EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2009)
C. Grimm and J. J. LaViola Jr. (Editors)

Visual Languages and Visual Thinking: Sketch Based Interaction and Modeling

M. D. Gross

Carnegie Mellon University, USA

Abstract

Research on sketching with computers dates to the earliest days of modern computing. Recent work in this area, combined with other advances in hardware and software technologies promises, finally, significant impact. The kinds, qualities, and purposes of sketch-based interaction, or visual languages, vary as widely as do other forms of language. In addition to practical applications in every domain, advances in sketch-based interaction and modeling can help us understand and support visual thinking.

1. Drawings matter because drawings help us think

Ken Iverson, the inventor/designer of the powerful programming language APL began his 1979 Turing award lecture, titled "Notation as a Tool of Thought", as follows:

The importance of nomenclature, notation, and language as tools of thought has long been recognized. In chemistry and in botany, for example, the establishment of systems of nomenclature by Lavoisier and Linnaeus did much to stimulate and to channel later investigation. Concerning language, George Boole in his Laws of Thought [1, p. 24] asserted "That language is an instrument of human reason, and not merely a medium for the expression of thought, is a truth generally admitted." [Ive79]

Language and thinking are intimately bound together. Usually we think of language as verbal—that is, spoken and written (and in the case of sign language, gestural). However, we recognize that there is a world of graphical languages: we use maps to navigate, we draw graphs to illustrate mathematical relationships, we draw diagrams to illustrate sports plays. Yet these are special cases in which drawing augments our verbal forms of communication for very particular purposes.

Still, most would agree that for some purposes, or in some domains, a drawing is a more concise way than verbal language to express something. We make a diagram of a traffic accident because it expresses in a few lines what would take a page of text to explain. We draw a picture of an object or a scene because we can show in an instant what might take a minute to explain, or which might be hard to convey in words at all. It encapsulates information more efficiently: at least, fewer symbols are needed.

In some cases a drawing is also a powerful way to reason. An electrical engineer can quickly infer from a circuit diagram—by inspection, a short circuit, a misplaced transistor—what would be tedious and laborious to determine from a list of connections or verbal description. Looking at a mechanical diagram we can see consequences: We can understand what happens when the cam turns.

Drawings leverage our inherent (that is, acquired through everyday life) experience. Although of course we can add, subtract, and multiply vectors without drawing, we experience a sense of comprehension—perhaps call it 'embodied'—when we make or see a vector sum expressed graphically.

Today it is in the arts that drawing and sketching are most appreciated, although in our modern culture we often do not consider the arts, and drawing, as a form of thinking. Perceptual psychologist Rudolf Arnheim remarked on this in the introduction to his 1969 book, "Visual Thinking"

Today, the prejudicial discrimination between perception and thinking is still with us.... ***Our entire educational system continues to be based on the study of words and numbers.*** *In kindergarten, to be sure, our youngsters learn by seeing and handling handsome shapes and invent their own shapes on paper or in clay by thinking through perceiving. But with the first grade of elementary school the senses begin to lose*

Sketch-Based Interfaces and Modeling 2009, New Orleans, LA, August 1–2, 2009.
© 2009 ACM 978-1-60558-602-1/09/0008 $10.00

educational status.... The arts are neglected because they are based on perception, and perception is disdained because it is not assumed to involve thought. In fact, educators and administrators cannot justify giving the arts an important position in the curriculum unless they understand that the arts are the most powerful means of strengthening the perceptual component without which productive thinking is impossible in any field of endeavor. The neglect of the arts is only the most tangible symptom of the widespread unemployment of the senses. ***What is most needed is ... a convincing case made for visual thinking quite in general*** [Arn69], p.2-3 emphasis mine]

Finally, in his popular little book, "How to Solve It," mathematician George Polya gives the advice: *"Draw a figure; introduce suitable notation. Figures are not only the object of geometric problems but also an important help for all sorts of problems in which there is nothing geometric at the outset."* [Pol45], p.93]

Iverson, Arnheim, and Polya (and the list could go on) all remind us to consider the work of the Sketch Based Interaction and Modeling research community as more than a mere matter of 'interface.' This work will result in enabling computers to handle our sketching and drawing, to support computer understanding of visual languages, and thereby visual forms of thinking.

2. Sketch based interaction comprises many communities

It has been a half century since the introduction of the light pen (1952) [Car09] and the first popular digitizing tablet [DE64]. These hardware technologies have enabled many software projects—beginning with Sketchpad (1963) and GRAIL [EHS]—that, together, we now call sketch-based interaction and modeling (SBIM). SBIM encompasses a broad range of work loosely defined by research around interactive graphics with input from a stylus. Under this rubric we find a variety of research interests and approaches.

Work on sketch based interaction and modeling takes place in a wide and somewhat disparate set of research communities. For example, work on visual thinking and diagrammatic reasoning takes place in the artificial intelligence community, and is published in the IJCAI and AAAI conferences and on occasion in the Spring and Fall symposia of the AAAI. Work on sketching for modeling takes place in the graphics communities of Eurographics and SIGGRAPH and is published in the corresponding venues. Interaction aspects of sketching and visual programming languages are studied in the human-computer interaction community, and this work is published in the Conference on Human Factors (CHI), User Interaction and Software Tools (UIST), the Visual Languages and Human-Centric Computing (VL-HCC) and other HCI venues. Research on the cognition of drawing and sketching is published in cognitive science conferences and journals. And still other work is published in the conferences and journals of specific domains, such as mechanical engineering, graphic, architectural, and industrial design. These rather diverse communities overlap only partially, which challenges those working in the field to keep track of what has been done. A partial survey of this literature can be found in [JGHD08].

With this diversity of research communities in mind we can turn to the space of kinds of drawings and kinds of systems that we include under the broad rubric of "sketching." Figures 1 and 2 below reflect an attempt to survey the spaces of drawings and systems for supporting sketch based interaction and modeling. As with any such effort to organize a discipline, the dimensions of these spaces will not neatly serve all cases, and alternative schemas are certainly possible. And the particular well-known examples in Figure 2 are intended solely to illustrate, not to define. I invite readers to locate their own work within this dimensional schema.

2.1. Kinds of drawings

Sketch based interaction and modeling encompasses a range of drawing types and uses—from the casual and quickly made napkin diagram to a carefully constructed drawing in which every stroke is deliberately and painstakingly shaped. Sketches include expressions in formal visual languages in which symbols and their spatial relationships are mapped explicitly to meanings as well as informal (but none-the-less meaningful) representations. Purposes vary, from symbol manipulation to direct shape- and form-making. The UI designer making a UML diagram or the engineer laying out a circuit operates within a clearly and strictly limited vocabulary and grammar, and anything beyond these bounds is error. Indeed, the clear and strict limits of the language help enforce domain constraints. On the other hand, an industrial designer shaping the form of a lamp works within a more open visual language, although still the drawings may reflect domain constraints (e.g., Cheng's law-encoding diagrams [Che96]). Even when, as in mechanical or architectural drawings, the visual language used to express a complete design is quite clearly defined, early sketches may be rather looser, not adhering to the formal grammar of the language used to express designs in the domain.

Figure 1 shows dimensions of the space of kinds of drawings. The vertical axis describes **domains**, from geometric to non-geometric. The geometric (bottom) end of the axis stands for domains in which the geometry of the sketch corresponds directly to geometry in the domain: for example, graphical user interfaces (GUIs), architecture, product design, geography, and the physical aspects of mechanical engineering. The geometry can be two-dimensional (as in a map or floor plan) or three dimensional, as in a drawing of a product or building. In these domains, although the drawing marks may represent things symbolically, the sizes and positions of the marks correspond more directly to the sizes

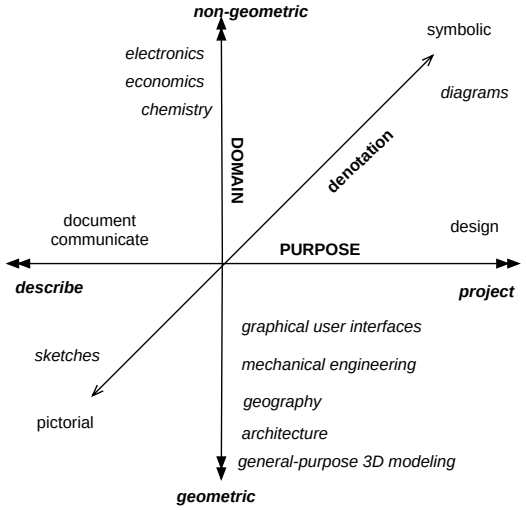

Figure 1: *Kinds of drawings.*

and positions of physical elements or features in the physical reality that is being represented.

On the non-geometric (top) end of the axis we find domains such as economics, chemistry, and electronics. Here the sizes and positions of the marks in the drawing correspond less directly or not at all to the domain elements represented. Instead, the sizes, positions, and dimensions of the drawing are (or can be) mapped to other quantities or qualities of the domain. (Figure 1 itself is an instance of this, in that the axes and the distances between points in the graph do not represent physical geometric space, but rather they represent attributes of drawings).

The horizontal axis in Figure 1 represents the different **purposes** that a drawing (sketch or diagram) can serve. The left end of the axis represents drawing to describe the existing state of something—a street map, a product, a scene. In these, the purpose is to accurately convey some reality. The right end of the axis represents drawing to project a possible future state, that is, to design. The purpose of these drawings is to consider possible alternatives that do not (yet) exist.

Finally, in Figure 1 the axis orthogonal to the xy (domain/purpose) plane represents the degree to which a drawing is directly **denotational**. At one end of this axis are drawings that are highly symbolic—there is a one-to-one correspondence between drawing marks and domain elements. We might say that these highly symbolic drawings are *diagrams*; whereas on the other end of the axes are *sketches*—drawings in which the drawing marks do not directly denote domain elements in a one-to-one mapping.

2.2. Kinds of systems

Figure 2 shows another set of dimensions, of sketch based interactions. The vertical **descriptiveness** axis represents the tasks for which pen based interaction is used. At the top of the vertical axis are systems in which pen based interaction is used to make a drawing or a three-dimensional representation, such as TEDDY [IMT99], Sketch,and more [ML07]. (This corresponds to the 'M' in SBIM). At the bottom of the vertical axis are systems in which pen based interaction is used mainly to interact with, point, edit, or mark up a document such as CrossY [AG05] (this corresponds to the 'I' in SBIM).

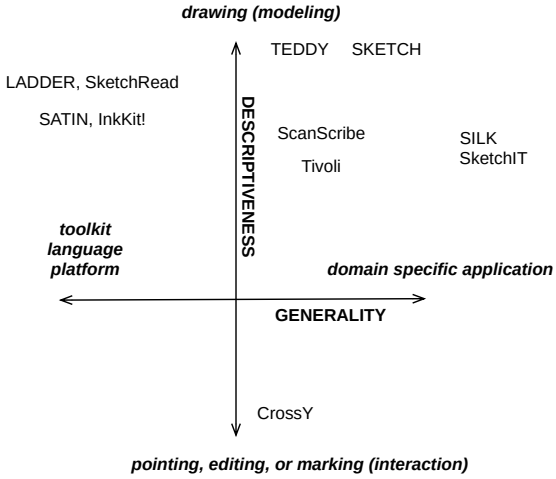

Figure 2: *Kinds of systems.*

The horizontal **generality** axis indicates the degree to which a system is intended or designed to support a specific domain. The right end of the axis corresponds to systems that are highly specific to a particular domain, problem, or task. The SILK system [LM95], for example, was tailored for prototyping graphical user interfaces. At the other end of this axis are systems like SketchRead [AD04], LADDER [HD06], SATIN [HL06],and InkKit! [PF07] which have no specific application domains, but are intended as general-purpose platforms for building sketch-based interactive systems. And we might put domain-agnostic systems such as Tivoli [MvMC98] or ScanScribe [SFLM03] in the middle — these are systems for end-users (not a toolkit, language, or platform) but they can be used to work with drawings in any domain. These systems are fairly high on the descriptiveness axis, because they all enable users to make drawings.

To be sure, many systems developed in the context of a specific domain or task (such as SILK, for prototyping user interfaces or MathPad for sketching mathematics [Lav05], or architecture [Do02] or SketchIT for mechanisms [Sta97]) convey ideas about sketch-based interaction

and modeling that can be much more broadly applied. To some extent, it is a matter of style whether one begins on the toolkit/language/platform side of the axis and builds domain applications to test and demonstrate the effectiveness of the platform, or whether one builds a domain application that exhibits interesting new underlying system architectures.

3. Poised for a resurgence

Every few years, it seems, sketch-based computing enjoys a resurgence of interest. Now, again, is one of those times. Several factors point to significant advances in the near future, and more widespread adoption. First, research continues to move the field forward incrementally, following the trend of the past fifteen or twenty years. Second, new hardware seems poised to radically change the ways in which people interact with displays. And third, the wider availability of versatile and powerful libraries enable developers of sketch-based interaction to harness sketching front ends more easily and effectively to back-end applications.

In their introduction to a 2007 special issue of IEEE Computer Graphics and Applications on sketch-based interaction Igarashi and Zeleznik identify two key research directions:

First, fundamental techniques for segmenting, recognizing, and parsing collections of ink strokes must be generalized and made more robust with better user models. Second, developers must go beyond redesign and actually reinvent their applications to leverage pen input's intrinsic capacity for rapid, direct, modeless 2D expression. [IZ07]

The larger SBIM community has long embraced the first research direction, and although much remains to be done, this work is well underway. Each year we see better and more refined techniques, informed by evaluations with users. The second research direction is more difficult because it relies on developers who may not see value in restructuring their applications to suit sketch-based interaction. Even so, here too progress is being made. Consider an application such as Crayon Physics [Pur07] : basically a physics engine with a simple sketching front end. These applications have become much easier to write because nowadays—quite differently even from a few years ago, a programmer can choose from a variety of physics engines with relatively easy-to-interface APIs. So more than before, a developer can concentrate on the sketching interface and rely on application libraries to provide the back-end simulation or other services.

3.1. The Marking Medium Matters

Sketching, like handwriting, is strangely a quite intimate act. Artists, designers, and engineers who draw have favored drawing instruments and media. Some prefer the soft HB pencil; others prefer a sharp felt-tip marker. (My father, a theoretical physicist, was quite particular about the Lindy ball-point pens he used to fill narrow-lined yellow pads with equations.) In light of this fine sensitivity to the marking medium and instruments, the physical media for computationally supported sketching has been impoverished: it is surprising that we have come as far as we have using light pens, digitizing tablets, and more recently, optically encoded paper.

At least as limiting have been the display technologies, especially when sensing and display are co-located. The still relatively low resolution of digitizing displays (compared with paper and physical ink) has been a discouraging obstacle. Even 'high resolution' LCD displays are far lower resolution than ink on paper. The Anoto technology addresses this, but is inherently not interactive responsive (though see [SGF*09] for using a small projector to augment optically encoded paper to provide interactivity). Small things like the screen-thickness separating the pen from the display, or the feel of a digitizing pen on a plastic surface do devalue the sketching experience. However, this may finally change. As flexible OLED displays and e-ink become more prevalent, the tactile experience of sketching on an interactive medium will be far better than before.

4. Why we need Sketch-based interaction and modeling

The promise is that we will be able to use sketching, drawing, diagramming, and other forms of marking with a stylus to communicate with computer software. The challenges of parsing, recognizing, interacting through, and understanding visual languages are similar to those in the natural language and speech recognition communities. Just as spoken language "in the wild" differs from textbook grammatical language, so drawing as people do it differs from formal visual language. And there is no one visual language, but many, and they vary in complexity, formality, purpose, and refinement. Lakin's vmacs system [Lak87] was among the first to attempt to cope with the variety of visual languages and the incremental refinement of sketches.

Certainly there are many practical benefits to addressing and resolving the challenges of sketch based interaction and modeling: the design, graphics, and media industries depend heavily on drawing, and being able to engage with artists and designers in their (still) preferred medium of choice is a tremendous advantage. On the other end of the spectrum, solving symbolic diagram recognition will aid scientists and engineers by providing computational support for a traditional form of visual reasoning. Notation certainly is a medium for thought.

Finally, in addition to the practical benefits, research on sketch-based interaction and modeling may shed some light on what, as Tversky asks, "do sketches say about thinking"? [Tve02]. Perhaps the most interesting aspect of this field of research is the hope that it will lead us, through the perhaps surprising avenue of computing, to address the dichotomy

between our views of thought and perception that Arnheim decried forty years ago.

References

[AD04] ALVARADO C., DAVIS R.: Sketchread: a multi-domain sketch recognition engine. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology* (New York, NY, USA, 2004), ACM, pp. 23–32.

[AG05] APITZ G., GUIMBRETIÈRE F.: Crossy: a crossing-based drawing application. In *ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), ACM, pp. 930–930.

[Arn69] ARNHEIM R.: *Visual Thinking*. University of California, Berkeley, 1969.

[Car09] CARLSON W.: A critical history of computer graphics and animation, accessed 1 June 2009. http://design.osu.edu/carlson/history/lesson2.html.

[Che96] CHENG P.: Scientific discovery with law encoding diagrams. *Creativity Research Journal 9*, 2 (1996), 145–162.

[DE64] DAVIS M. R., ELLIS T. O.: The RAND tablet: a man-machine graphical communication device. In *AFIPS '64 (Fall, part I): Proceedings of the October 27-29, 1964 Fall Joint Computer Conference, part I* (1964).

[Do02] DO E. Y.-L.: Drawing marks, acts, and reacts: Toward a computational sketching interface for architectural design. *Artif. Intell. Eng. Des. Anal. Manuf. 16*, 3 (2002), 149–171.

[EHS] ELLIS T. O., HEAFNER J. F., SIBLEY W. L.: *The Grail Project: An Experiment in Man-Machine Communications*. Tech. Rep. 5999-ARPA, RAND.

[HD06] HAMMOND T., DAVIS R.: Ladder: a language to describe drawing, display, and editing in sketch recognition. In *SIGGRAPH 2006 Courses* (New York, NY, USA, 2006), ACM, p. 27.

[HL06] HONG J. I., LANDAY J. A.: Satin: a toolkit for informal ink-based applications. In *SIGGRAPH 2006 Courses* (New York, NY, USA, 2006), ACM, p. 7.

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: A sketching interface for 3d freeform design. In *Proceedings of the SIGGRAPH 1999 Annual Conference on Computer Graphics* (1999), ACM, pp. 409–416.

[Ive79] IVERSON K.: Notation as a tool of thought, turing award lecture, 1979. http://www.acm.org/awards/ps/a1979-iverson_ps.pdf.

[IZ07] IGARASHI T., ZELEZNIK R.: Guest editors' introduction: Sketch-based interaction. *Computer Graphics and Applications, IEEE 27*, 1 (2007), 26–27.

[JGHD08] JOHNSON G., GROSS M. D., HONG J., DO E. Y.-L.: Computational support for sketching in design: A review. *Foundations and Trends in Human-Computer Interaction 2*, 1 (2008), 1–93.

[Lak87] LAKIN F.: Visual grammars for visual languages. In *National Conference on Artificial Intelligence (AAAI)* (1987), pp. 683–688.

[Lav05] LAVIOLA JR. J. J.: *Mathematical sketching: a new approach to creating and exploring dynamic illustrations*. PhD thesis, Providence, RI, USA, 2005. Adviser-Dam, Andries Van.

[LM95] LANDAY J. A., MYERS B. A.: Interactive sketching for the early stages of interface design. In *CHI '95 - Human Factors in Computing Systems* (Denver, Colorado, 1995), ACM Press, pp. 43–50.

[ML07] MASRY M., LIPSON H.: A sketch-based interface for iterative design and analysis of 3d objects. In *SIGGRAPH 2007 courses* (New York, NY, USA, 2007), ACM, p. 31.

[MvMC98] MORAN T. P., VAN MELLE W., CHIU P.: Spatial interpretation of domain objects integrated into a freeform electronic whiteboard. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology* (New York, NY, USA, 1998), ACM, pp. 175–184.

[PF07] PLIMMER B., FREEMAN I.: A toolkit approach to sketched diagram recognition. In *BCS-HCI '07: Proceedings of the 21st British CHI Group Annual Conference on HCI 2007* (Swinton, UK, 2007), British Computer Society, pp. 205–213.

[Pol45] POLYA G.: *How to Solve It*. Princeton University Press, Princeton, NJ, 1945.

[Pur07] PURHO P.: Crayon physics, 2007. http://www.crayonphysics.com.

[SFLM03] SAUND E., FLEET D., LARNER D., MAHONEY J.: Perceptually-supported image editing of text and graphics. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology* (New York, NY, USA, 2003), ACM, pp. 183–192.

[SGF*09] SONG H., GROSSMAN T., FITZMAURICE G. W., GUIMBRETIÈRE F., KHAN A., ATTAR R., KURTENBACH G.: Penlight: combining a mobile projector and a digital pen for dynamic visual overlay. In *Proc. Human Factors in Computing (CHI)* (2009), ACM, pp. 143–152.

[Sta97] STAHOVICH T. F.: Interpreting the engineer's sketch: A picture is worth a thousand constraints. In *AAAI Fall Symposium on Reasoning with Diagrammatic Representations II*. 1997, pp. 31–38.

[Tve02] TVERSKY B.: What do sketches say about thinking? In *2002 AAAI Spring Symposium Series – Sketch Understanding*, Stahovich T., Landay J., Davis R., (Eds.). AAAI, 2002.