for recognition of visual symbols, *qualitative spatial representations* to provide richer visual and conceptual understanding of what is being communicated, *arrows* to express domain relationships, *layers* to express within-sketch segmentation (including a *meta-layer* to express subsketch relationships themselves via sketching), and *analogical comparison* to explore similarities and differences between sketched concepts. These ideas will be illustrated with examples from the current version of sKEA. Experience with sKEA so far and future plans will also be discussed.

THE sKEA APPROACH

sKEA is based on our evolving computational model of sketching [19]. Briefly, we argued that sketching can be decomposed into four dimensions: Visual understanding, conceptual understanding, linguistic understanding, and presentation skills. Our work tends to focus on rich conceptual and visual understanding [13,14], as does [2,31]. Most multimodal interfaces (e.g., [3,6,23,25,26,28,32]) strive to maximize fluid interaction, combining statistical recognition of ink strokes and speech recognition to automatically interpret user actions in terms of a fixed vocabulary of conceptual entities. Unfortunately, what they gain in interactive naturalness comes at the expense of sharp limitations in expressive power. Their conceptual vocabulary must be fixed in advance, since the appropriate recognizers, natural language vocabulary, and speech grammars must be constructed to cover it.

sKEA explores a different point in the tradeoff between expressiveness and naturalness. sKEA can operate in arbitrary domains, limited only by the underlying knowledge base and what is natural to express via sketching. The cost is a reduction in interaction flexibility,

because we are, through the design of the interface, asking users to provide information that (in some cases) could in narrower systems be provided automatically via recognition. We think that this approach is important for two reasons. First, it leads to immediately useful systems that can cover a far broader range of domains than today's domain-specific multimodal systems. While architectures such as QuickSet for instance can be set up for a new domain, doing so requires extensive data collection, reengineering grammars, and training recognizers, in addition to whatever hooks are needed to the underlying application program. For many applications, this cost is easily justified in terms of the increased fluidity of the resulting interface. However, for the task of knowledge capture, these additional requirements are especially burdensome, since the system designers do not know in detail what the experts will be telling it in advance. The second reason is that the sketches we accumulate using sKEA constitute a conceptually tagged body of time-stamped ink: Exactly the kind of corpus that is necessary for research into improved visual understanding! Thus we are gathering the data for future improvements, even while providing immediate utility.

GLYPHS AND BLOB SEMANTICS

We call a collection of ink strokes that is intended to represent an entity or relationship a *glyph*. We call that which is represented by a glyph its *content*. Understanding glyphs requires solving two problems: (1) knowing when a glyph has been drawn (*segmentation*) and (2) knowing what a glyph is supposed to mean (*interpretation*). Let us consider each in turn.

Segmentation: In human-to-human sketching, segmentation is solved in a variety of ways, including spatiotemporal contiguity, linguistic cues, and recognition of conventional visual symbols [28]. In multimodal interfaces, constraints such as the pen leaving a surface or timeouts are typically used.

Interpretation: In human-to-human sketching, interpretation is solved by recognition of visual symbols, linguistic labeling (e.g. "Here's the downstream entrance"), and composition of meaning from interpretations of more primitive parts (e.g. the downed pilot example in [19]). In multimodal interfaces, interpretation of glyphs is either done through linguistic cues (e.g., placing a runway in QuickSet) or via recognizing which member

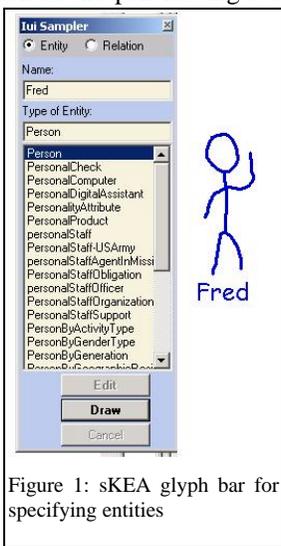


Figure 1: sKEA glyph bar for specifying entities

```
;; Interpretation of glyph in terms of
;; a domain object. Name given applies to
;; the object
(glyphRepresentsObject SKEA-GLYPH-1 OBJECT-1)
(isa OBJECT-1 Person)
(nameString OBJECT-1 "Fred")

;; Information about the glyph itself
(isa SKEA-GLYPH-1 NuSketchGlyph)
(inkLastModifiedTime SKEA-GLYPH-1
  (NuSketchSketchTimeFn 63750))
(nuSketchLayerOf SKEA-GLYPH-1
  USER-DRAWN-SKETCH-LAYER-1)
(q-roundness SKEA-GLYPH-1 NotVeryRound)
(q-2D-orientation SKEA-GLYPH-1 0 1)
```

Figure 2: Typical assertions about a sKEA glyph

of a pre-trained set of glyphs best fits the ink and speech data (e.g. [2,6]).

For knowledge capture, the standard multimodal solutions are not optimal. Adding new knowledge requires adding new visual symbols and new vocabulary. Adding new visual symbols with today's technologies requires extensive data gathering to train statistical recognizers. Adding new vocabulary to speech engines also requires training. Extending the grammar of a parser or speech system requires considerable programming and linguistics skills. Even if tools are provided to carry out this sort of training and development during an interaction, without developers present, it would substantially decrease the naturalness of the interaction due to the huge number of inputs required for today's recognizers compared to human recognition abilities.

In sKEA we take a radically different approach. For segmentation, we provide a button that the user presses to indicate when they are starting to draw a glyph, and that they press again when they are finished. This lets them take as long or as short a time as they want, drawing their strokes in any order and letting them pick up and put down the pen as often as they like. For the interpretation, we provide a selection field where users select which predicate from the underlying knowledge base represents the interpretation of this glyph. For glyphs representing entities, the collections of the knowledge base¹ are available for selection, via a string-completion box. Figure 1 shows the *glyph bar* for entities in the sKEA interface. (We discuss relations shortly.) Once the user completes the glyph, entities and assertions representing both the glyph and its contents are added to sKEA's working memory². Figure 2 shows an example. The time stamp associated with ink is used as one of the antecedents for all conclusions drawn about the visual properties of the glyph,

¹ We currently use Cycorp's Cyc IKB contents for our knowledge base, with Northwestern-developed extensions. A collection is basically a class or category in its ontology.

² sKEA uses the FIRE reasoning engine, being jointly developed by Northwestern and Xerox PARC. FIRE's working memory is an LTMS-based rule engine [17].

so that if it is moved or resized, everything is recomputed appropriately.

We call this representation *blob semantics* because we do not attempt to further decompose the glyph into component parts. A human looking at the sketch of Fred would further interpret part of the ink as a head, part of it as legs, arms and so forth. sKEA doesn't. Its visual analysis of the ink treats it as a blob, constructing for it a bounding box, a connected boundary (there is no requirement that the ink in a glyph be a single connected component), and some simple properties such as an estimate of its principle axis and how round it is. Ultimately we plan to incorporate more sophisticated visual analyses (such as [20,29,30]), but in the meantime, we observe that for many kinds of sketches (e.g., process descriptions, abstract diagrams, node and link diagrams, maps, and some structural descriptions) most of the interesting visual content is in the relationships between the visual entities, rather than in the visual properties of the entities themselves. Moreover, when articulated structure is required, one can still express it using blob semantics, by drawing figures out of multiple blobs (e.g., Figure 8 below). Thus we believe that blob semantics for glyphs is a "sweet spot" in sketch-based systems, and sufficient for a variety of important kinds of knowledge capture problems.

QUALITATIVE SPATIAL REPRESENTATIONS

Visual relationships often convey conceptual information. The relative placement of parts in a structural description

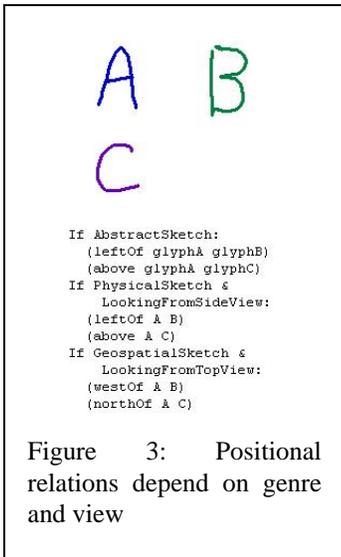


Figure 3: Positional relations depend on genre and view

and the location of buildings and landmarks on a map indicate spatial relationships between the represented contents. Sometimes (e.g., in scale drawings) quantitative information about the contents can be read off the specific distances and directions in the drawing. However, sketches created by hand rarely have this property.

More generally, the stable, intended visual relationships are qualitative in nature. Thus sKEA computes

qualitative spatial descriptions [15] based on the user's ink.

sKEA currently computes two kinds of qualitative spatial relationships. First, it computes qualitative topological descriptions between every pair of glyphs, describing its results using the RCC8 relational vocabulary [7]. This is straightforward and efficient, given the ink as input. Second, sKEA computes positional relationships between glyphs and, when appropriate, between their contents. The positional relationships between glyphs are deictic and

based on the user's perspective of the sketch (i.e., leftOf, rightOf, above, below). The relationships between glyphs lead to inferences about the relationships between their contents depending on two properties of sketches, both of which are explicitly represented in sKEA's KB:

- o The *genre* of a sketch describes the overall type of sketch being made. Examples of genre include AbstractSketch, PhysicalSketch, GeospatialSketch, and DiscreteGraphSketch.
- o The *viewpoint* of a sketch describes the relationship between the visual frame of reference of the glyphs and the spatial frame of reference for the contents. Examples of viewpoint include LookingFromTopView, LookingFromSideView, LookingFromBelowView, and LookingFromDirectionView

Only certain combinations of genre and viewpoint sanction inferences about spatial relationships between contents from visual positional relations on glyphs. For instance, given a combination of PhysicalSketch and LookingFromSideView, the same deictic user-centered vocabulary is assumed to be appropriate. On the other hand, for a GeospatialSketch and LookingFromTopView, the vocabulary eastOf, westOf, northOf, and southOf is used instead. Figure 3 illustrates.

These qualitative spatial descriptions serve two purposes. First, they provide a symbolic summary of visual properties that can be used in analogical matching. That is, when matching both conceptual and visual properties are used, so that diagrams that look alike will to sKEA, like people, seem more alike. Second, sKEA infers conceptual relationships among the entities represented by glyphs when possible, based on knowledge about the sketch. This process is surprisingly subtle. The first step is a translation from qualitative spatial relationships to conceptual relationships. For instance, the RCC8 relationship NTPP ("non-tangential proper part") between two glyphs in a PhysicalSketch such as the cell shown in Figure 4 indicates that the inRegion relationship holds between their contents. Since inRegion is a very general relationship, it

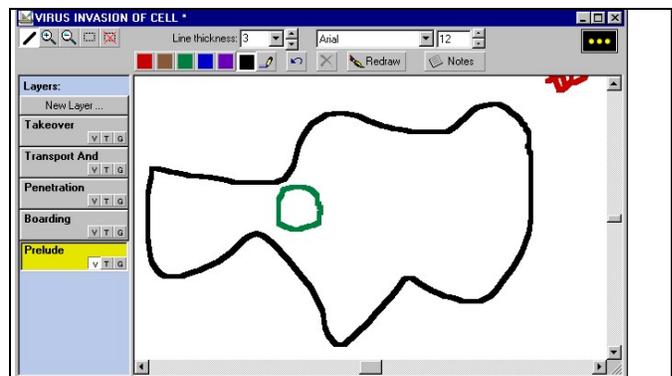


Figure 4: Qualitative spatial relationships suggest conceptual relationships.

is worth looking to see if this relationship could be specialized, to provide more information. Candidates for more specific relationships are filtered via type restrictions on their arguments, i.e., they have to be consistent with the type of entity declared when the glyphs were created. In this case, there are two relationships for these glyphs consistent with `inRegion`, namely that the nucleus is part of the cell itself, versus something that just happens to be found there. Since there is nothing in the sketch that can shed light on this, sKEA must ask the user to select the appropriate interpretation. Such disambiguation questions are queued up because our users found it annoying to be interrupted while they are drawing. Instead, users can choose when (and if) sKEA can question them further.

ARROWS AND BINARY RELATIONSHIPS

A widespread convention in sketches is to use arrows to depict binary relations. That is, the arrow represents a statement relating the entity at its tail and the entity at its tip. The broad cross-domain applicability of this convention and the leap in expressive power it provides led us to include recognition of arrows in sKEA.

As with other glyphs, the glyph bar is used to select a relationship (instead of collection) from the knowledge base, and the Draw button used to indicate the beginning and ending of the ink indicating the relationship. There are two differences however from the entity case: sKEA attempts to automatically recognize which is the tail and which is the tip of the arrow, and based on this information, makes guesses about which entities should be treated as the arguments of the statement which is the content of the relationship glyph.

In the general case arrow recognition can be quite difficult, because arrows can be drawn in a wide variety of ways and their shapes, as they snake around obstacles, can be quite complex. sKEA's arrow recognition routine is a

compromise, restricting the ways arrows can be drawn to maximize robustness. We stipulate that an arrow consists of either two or three strokes. In the two-stroke case, the shorter stroke is interpreted as the head of the arrow, and the longer stroke as the shaft of the arrow. The position of the head of the arrow

is the end of the shaft that is closest to the (centroid of the) head of the arrow, and the position of the tail of the arrow is the other end of the base. In the case of three strokes, the two shortest strokes are interpreted as the head of the arrow, and the rest is handled as in the two-stroke case. Figure 5 illustrates.

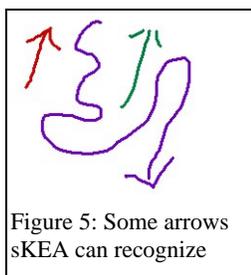


Figure 5: Some arrows sKEA can recognize

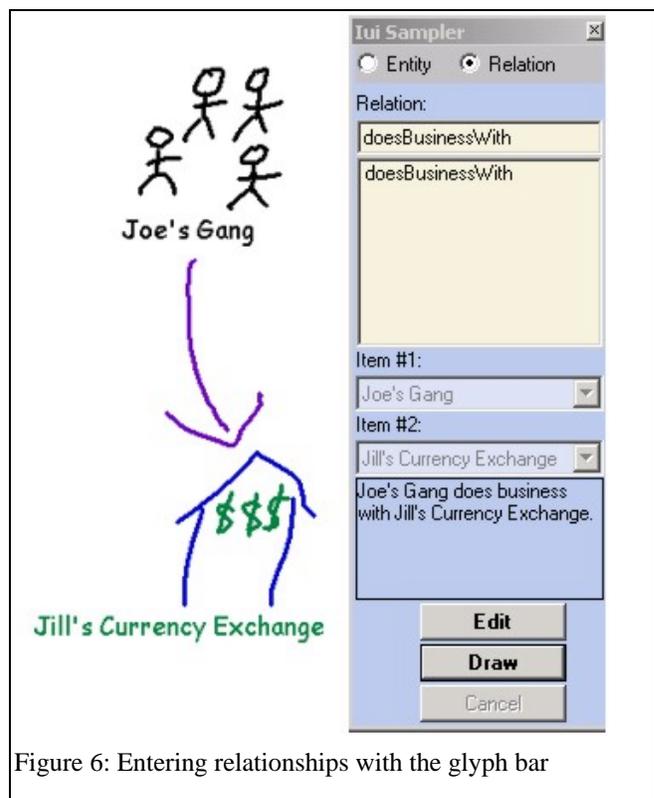


Figure 6: Entering relationships with the glyph bar

Once the head and tail of an arrow have been identified, sKEA uses this information to look for what the arguments of the relationship should be. It selects the closest glyph whose content satisfies the type constraint of that argument of the relationship. (Which argument corresponds to the head and which to the tail is stored in the knowledge base, which is extended by sKEA based on user input when a new binary relationship is used for the first time.) Figure 6 illustrates. Notice the slots for the arguments in the glyph bar, identifying sKEA's conjecture about the arguments to the relation. Should sKEA fail to recognize the user's arrow, or draw an incorrect conclusion about what an argument should be, the user can still indicate the correct argument by dragging its glyph onto the slot.

This design choice has important consequences for the expressiveness of sKEA. It is well known that, using reification, binary relationships are in theory adequate to represent any higher-arity relationship. Thus sKEA's ability to use arrows to represent binary relations, and entities to represent arbitrary collections, means the range of ideas it can be used to express is extremely broad. For example, one can draw concept maps [27] in sKEA, in addition to more overtly physical and geospatial concepts. The main limits of sKEA's expressivity are (a) the predicate vocabulary in its knowledge base and (b) how natural it is to express a piece of information via sketching.

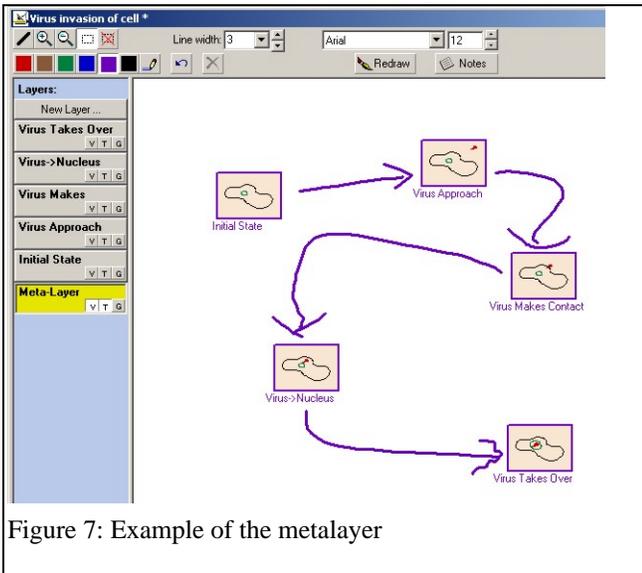


Figure 7: Example of the metalayer

SKETCHES, SUBSKETCHES, AND LAYERS

Sketches often consist of multiple parts. For example, when sketching out a complex process, each step is typically illustrated in a separate portion of the sketch. In describing a complex artifact, one part of the sketch might indicate how the overall artifact works, while other parts of the sketch focus on specific details. In such cases we view a sketch as consisting of a set of *subsketches*, each of which can be viewed as a sketch in its own right. Each subsketch can in principle have a distinct genre and viewpoint. (Imagine for example describing a terrorist attack, where one subsketch is a map of where it happened, one subsketch describes how the weapon used works, and another subsketch traces the financial and command structure of the organization that carried out the attack.) There are a variety of relationships that can hold between subsketches, such as temporal order and causality (e.g., in describing sequences or history), detail/overview, different perspectives, etc. Handling subsketches and relationships between them is thus an important problem for sketch-based interfaces.

In human-to-human sketching, subsketches are segmented in a variety of ways. Explicit linguistic cues are often used. Sometimes explicit boundaries between subsketches are drawn, or separate pieces of paper used. In other cases, it is only spatiotemporal differences and indirect topic shifts that support the inference of subsketch boundaries. As with segmentation for glyphs, we believe that the current state of the art is not reliable enough to do this without causing our users substantial frustration (cf [4,5]). Consequently, as with glyphs, we use an interface organized around knowledge of sketching to provide a workable solution.

In sKEA, users explicitly indicate when they want to create a new subsketch. In terms of the interface, subsketches are depicted as *layers*. sKEA's notion of layer is similar to that used in graphical design and artistic software, as well as that used in military planning. At any point there is a currently selected layer, upon which operations (like adding

glyphs) can occur. Multiple layers can be made visible (like adding acetate overlays on a map), or layers can be grayed out, so that their glyphs are visible but less distracting. An important difference in sKEA's layers is that each layer represents something: That is, layers, like glyphs, have a content, which is an entity that is an instance of one or more collections in the knowledge base. The content for a layer in a sequence is typically an instance of some subclass of *Event*, for example, while the content for a layer in a causal explanation is typically an instance of some subclass of *Situation*. The content of structural descriptions is an instance of the collection being depicted, e.g., *Rabbit*.

When a user adds a layer to a sketch, they must also specify the genre and viewpoint of that layer in addition to selecting what that layer represents. Users have the option of copying the current layer to serve as the starting point of the new layer, which greatly simplifies entering complex sequences and structural descriptions.

Layers provide a means of representing subsketches. But how are relationships between subsketches to be expressed? Again, in human-to-human sketching such relationships are often expressed verbally, but this can be clumsy, especially if the user has to keep track of what relationships already exist between layers. In keeping with the rest of the design of sKEA, we instead use a sketch-based solution. That is, each sketch has a special layer, the *metalayer*. Every other layer in the sketch appears as a glyph on the metalayer. The content of these glyphs is the content of the layer they depict. Relationships between the contents of subsketches are expressed via arrows, just as in other layers. Figure 7 shows the metalayer for a sequence as an illustration.

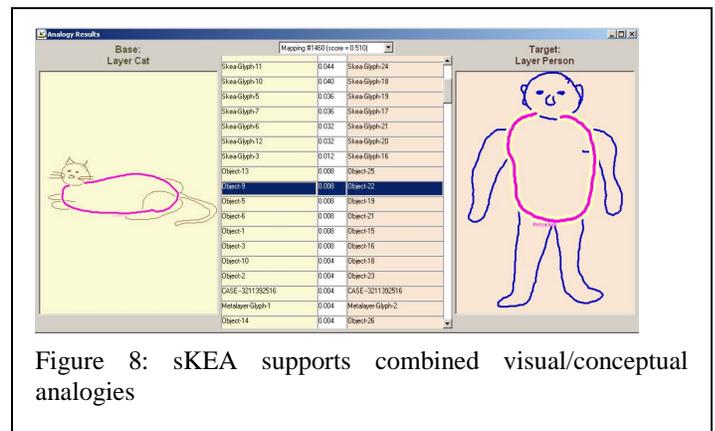


Figure 8: sKEA supports combined visual/conceptual analogies

ANALOGICAL COMPARISON OF SKETCHES

Analogy provides a powerful means of entering and testing knowledge. Currently sKEA enables users to compare two layers, which is useful for examining similarities and differences. We use the Structure-Mapping Engine (SME) [11,18] to perform the matching. SME is a general-purpose analogical matcher, which operates in polynomial time.

SME has been successfully used to model a variety of psychological phenomena, and has generated predictions that have been borne out in subsequent experiments [16]. Psychological plausibility is useful in this task because a shared notion of similarity should facilitate communication between the user and the software. SME has been used in a variety of domains, including visual representations (e.g. [13]). Given the goal of building a knowledge capture tool that can work in a broad variety of domains, this flexibility is essential.

Layers are matched via a drag-and-drop interface using the metalayer. Figure 8 illustrates our interface for browsing matches. The two subsketches being matched are on the left and right sides, with hypothesized matches listed in the middle. Moving to a hypothesized match leads to the corresponding ink parts being highlighted (here, the cat's body and the person's torso). Further hypertext drill-down facilities are provided for inspecting the match and its inferences in detail.

sKEA's analogies are based on both the visual and conceptual material in the sketch for two reasons. First, psychologically people tend to use both factors in judging similarity [22]. Second, people tend to reuse the same visual conventions when drawing the same things [24]. For development, this gives us a useful means of bootstrapping our visual representation scheme, since what looks more similar to us should also look more similar to the software. However, for deployment we plan to offer the option of only retrieving and matching on conceptual content, to see if eliminating the surface bias (especially with sketches entered by different experts) leads to better results.

SME produces *candidate inferences*, conjectures about one description based on its alignment with another. Candidate inferences are useful in knowledge capture because they suggest ways to flesh out a description based on similarities with prior knowledge. Since the analogies concern both visual and conceptual material, the candidate inferences make suggestions about both what might be added to a sketch (e.g., the person does not have a tail nor whiskers) but also where (in terms of the qualitative spatial relationships in the sketch).

EXPERIENCE WITH sKEA SO FAR

The first version of sKEA, without the metalayer, positional relations, and analogical matching capabilities, was delivered to both teams in the DARPA RKF program in May 2001. Participants in that program have been a great source of formative feedback. As an informal experiment, we also asked a number of graduate students and undergraduates, not involved with sKEA development, to try using it. All were able to complete sketches that expressed the gist of what they wanted to represent. Like the RKF users, formative feedback from this experiment led to a number of significant changes that increased the usability of sKEA.

SUMMARY AND FUTURE WORK

sKEA explores a different point in the naturalness versus expressiveness tradeoff than explored by existing multimodal interfaces; where most multimodal systems strive for extremely natural interaction in a tightly constrained domain, sKEA can operate in arbitrary domains, limited only by the underlying knowledge base and what is natural to express via sketching, at the cost of somewhat reduced naturalness. sKEA has already been used, by us and by others, to create a wide variety of descriptions, including structural descriptions of animals, descriptions of sequences in biological processes, and concept maps.

While sKEA is already useful for some purposes, there are a number of immediate improvements we plan to make, including

- Currently sKEA only produces case descriptions as output. While fine for analogical reasoning, producing general axioms would be useful as well. We are adding an interactive explanation-based generalization module to handle this.
- As a person (or team) uses sKEA over time, they will accumulate a portfolio of sketches that (based on observations of human sketching) they will want to refer back to. We will use our MAC/FAC model of similarity-based retrieval to find similar sketches, based on combined conceptual/visual properties (cf. [10,24]).
- The qualitative spatial vocabulary, while already useful, needs to be extended to provide a semantic basis for the full range of human spatial prepositions and spatial relationship systems. Part of this will require adding richer spatial relationships (e.g., Voronoi diagrams for certain spatial prepositions [9]), but part of it will require ensuring that the necessary background knowledge is available in the knowledge base³

In the longer term, we plan to add natural language facilities, creating a task dialogue model for sketching along the lines of [1]. We also plan to incorporate a scale-space blackboard [29] and a MAPS-style visual routines processor [20] to provide more human-like visual abilities.

ACKNOWLEDGMENTS

This research is being carried out as part of the DARPA Rapid Knowledge Formation Program. We thank Jesse Alma, Marion Ceruti, and the anonymous reviewers for insightful comments.

³ Coventry [8] and Feist & Gentner [12] have demonstrated that human use of spatial prepositions is not purely geometric, but incorporates physical and teleological knowledge as well.

REFERENCES

1. Allen, J.F. et al, The TRAINS Project: A Case Study in Defining a Conversational Planning Agent, *Journal of Experimental and Theoretical AI*, 1995.
2. Alvarado, Christine and Davis, Randall (2001). Resolving ambiguities to create a natural sketch based interface. *Proceedings of IJCAI-2001*, August 2001.
3. Bolt, R. A. (1980) Put-That-There: Voice and gesture at the graphics interface. *Computer Graphics*. **14**(3), 262-270.
4. Clark, H. H. 1996. *Using language*. Cambridge University Press.
5. Clark, H.H. 1999. Speaking in Time. Proceedings of the ESCA workshop on dialogue and prosody, September 1-3, Veldhoven, the Netherlands.
6. 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.
7. 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.
8. 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.
9. 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.
10. Egenhofer, M. (1997) Query Processing in Spatial-Query-by-Sketch in *Journal of Visual Languages and Computing* 8(4), 403-424 pp.
11. Falkenhainer, B., Forbus, K., Gentner, D. (1989) The Structure-Mapping Engine: Algorithm and examples. *Artificial Intelligence*, 41, pp 1-63.
12. 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*
13. Ferguson, R.W. and Forbus, K.D. 2000. GeoRep: A Flexible Tool for Spatial Representation of Line Drawings. *Proceedings of AAAI-2000*. Austin, Texas.
14. 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.
15. 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.
16. Forbus, K. 2000. Exploring analogy in the large. In Gentner, D., Holyoak, K. and Kokinov, B. (Eds) *Analogy: Perspectives from Cognitive Science*. Cambridge, MA: MIT Press.
17. Forbus, K., and de Kleer, J. 1993. *Building Problem Solvers*, MIT Press.
18. Forbus, K., Ferguson, R. and Gentner, D. (1994) Incremental structure-mapping. *Proceedings of the Cognitive Science Society*, August.
19. Forbus, K., Ferguson, R. and Usher, J. 2001. Towards a computational model of sketching. *IUI'01*, January 14-17, 2001, Santa Fe, New Mexico
20. 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.
21. Forbus, K., Nielsen, P. and Faltings, B. "Qualitative Spatial Reasoning: The CLOCK Project", *Artificial Intelligence*, **51** (1-3), October, 1991.
22. Gentner, D. and Markman, A. 1997. Structure Mapping in Analogy and Similarity. *American Psychologist*, January, pp 45-56
23. Gross, M. (1996) The Electronic Cocktail Napkin - computer support for working with diagrams. *Design Studies*. 17(1), 53-70.
24. 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.
25. Landay, J. and Myers, B. 1996. Sketching storyboards to illustrate interface behaviors. CHI'96 Conference Companion: Human Factors in Computing Systems, Vancouver, Canada.
26. Maybury, M. and Whalster, W. 1998. *Readings in Intelligent User Interfaces*. Morgan Kaufmann.
27. J.D. Novak and D.B. Gowin, *Learning How To Learn*, New York: Cambridge University Press. 1984.
28. Oviatt, S. L. 1999. Ten myths of multimodal interaction, *Communications of the ACM*, Vol. 42, No. 11, November, 1999, pp. 74-81.

29. Saund, E. 1990 Symbolic Construction of a 2-D Scale-Space Image. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, No. 8.
30. Saund, E., and Moran, T, (1995) Perceptual Organization in an Interactive Sketch Editing Application. *ICCV '95*
31. 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.
32. Waibel, A., Suhm, B., Vo, M. and Yang, J. 1996. Multimodal interfaces for multimedia information agents. *Proc. of ICASSP 97*