User Interface Softbots

Robert St. Amant and Luke S. Zettlemoyer
Department of Computer Science
North Carolina State University
EGRC-CSC Box 7534
Raleigh, NC 27695–7534
stamant@csc.ncsu.edu

Softbots in the interface

Human-computer interaction (HCI) and artificial intelligence (AI) share a long history of research. Concepts such as problem spaces, goals and operators, rationality, and computational models of cognition have significantly influenced research directions in both fields. Recently the concept of *agents* has sparked a common interest among AI and HCI researchers. Our demonstration focuses on interface agents, those that assist the user in the rich, often complex environment of the graphical user interface (Maes 1994; Lieberman 1995).

Our general interest lies in the interaction between agents and their environments. Conventional interface agents interact with other applications through an application programming interface (API) or access to source code. We have developed a novel class of agents we call interface softbots, or *ibots*, that control interactive applications through the graphical user interface, as human users do (Zettlemoyer & St. Amant 1999; Zettlemoyer, St. Amant, & Dulberg 1999). Our ibots are based on a programmable substrate that provides sensors and effectors for this purpose. Sensor modules take pixel-level input from the display, run the data through image processing algorithms, and build a structured representation of visible interface objects. Effector modules generate mouse and keyboard gestures to manipulate these objects. These sensors and effectors act as the eyes and hands of an artificial user, controlled by an external cognitive system. Together the sensors, effectors, and controller provide a general-purpose means of managing interactive applications, through the same medium as a real user.

System components

Conceptually, we can break down ibot functionality into three components, as we might do with any physically realized robot: perception, action, and control.

Perception

Ibot perception relies on conventional image processing techniques. Object identification follows a three-stage process of segmentation, feature computation, and interpretation. The process starts by examining the contents of the screen buffer, at the pixel level, as shown in Figure 1. Segmentation breaks the image into pixel groups of different colors. Bottom-up feature computation follows, to associate properties such as color, bounding box points, area, and perimeter with each group. Finally, interpretation rules combine these features to identify interface objects in top-down fashion.

The process recognizes all familiar user interface controls in Microsoft Windows, including buttons, scroll bars (including the scroll box, scroll arrows, and background regions), list boxes, menu items, check boxes, radio buttons and application windows. It can even parse text, doing a simple form of optical character recognition, for the standard system typeface.

Action

Ibot effectors insert events into the operating system's event queue, resulting in actions indistinguishable from user-generated events. Primitive events include `move-mouse`, `mouse-down`, `mouse-up`, `key-down`, and `key-up`. This allows ibots to select icons, click buttons, pull down menus, turn on radio buttons, and carry out other standard, familiar operations that human users are capable of.

Control

Control of ibot sensors and effectors is through standard AI planning techniques. We find that the assumptions made by theoretically motivated planning systems (e.g., Graphplan and its relatives (Blum & Furst 1997; Weld 1999)) are

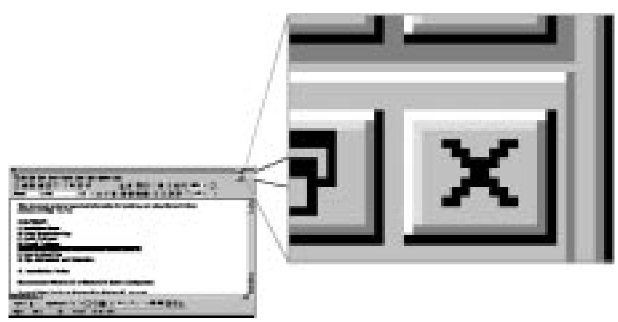

Figure 1: Source data for visual processing

closely matched by design features built into graphical user interfaces. Planners often abstract away the continuous, uncertain, dynamic, and unobservable properties of an environment, such that it becomes discrete, deterministic, static, and accessible—properties associated with broad classes of modern graphical user interfaces (St. Amant 1999). In our most recent work (St. Amant & Zettlemoyer 2000) we have developed a very simple hierarchical planner to control the perception and action components of an ibot.

The planner can direct an ibot to take a wide variety of action sequences, ranging from selecting objects to choosing pulldown menu items to more complex, domain-specific activities in the interface.

Discussion

We have built ibots for a number of interactive applications, including a word processor (St. Amant 2000), an illustration package (St. Amant & Zettlemoyer 2000), the Windows OS interface (Dulberg, St. Amant, & Zettlemoyer 1999), and even Microsoft Solitaire (Zettlemoyer & St. Amant 1999). A generic ibot architecture is shown in Figure 2. The implementation is in C++ and Common Lisp; it bears some similarity at the most basic level to the Java Robot class. The work has significant limitations, but progress has been surprisingly rapid.

Our long-term goal, as stated, is the development of ibots that can solve real-world problems of significant complexity, those that a user might be interested in to turn over to an automated system. We have only taken a few initial steps in this direction. Nevertheless, we have found ibots to be a flexible, powerful vehicle for agents research. Our preliminary work has given us insights in areas such as intelligent interaction mechanisms (Dulberg, St. Amant, & Zettlemoyer 1999), programming by demonstration (St. Amant *et al.* 2000), user interface evaluation (Zettlemoyer, St. Amant, & Dulberg 1999), programmable user models (St. Amant, Riedl, & Zettlemoyer 2000), and most importantly the relationship between AI planning and the user interface (St. Amant 1999; St. Amant & Zettlemoyer 2000; Zettlemoyer & St. Amant 1999).

We believe that our work on ibots sets the stage, in the longer term, for agents that can use interactive applications with all the facility of human users. Consider the human-oriented environments that agents can act in today: robots in offices, hallways, or on the road, or softbots moving through file systems and over the Internet. Agents are often at some disadvantage with respect to their sensing and effecting capabilities in these environments, in comparison with human agents. In contrast, ibots in the restrictive environment of a user interface have access to all the same information and all the same actions that human users have, with little or no degradation in quantity or quality—the only difference between users and agents in this environment is the knowledge and cognitive processing power they bring to bear. Our work will level the playing field for humans and agents solving real-world problems in an extremely powerful and flexible environment.

References

Blum, A., and Furst, M. 1997. Fast planning through planning graph analysis. *Artificial Intelligence* 90:281–300.

Dulberg, M. S.; St. Amant, R.; and Zettlemoyer, L. 1999. An imprecise mouse gesture for the fast activation of controls. In Proceedings of INTERACT '99, 375–382.

Lieberman, H. 1995. Letizia : An agent that assists web browsing. In Proceedings of the International Joint Conference on Artificial Intelligence, 924–929.

Maes, P. 1994. Agents that reduce work and information overload. *Communications of the ACM* 37(7):31–40.

St. Amant, R., and Zettlemoyer, L. S. 2000. The user interface as an agent environment. Forthcoming.

St. Amant, R.; Lieberman, H.; Potter, R.; and Zettlemoyer, L. S. 2000. Visual generalization in programming by example. *Communications of the ACM* 43(3):107–114.

St. Amant, R.; Riedl, M. O.; and Zettlemoyer, L. S. 2000. A practical perception substrate for cognitive modeling in HCI. Forthcoming.

St. Amant, R. 1999. User interface affordances in a planning representation. *Human Computer Interaction* 14(3):317–354.

St. Amant, R. 2000. Interface agents as surrogate users. Forthcoming.

Weld, D. S. 1999. Recent advances in AI planning. *AI Magazine* 20(2):93–123.

Zettlemoyer, L., and St. Amant, R. 1999. A visual medium for programmatic control of interactive applications. In CHI '99 (ACM Conference on Human Factors in Computing), 199–206.

Zettlemoyer, L.; St. Amant, R.; and Dulberg, M. S. 1999. Ibots: Agent control through the user interface. In Proceedings of the Fifth International Conference on Intelligent User Interfaces, 31–37.

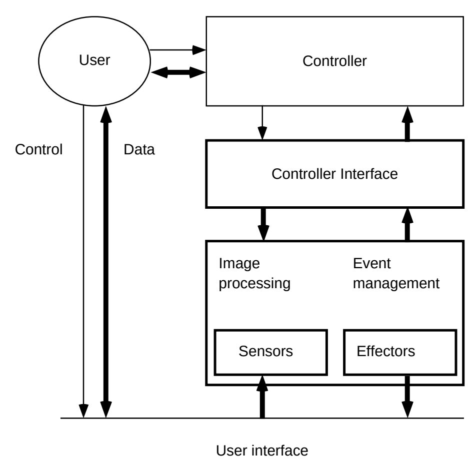

Figure 2: A generic ibot architecture