Natural Editing and Recognition of UML Class Diagrams

Tracy Hammond, Kalani Oshiro, and Randy Davis

The Problem: Sketching is a natural and integral part of software design. Software developers use sketching to aid in the brainstorming of ideas, visualizing programming organization, and understanding of requirements. Unfortunately, when it comes to coding the system, the drawings are left behind. We see natural sketch recognition as a way to bridge that gap.

Motivation: We have selected UML diagrams because they are a de facto standard for depicting software applications. Within UML we focused on class diagrams, first because of their central role in describing program structure, and second because many of the symbols used in class diagrams are quite similar, and hence, they offer an interesting challenge for sketch recognition.

Previous Work: Work at Berkeley by Hse [3] has shown that users prefer a single-stroke sketch-based user interface to a mouse-and-palette based tool for UML design.

One company [2] has developed a gesture based diagramming tool, Ideogramic UML,™which allows users to sketch UML diagrams. The tool is based on a graffiti-like implementation and requires users to draw each gesture in one stroke, and in the direction and style as specified by the user manual. As a consequence, some of the gestures drawn only loosely resemble the output glyph. For example, φ is the stroke used to indicate an actor, drawn by the system as a stick figure.

Work at Queen's University has developed a system to recognize sketches of UML diagrams using a distance metric [4]. Each glyph (square, circle, or line) is classified based on the total stroke length compared to the perimeter of its bounding box (e.g., if the stroke length is approximately equal to the perimeter of the bounding box, it is classified as a square). The shape of the stroke is not considered.

Approach: Our goal is a system in which the users can sketch UML diagrams on a tablet or whiteboard in the same way they are drawn on paper, and have the diagrams recognized by the computer. Thus far, we have created a natural sketch recognition environment for UML (Unified Modeling Language) [1]. Our system differs from graffiti-based approaches to this task in that it recognizes objects by how they look, not by how they are drawn.

While sketching, the sketcher can seamlessly switch between two views: the interpreted designs or the original strokes. Editing commands operate identically in the two views. Some sketchers become distracted by the sketch recognition process when it replaces their strokes with the interpreted version. The alternate views allow the users to sketch as they are more comfortable.

Impact: The recognition system is connected to Rational Rose, a popular CASE tool, and can output diagrams directly to Rational Rose. This enables the user to take full advantage of the benefits of a CASE tool, such as the ability to auto-generate code stubs, while still retaining the natural feeling of a sketch tool.

Future Work: Future system enhancements include allowing the user to sketch more detail about a program. For instance, we plan to add the ability to recognize multiplicity relationships by noting properties sketched around associations.

Research Support: This work is supported in part by the Ford/MIT Alliance and by the MIT Oxygen Collaboration.

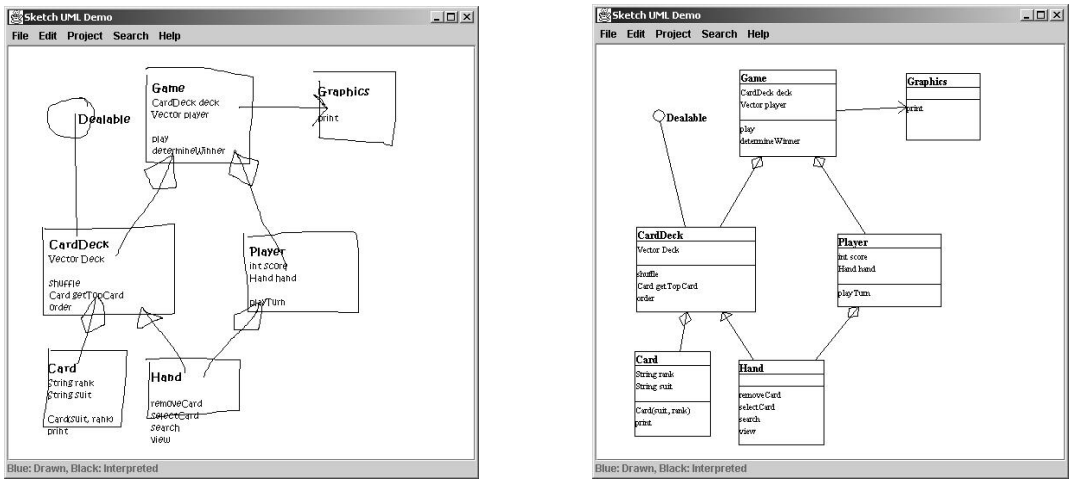

Figure 1: Hand drawn UML class diagram and its accompanying interpreted diagram

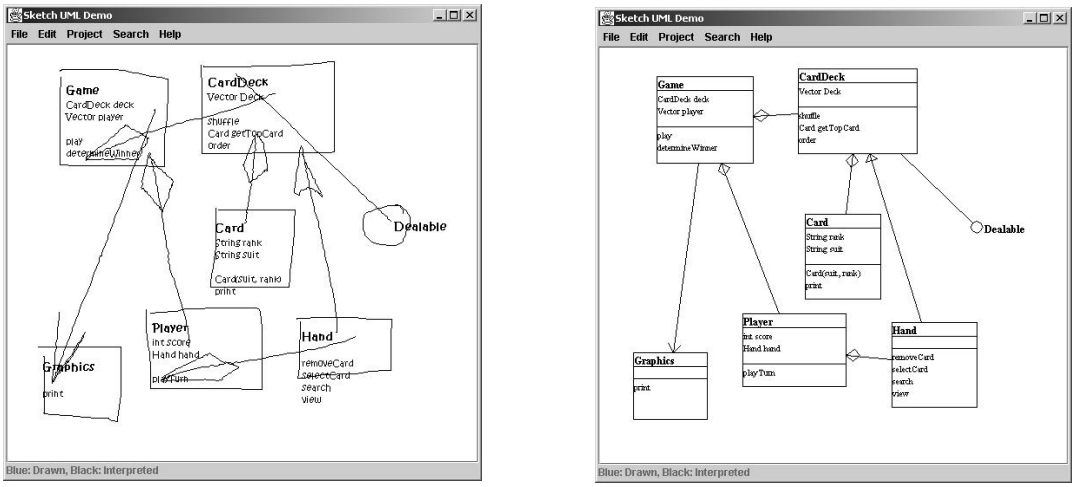

Figure 2: The same class diagram as in Figure 1 with the classes rearranged suing the system's editing ability and the interpreted diagram.

References:

[1] Sinan Si Alhir. *UML in a Nutshell: a desktop quick reference*. O'Reilly & Associates, Inc., Cambrigde, MA, 1998.

[2] Christian Heide Damm, Klaus Marius Hansen, and Michael Thomsen. Tool support for cooperative object-oriented design: Gesture based modeling on an electronic whiteboard. In *CHI 2000*. CHI, April 2000.

[3] Heloise Hse, Michael Shilman, A. Richard Newton, and James Landay. Sketch-based user interfaces for collaborative object-oriented modeling. Berkley CS260 Class Project, December 1999.

[4] Edward Lank, Jeb S. Thorley, and Sean Jy-Shyang Chen. An interactive system for recognizing hand drawn UML diagrams. In *Proceedings for CASCON 2000*, 2000.