Towards a Computational Model of Sketching

Kenneth D. Forbus, Ronald W. Ferguson, Jeffrey M. Usher

Qualitative Reasoning Group, Northwestern University

Abstract

Sketching is a powerful means of communication between people, and while many useful programs have been created, current systems are far from achieving human-like participation in sketching. A computational model of sketching would help characterize these differences and better understand how to overcome them. This paper is a first step towards such a model. We start with an example of a sketching system, designed to aid military planners, to provide context. We then describe four dimensions of sketching, *visual understanding, conceptual understanding, language understanding,* and *drawing,* that can be used to characterize the competence of existing systems and identify open problems. Three research challenges are posed, to serve as milestones towards a computational model of sketching that can explain and replicate human abilities in this area.

Introduction

Person-to-person communication often involves diagrams, charts, white boards, and other shared surfaces. People point, mark, highlight, underscore, and use other gestures to help disambiguate what they are saying. Being able to use multiple modalities, i.e., speech and gesture, to communicate ideas is especially crucial for spatial information [1,4,6,21]. The ability to understand spatial representations, and to use them appropriately in dialogue, is a critical skill that we need to embed in software, in order to create systems that understand the users they are interacting with.

We focus here on *sketching*, meaning a communication activity involving a combination of interactive drawing plus linguistic interaction. The drawing carries the spatial aspects of what is to be communicated. The linguistic interaction provides a complementary conceptual channel that guides the interpretation of what is drawn. Most people are not artists, and even artists cannot produce, in real time, drawings of complex objects and relationships that are recognizable solely visually without breaking the flow of conversation. The verbal description that occurs during drawing, punctuated by written labels, compensates for inaccuracies in drawing. Follow-up questions may be needed to disambiguate what aspects of a drawing are intended versus accidental.

There is now a substantial body of research on multimodal interfaces [19]. Sketching is clearly a form of multimodal interaction, but not all multimodal interaction is sketching. Many multimodal interfaces focus on placement of predefined entities, e.g., selecting a location, often via pointing (c.f. [1,6]). Such selection operations require a fairly minimal shared understanding on the part of the participants, and hence has provided a natural starting point for multimodal interface research. Work that comes closer to sketching (c.f. [8,13,23]) incorporates more domain semantics, to increase the level of shared understanding. 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, AI research. This paper examines sketching in that light, to provide a framework for understanding the phenomena and suggesting new research directions.

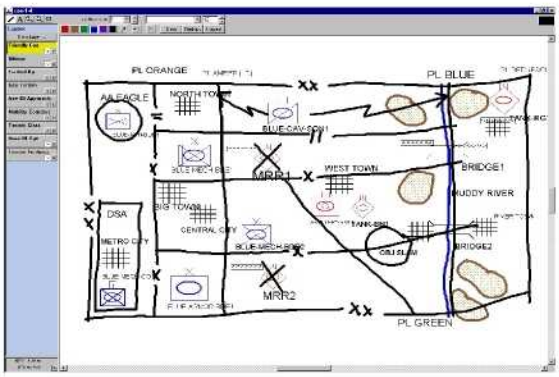

Figure 1: A Course of Action Sketch

The rest of this paper describes our progress towards a computational model of sketching. We start with an example, our nuSketch multimodal interface architecture, showing how it has been used to create a system for reasoning about military courses of action. We then step back and describe a framework for sketching, motivated by a combination of constraints from computation and from cognitive science research. We end by identifying three challenges for research on sketching.

Submitted to AAAI-2000

nuSketch: A multimodal architecture for sketching

nuSketch is designed as a general-purpose multimodal architecture to support sketching. The best way to illustrate nuSketch's abilities is through an example application. Military planners use a *Course of Action sketch* (COA sketch) when designing an operation. COA sketches express the gist of a plan, before many details, such as timing, have been worked out. Traditionally such sketches are created using acetate overlays on maps, or on paper starting with hand-drawn abstractions of critical terrain features. A well-worked out vocabulary of visual symbols is used to represent terrain features, military units, and tasks assigned to units.

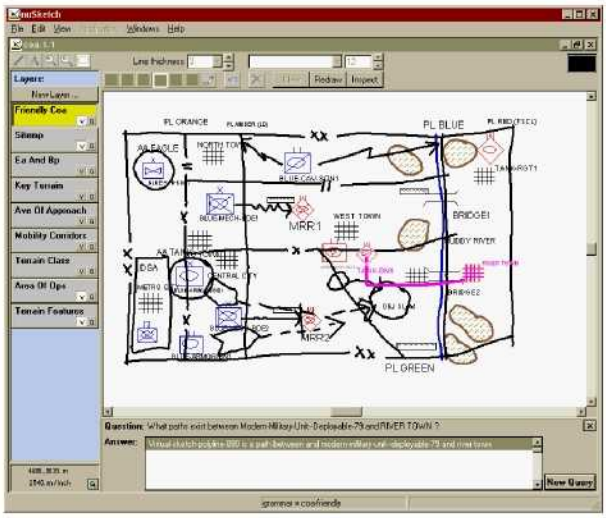

Figure 2: nuSketch provides geographic reasoning. Here the user asked for shortest trafficable path to an objective. The path found is outlined in pink

Figure 1 illustrates a course of action drawn using the nuSketch COA Creator. A *layer* metaphor organizes the interface. Like acetate layers, each nuSketch layer corresponds to some category of domain information, such as terrain analysis, enemy disposition, disposition of your 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. 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, e.g. towns, the user's gesture indicates a bounding box, and the appropriate glyph is retrieved from the KB and displayed there, scaled appropriately. A set of assertions constituting the system's conceptual understanding of the visual element and what it represents in domain terms is also created. This conceptual understanding facilitates reasoning to support the user. For instance, geographic queries are made by dragging and dropping sketch elements onto a simple parameterized dialog (Figure 2). These queries are answered by using qualitative and visual reasoning to interpret the spatial entities and relationships in the sketch. Similarly, users can request critiques based on analogies with prior plans, with the application of the advice to their plan illustrated by the system highlighting the appropriate visual elements of the sketch (Figure 3). The analogical mapping is driven by the visual and conceptual descriptions constructed during sketching.

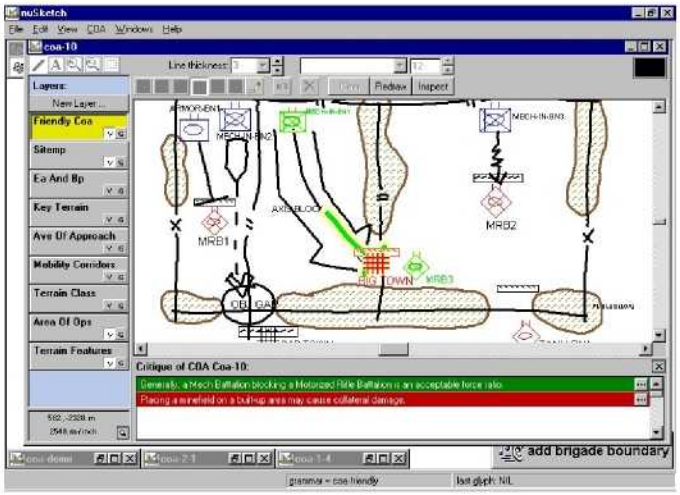

Figure 3: Analogies can be used to provide critiques, with results displayed by highlighting elements of the user's sketch

Figure 4 shows the nuSketch architecture. 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 (DTE[1]), which provides integrated access to

[1] DTE stands for *Domain Theory Environment*, the reasoning system. It combines a prolog-style query-driven inference system with a logic-based TMS to enable heterogeneous inference systems to interoperate.

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 how 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).

Several aspects of nuSketch are inspired by Quickset [6], a multimodal interface system for setting up military simulations. Like Quikset, we use off-the-shelf speech recognition and time-stamp ink and speech signals to facilitate integrating information across modalities. Quickset incorporates ink-recognition schemes that nuSketch does not (as a matter of principle; see below). Because Quickset was designed as an interface for legacy

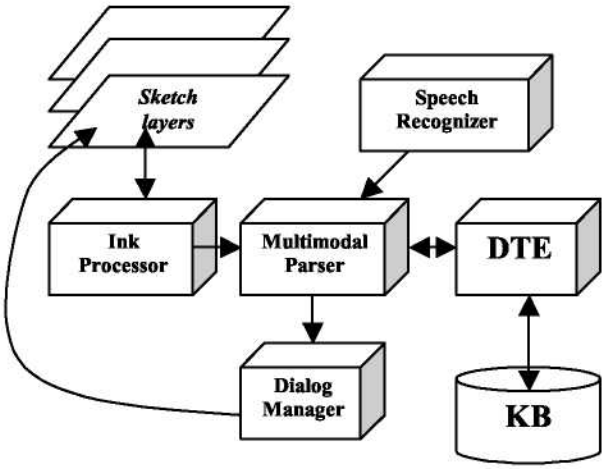

Figure 4: nuSketch architecture

computer systems, it lacks an integrated reasoning system. As the discussion below will make clear, this significantly limits its potential as a model of sketching. For example, it does not reason about depiction as nuSketch can.

Dimensions of sketching

The power of sketching in human communication arises from the high bandwidth it provides [21]. There is high *perceptual bandwidth* because the shared drawing is interpreted by the participants' powerful visual apparatus. There is high *conceptual bandwidth* because the combination of visual and linguistic channels facilitates the interaction needed to create a shared conceptual model.

Sketching covers a wide span of activities that occur under a variety of settings. A computational model of sketching must identify what knowledge and skills the participants need. We characterize these competencies along four dimensions: *visual understanding*, *language understanding*, *conceptual understanding*, and *drawing skills*. Variations along these dimensions determine how many different types of interactions something having those skills can participate in. We describe each in turn.

Visual understanding. This dimension characterizes how deeply the spatial properties of the ink are understood. The simplest level of understanding is recognizing gestures. Gestures indicate locations or sizes, often including an action to be taken with regard to something at that location (e.g., selecting or deleting) [2,6,24]. We do not consider a system with only this level of visual understanding to be capable of sketching, since it does not understand the spatial relationships between visual elements.

The next level of visual understanding is the use of a visual symbology, i.e. a collection of glyphs, representing conceptual elements of the domain whose spatial properties can also convey conceptual meaning. Schematic diagrams in various technical fields and formal visual languages such as the military task language illustrated above are two examples. Is a CAD system a participant in sketching? We argue no, for two reasons. First, it is not taking an active role as a participant. In multimodal interactions, even during data entry the system is engaged in recognizing the kinds of entities and actions the user intends. Second, the time and conceptual overhead needed to deal with menus prevents the maintenance of a conversation-like flow [6,21]. By contrast, multimodal systems that use recognition procedures to "parse" ink automatically (c.f. Quickset), or use speech plus gesture to identify entities (c.f. nuSketch) keep the interaction more like dealing with another person, someone capable of looking at what you are drawing and hearing what you are saying, and responding appropriately.

Most multimodal systems rely on a combination of speech and black-box recognition algorithms (e.g., hidden Markov models or neural nets) operating on digital ink to identify a user's intent (c.f. [6,12]. While certainly useful in some applications, we claim that they are detours from the path that leads to human-like sketching capabilities. The reason is that people are very flexible in their use of visual symbols, and they expect the same flexibility from their partners. Three examples illustrate this point.

Figure 5 illustrates how complex visual symbols can be. In terrain analysis, broad red arrows indicate avenues of approach. This multi-headed "arrow", drawn by a military officer, accurately conveys where units might move. However, it is hard to see

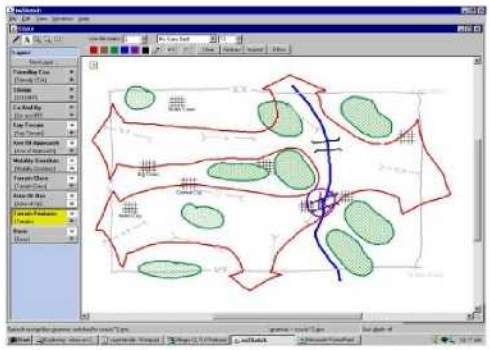

Figure 5: Visual symbols can be complex

how any statistical recognizer could be trained up in advance to recognize such a complex figure.

Figure 6: Two arrows

Figure 6 shows two arrows that are very similar, except for the style in which they are drawn. Most visual symbologies assign different meanings to dashed versus solid arrows, so it would not be enough to simply recognize both of these as arrows. The richness of visual properties that can arise even with very stylized visual symbologies is illustrated by Figure 7, which shows a symmetric pair of attacks that has been automatically identified by high-level visual reasoning [9].

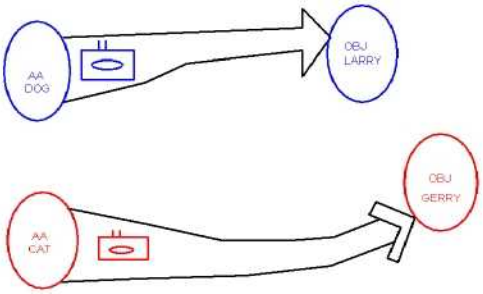

Figure 7: Higher-level visual constructs such as symmetry communicate information

These examples suggest that powerful visual skills are one of the keys to human-like sketching. We suspect that work like Saund's [22] on perceptual organization will play a major role in bringing sketching systems closer to human capabilities.

Conceptual understanding: As a communicative act, sketching requires common ground [3]; 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 model 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 (c.f. [5,6,12,24]). 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 (e.g., Quickset, particular nuSketch applications) 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.* Being able to reason about regions, paths, and relative locations is important in every spatial domain. [10,7]
- *Qualitative representations of shape.* The ability to abstract away minor differences in order to describe important properties facilitates recognition. [8]

We claim that qualitative 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. [11,23] The nuSketch COA Creator, 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.

Other types of general knowledge are needed for flexible sketching as well:

- *Graphical conventions.* Many conventions used in drawings are deliberately unrealistic, e.g., cutaways to show the internal structure of a complex object. Using sequences of snapshots to depict dynamics requires interpreting spatial repetition as temporal progression. Understanding these conventions is necessary for many types of sketches.
- *Standard visual symbols.* Part of our shared visual language consists of simplified drawings that convey complex concepts easily. Stick-figure drawings and many types of cartoons provide examples.

Graphical conventions and visual symbols require combining visual/spatial knowledge with conceptual knowledge, and thus we suspect are a crucial area for improvement to create better sketching systems.

Language Understanding: Language provides several services during sketching. It can ease the load on vision by labeling entities, specifying what type of thing is being drawn, stating what spatial relationships are essential versus accidental, and describing entities and relationships not depicted in the drawing. Speech is the most common modality used during sketching because it enables visual attention to remain on the diagram, although short handwritten labels are often used as well. Existing multimodal systems tend to use off-the-shelf speech recognition systems, limiting them to finite-state or definite clause grammars (c.f. [4,5,19]). Given the differences in complexity between spoken and written text, such grammars, albeit with multimodal extensions, are likely to remain sufficient [1]. The most important dimension for characterizing language understanding in sketching systems concerns dialogue management [15,18]. 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.

Drawing capabilities: Sketching is a two-way street; ideally visual and linguistic expression should be modalities available to all participants (c.f. [20]). The state of the art in natural language generation and text to speech is constantly improving, and such improvements will of course benefit sketching systems. Visual expression by sketching programs provides some new challenges. The simplest forms of visual expression are highlighting and performing operations on human-drawn elements (e.g., moving, rotating, or resizing). Some systems complement their human partners by neatening their diagrams (c.f. [17]). The ability to modify a user's sketch, and generate their own sketches to start a dialog, are beyond the present state of the art. Significant progress has been made on expressing the visual skills needed for graphical production tasks such as layout (c.f. [19]), the key barriers for sketching are the lack of understanding of both the domain and visual representations, as outlined above.

Challenges for computational models of sketching

The discussion of the dimensions of sketching should make it clear that, while currently we can build software that participates in sketching in a limited way, the state of the art is far from creating systems that have the depth and flexibility of a human partner. In the spirit of encouraging progress, we suggest three challenges as useful benchmarks to measure progress in the area.

Integrated compositional semantics: The preponderance of systems that use "black-box" recognizers is more a function of them being easily available and of the limited range of tasks tackled to date than their suitability for use in sketching. An example provides the best illustration. The symbols used on military maps are highly standardized, with thick books providing visual symbols for almost every conceivable occasion. Nevertheless, during military exercises unique visual symbols are sometimes generated to cover special needs. Figure 8 shows an icon, drawn on a post-it, that appeared in several places on an intelligence map in a recent US Army exercise.

Figure 8: A novel, but easily understandable, visual symbol

This symbol represents a downed US pilot at its location. Although this symbol cannot be found in any military manual, it is quite easy to interpret. Even non-military people tend to get it after one or two leading questions (what is the thing on the right? Okay, it's a crashed airplane. Who might that be?). There are degrees of ambiguity in the interpretation: Some people interpret the dashed lines coming out of the pilot's head as sweat rather than tears, and some think the person is a passenger. But no one who is told what the symbol means has trouble identifying the airplane and the pilot and the pilot's unhappy/stressed state as a consequence.

This very simple interpretation problem requires an enormous amount of knowledge: Of airplanes, pilots, and their relationships, of the visual appearance of airplanes and how modifications of that appearance might be interpreted (i.e., a crashed airplane), and of conventions for depicting people and their states. The ability to combine visual and conceptual understanding in a compositional way to decode complex sketches is, we believe, a major challenge for computational models of sketching.

User-extensible visual symbologies: As more open domains are attempted, restricting users to a predefined vocabulary of visual symbols will be infeasible. Asking a user to provide dozens to hundreds of samples to train a statistical recognition system in order to add a new glyph, for instance, is quite unnatural. When a new glyph is introduced in human-to-human sketching, the introducer may have to linguistically mark its occurrence for a while when first used, but over time the other participants learn to recognize it. Being able to interactively specify the domain semantics of a new glyph, and have the software start picking up how to recognize it through normal interactions, will be an important benchmark since it will enable the bootstrapping of sketching systems.

Visual analogies: Being able to compare sketches is an important aspect of comparing what the sketches are about (e.g., comparing engineering designs or comparing COAs).

Shared history provides an important form of common ground, so the ability to recognize when aspects of the current sketch have been seen before will enable software participants to take on more of a community memory role. Currently there are domain-specific systems that do sketch-based retrieval [8,14], but these only operate in narrow domains. Some progress has been made on using similarity in visual encoding, particularly to detect symmetry and regularity in line drawings[9], but using these and other *analogical encoding* techniques in visual understanding is currently an area of active research.

Sketching systems that can carry out such visual analogies and retrievals in a broad range of domains will be another important benchmark in modeling sketching.

Discussion

Sketching is a powerful human-to-human means of communication, and a potentially powerful metaphor for human-computer interaction. We have argued that the state of the art is still far from creating software that participates in sketching with the same fluency as humans. We argued that two key areas of improvement are depth of conceptual understanding and visual processing. The three challenges we outlined provide, we believe, benchmarks that would mark significant advances towards more

human-like sketching systems. Even leaving aside its importance as an interface modality, 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. We hope that this paper encourages more research in this area.

Acknowledgements: This research was supported by DARPA under the High Performance Knowledge Bases and Command Post of the Future programs.

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. Bolt, R. A. (1980) Put-That-There: Voice and gesture at the graphics interface. *Computer Graphics*. **14**(3), 262-270.
3. Clark, H. 1996. *Using language*. Cambridge University Press.
4. Cohen, P. (1992) The role of natural language in a multimodal interface. UIST92, pp 143-149.
5. Cohen, P., Dalrymple, M., Moran, D., Pereira, F., Sullivan, J., Gargan, R., Schlossberg, J. and Tyler, S. (1989) Synergistic use of direct manipulation and natural language. *Proceedings of CHI-89*, pp 227-233.
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. Egenhofer, M. (1997) Query Processing in Spatial-Query-by-Sketch in Journal of Visual Languages and Computing 8(4), 403-424 pp.
9. Ferguson, R.W. and Forbus, K.D. 2000. GeoRep: A Flexible Tool for Spatial Representation of Line Drawings. *Proceedings of AAAI-2000*. Austin, Texas.
10. 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.
11. Forbus, K., Nielsen, P. and Faltings, B. "Qualitative Spatial Reasoning: The CLOCK Project", *Artificial Intelligence*, **51** (1-3), October, 1991.
12. Gross, M. (1994) Recognizing and interpreting diagrams in design. In T. Catarci. M. Costabile, S. Levialdi, G. Santucci eds., Advanced Visual Interfaces '94, ACM Press, 89-94.
13. Gross, M. (1996) The Electronic Cocktail Napkin - computer support for working with diagrams. Design Studies. 17(1), 53-70.
14. 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.
15. Grosz, B.J., and Sidner, C.L., "Attention, Intentions, and the Structure of Discourse", *Computational Linguistics*, 12:3 (1986)
16. Henderson, K. 1999. *On Line and On Paper: Visual representations, visual culture, and computer graphics in design engineering*. MIT Press.
17. Julia, L. and Faure, C. 1995. Pattern Recogntion and Beautification for a pen-based interface. *Proceedings of ICDAR'95*: Montreal (Canada), pp 58-63.
18. Luperfoy, S. (editor) *Automated Spoken Dialogue Systems*. MIT Press (in preparation).
19. Maybury, M. and Whalster, W. 1998. *Readings in Intelligent User Interfaces*. Morgan Kaufmann.
20. Moran, D.B., Cheyer, A.J., Julia, L.E., Martin, D.L., and Park, S. (1997) Multimodal user interfaces in the Open Agent Architecture. *Proceedings of IUI97*. pp 61-68.
21. Oviatt, S. L. 1999. Ten myths of multimodal interaction, *Communications of the ACM*, Vol. 42, No. 11, November, 1999, pp. 74-81.
22. Saund, E., and Moran, T, (1995) Perceptual Organization in an Interactive Sketch Editing Application. *ICCV '95*
23. 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.
24. Waibel, A., Suhm, B., Vo, M. and Yang, J. 1996. Multimodal interfaces for multimedia information agents. *Proc. of ICASSP 97*