Designs for the Future

Randall Davis, Aaron Adler, Christine Alvarado, Tracy Hammond, Rebecca Hitchcock, Michael Oltmans, Tevfik Metin Sezgin & Olya Veselova

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

http://www.ai.mit.edu

The Problem: Computer aided design tools built to date have typically shared three important characteristics: *(i)*, they are intended for use relatively late in the design process, *(ii)* they typically employ a style of interaction familiar to programmers but not necessarily intuitive to designers, and *(iii)* they capture the final product alone, recording almost none of the thinking, exploration, or rationale that motivated the endproduct. We are working to change this by designing and building a set of programs that are focused on the early, conceptual stage of design; that can take as input the familiar and natural kinds of interaction we routinely use with one another, namely sketching, gesturing, and talking; and that are capable of capturing the design process as well as the final product.

Motivation: We see considerable value in creating tools that aid at the earliest design stages and that can provide for the capture and subsequent dissemination of design rationale information. Such tools would assist those faced with redesign avoid the traditional task of intellectual archeology, i.e., attempting to determine why the design is as it is, typically tripping over the very same problems encountered and solved, but not recorded, by the orginal designers.

We likewise see considerable value in tools that enable natural forms of interaction. Where typing is appropriate for creating textual documents, design tasks frequently involve sketching (to indicate structural and spatial information), gesturing (to indicate movement and behavior), and verbal descriptions. Design tools that require typing even when information is spatial or gestural information often see far less use simply because the input effort is so large.

We also want designers to be able to communicate with their software tools by sketching out their conceptual designs with the same *imprecision* they are used to, using rough sketches that indicate relationship among components rather than precise details that cannot be known until later in the effort.

Previous Work: Our earliest work produced a program that began with a sketch and produced a model that we call a *behavior-ensuring parametric model*, a parametric model augmented with constraints from the sketch itself that ensure that any solution to the parametric model will produce the behavior specified for the device [8].

More recent work has focused on natural interaction, and has resulted in a number of systems. One permits sketches of simple mechanical devices to be understood, then animated [1]. Another demonstrates a degree of domain independence by using the same technology to enable sketching the class structure of an object-oriented program, then having the parsed version of the sketch turned into base code (by a software engineering tool) [7]. A third system is capable of understanding a description of intended device behavior expressed in a restricted vocabulary of speech and gestures [5].

Work by others similar in spirit to ours includes [3], [4], and [6], but none of these makes any significant effort to interpret the raw strokes of the sketch, either because the raw strokes are the desired outcome [4] or because the identity of the thing drawn is indicated by other means (e.g., speech) [6].

Approach: Our system is intended to act as an intelligent assistant, interpreting a sketch and any attendant gestures and verbal description, so that it understands what the designer is doing. That understanding should manifest in terms of the ability to hand off the interpreted sketch to another application (e.g., a simulator or code generator), the ability to infer design rationale, and eventually the ability to suggest variations on the design.

We use multiple modalities in part to ensure that interaction feels natural: some things (e.g., part placement) are far easier to draw than state in words, while others (e.g., material composition) are far easiier to state than draw. We also intend to employ the common strategy of mutual disambiguation, using information from each mode to help interpret the others.

The system uses a multi-level blackboard with multiple independent knowledge sources (KSs) [2] as the framework for interpretion of the sketch, speech and gestures (Fig. 1). Sketch input is in the form of strokes, which are interpreted as primitive lines, arcs, circles, etc.; these are aggregated to produce higher level objects such as polygons, which are further interpreted as objects in the domain (e.g., a motor or gear in a mechanical design). These objects may in turn be composed to produce more abstract objects in the domain (e.g., a latch). Our system will combine the blackboard with a novel form of Bayes Net to produce the best overall interpretation of the signals at any given moment.

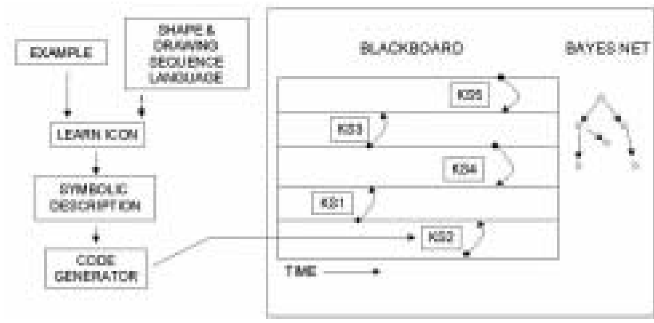

Figure 1: The architecture of our system.

Creating recognizers for shapes and gestures is currently a labor-intensive process that requires considerable coding skill. We are developing languages for describing shape, drawing sequence, and gestures, in order to make it possible to generate recognizers from these descriptions, rather than hand-crafting each one.

But writing even these descriptions can be a complex and demanding task, hence we are also working to provide the ability to learn them from one or two sketched examples. A code generator will use the resulting description as input and generate efficient code for a recognizer knowledge source to be used on the blackboard (Fig. 1).

Impact: Our work on natural interaction will help to to make interaction with software tools more like interaction with people, making those tools far more usable. Natural interaction is also a key to improving design tools: if we can make interaction with these tools feel like interaction with a person, we believe that will in turn make rationale capture *less* trouble than it is worth, a key element in making rational capture feasible.

This in turn may make a substantial change to the practice of design and re-design. Designs will carry a sizable body of machine-interpretable information in addition to the design itself; that information will be used to check design modifications to ensure that the new design does not defeat some goal accomplished by the original design.

Research Support: This work is supported by MIT's Project Oxygen.

References:

[1] Christine Alvarado and Randall Davis. Resolving ambiguities to create a natural sketch based interface. *Proceedings of IJCAI-2001*, 2001.

[2] Erman, Hayes-Roth, Lesser, and Reddy. The Hearsay-II speech understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, 12(2), 1980.

[3] Forbus, Ferguson, and Usher. Towards a computational model of sketching. *Proc IUI 01*, 2001.

[4] James Landay and Brad Myers. Sketching interfaces. *IEEE Computer*, 34:56–64, 2001.

[5] Michael Oltmans and Randall Davis. Naturally conveyed explanations of device behavior. *Workshop on Perceptive User Interfaces*, 2001.

[6] Oviatt and et al. Designing the user interface. *Human Computer Interaction*, 15(4):263–322, 2000.

[7] Hammond T and Davis R. Tahuti: A geometrical sketch recognition system for uml class diagrams. *AAAI Spring Symposium on Sketch Understanding*, 2002.

[8] Stahovich T, Davis R, and Shrobe H. Generating multiple designs from a single sketch. *Artificial Intelligence*, 104(1-2):211–264, 1998.