Multi-Domain Sketch Recognition

Tracy Hammond HAMMOND@AI.MIT.EDU
Metin Sezgin MTSEZGIN@AI.MIT.EDU
Olya Veselova OLYAV@AI.MIT.EDU
Aaron Adler CADLERUN@MIT.EDU
Michael Oltmans MOLTMANS@AI.MIT.EDU
Christine Alvarado CALVARAD@AI.MIT.EDU
Rebecca Hitchcock REBECCA8@MIT.EDU
MIT Artificial Intelligence Laboratory, 200 Technology Square, Cambridge MA, 02139 USA

Abstract

In this paper, we describe a new framework for multi-domain sketch recognition which is being developed by the Design Rationale Group at the MIT AI laboratory. The framework uses a blackboard architecture for recognition in which the knowledge sources are a combination of domain-independent and domain-specific recognizers. Domain-specific recognizers are automatically generated from the domain description which is written using the domain description language syntax. Domain descriptions can be automatically generated by a system that learns shape descriptions from a drawn example.

1. Introduction

Pervasive computing strives to provide natural access to a wide variety of traditional applications. Sketching, for example, is a natural interface for design tools at the conceptual design stage. Thus, a number of sketch recognition design applications have been built for the Oxygen platform including Tahuti (Hammond & Davis, 2002), Assist (Alvarado, 2000), and Assistance (Oltmans, 2000). To date, sketch recognition systems have been domain-specific, with the recognition details of the domain hard-coded into the system, making the design process of a new sketchable user interface difficult. A domain-independent recognition system that could be used for all domains could simplify the development process. However, to function properly the recognition system must also have domain-specific information, such as which shapes are significant in the domain and what do they look like.

We propose a multi-domain recognition framework in which the recogition system uses a blackboard architecture with domain-independent recognizers. Domain-specific information is described in a domain description text file written in the domain description language syntax. The domain description is compiled into recognizers for use by the blackboard. The domain description can be written by hand or generated automatically by a system that learns shape descriptions from drawn examples.

1.1 Related Work

The Electronic Cocktail Napkin Project (ECNP) (Gross & Do, 1996) allows users to define domain shapes by drawing them. The ECNP does not incorporate helpful top-down contextual information in its recognition process nor does it handle ambiguity. The ECNP language cannot describe non-shape information such as editing behavior. The low level recognizers have stroke order and direction requirements and need to be trained a multitude of times rather than recognizing based on shape.

2. Domain-Specific Recognizers

A domain description text file (Figure 1-3), written using the syntax of the domain description language (Figure 1-1) (Hammond, 2002), specifies the domain-specific information needed by the recognition system. The language consists of pre-defined shapes, constraints, and editing behaviors, as well as a syntax for specifying a domain description. Shapes in a domain can be defined hierarchically, and although the language is primarily based on shape, the domain description can include any type of information that would be helpful to the recognition process, such as stroke order or direction. It can also specify editing behaviors and display information.

Domain descriptions can be written by hand or generated automatically by a system that learns shape descriptions from one or two drawn examples and a couple of questions (Figure 1-2) (Veselova, 2002). The system uses knowledge about human perception to determine which properties and constraints are relevant.

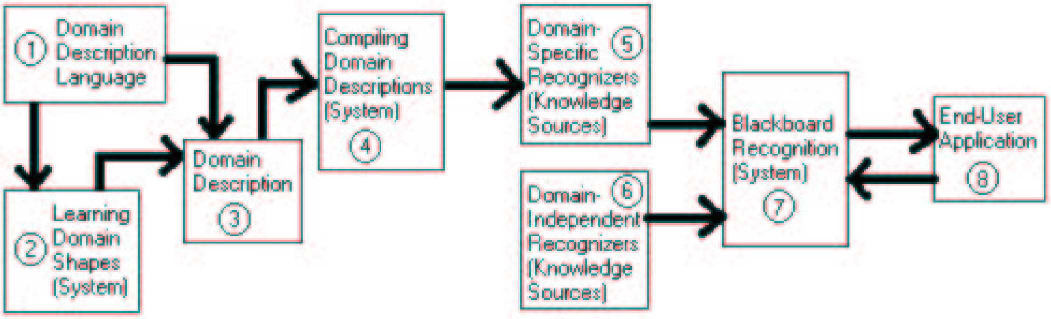
Domain Description Language ①
Learning Domain Shapes (System) ②
Domain Description ③
Compiling Domain Descriptions (System) ④
Domain-Specific Recognizers (Knowledge Sources) ⑤
Domain-Independent Recognizers (Knowledge Sources) ⑥
Blackboard Recognition (System) ⑦
End-User Application ⑧

Figure 1. Multi-Domain Sketch Recognition Framework.

Domain-specific recognizers (Figure 1-5) are generated automatically from the domain description by a compiler (Figure 1-4) (Sezgin, 2002). The recognizers are in the form of human readable recognition code and data structures. They create templates to help identify partial recognitions which can resume recognition when more data is available.

3. Domain-Independent Recognizers

Domain-independent recognizers (Figure 1-6) can be used for low-level stroke recognition (Sezgin et al., 2001). In low-level stroke recognition, strokes are processed and broken down into lines, circles, poly-lines, and complex shapes. Corners are found using speed and curvature data.

Domain-independent recognizers also may process speech events. Some things are more easily specified by drawing and others by speech. Speech recognizers allow users to communicate sketch information more naturally.

4. Blackboard Recognition Architecture

The recognition system (Figure 1-7) is based on blackboard architecture in which available information is posted on a blackboard (Alvarado et al., 2002). Knowledge sources, which are in our case domain-independent and domain-specific recognizers, search the board for information that they can process. The system integrates top-down and bottom-up recognition to combine both contextual and shape information. The system handles ambiguity through the use of a Baysian network.

5. Conclusions

We propose a multi-domain sketch recognition framework. The recognition system uses a blackboard architecture in which information is posted to a blackboard. Domain-independent and domain-specific recognizers process the information on the blackboard. Domain-specific information is written to a domain description text file using the domain description language syntax. The domain description can be written by hand or it can be automatically generated using a system that learns shape descriptions from drawn examples.

Future work should focus on system implementation and comparison of the recognition accuracy of a hard-coded sketchable user interface to the recognition accuracy of a sketchable user interface generated automatically from a domain description.

Acknowledgements

This work describe here is advised by Randall Davis(Davis, 2002). This work is supported in part by the MIT Project Oxygen.

References

Alvarado, C. (2000). A natural sketching environmant: Bringing the computer into early stages of mechanical design. Master's thesis, MIT.

Alvarado, C., Oltmans, M., & Davis, R. (2002). A framework for multi-domain sketch recognition. *Sketch Understanding, Papers from the 2002 AAAI Spring Symposium*, 1–8.

Davis, R. (2002). Sketch understanding in design: Overview of work at the mit ai lab. *Sketch Understanding, Papers from the 2002 AAAI Spring Symposium*, 24–31.

Gross, M. D., & Do, E. Y.-L. (1996). Demonstrating the electronic cocktail napkin: a paper-like interface for early design. *'Common Ground' appeared in ACM Conference on Human Factors in Computing (CHI)*, 5–6.

Hammond, T. (2002). A domain description language for sketch recognition. *MIT Student Oxygen Workshop*.

Hammond, T., & Davis, R. (2002). Tahuti:a geometrical sketch recognition system for uml class diagrams. *AAAI Spring Symposium on Sketch Understanding*, 59–68.

Oltmans, M. (2000). Understanding naturally conveyed explanations of device behavior. Master's thesis, MIT.

Sezgin, M. (2002). Generating domain specific sketch recognizers from object descriptions. *MIT Student Oxygen Workshop*.

Sezgin, T. M., Stahovich, T., & Davis, R. (2001). Sketch based interfaces: Early processing for sketch understanding. *Proceedings of 2001 Perceptive User Interfaces Workshop (PUI'01)*.

Veselova, O. (2002). Perceptually based learning of shape descriptions from one example. *MIT Student Oxygen Workshop*.