

Picture-driven computing (w/ Video)

January 20 2010, by Larry Hardesty



Graphic: Christine Daniloff

Until the 1980s, using a computer program meant memorizing a lot of commands and typing them in a line at a time, only to get lines of text back. The graphical user interface, or GUI, changed that. By representing programs, program functions, and data as two-dimensional images -- like icons, buttons and windows -- the GUI made intuitive and spatial what had been memory intensive and laborious.

But while the GUI made things easier for computer users, it didn't make them any easier for computer programmers. Underlying GUI components is a lot of [computer code](#), and usually, building or customizing a program, or getting different programs to work together, still means manipulating that code. Researchers in MIT's Computer Science and Artificial Intelligence Lab hope to change that, with a

system that allows people to write programs using screen shots of GUIs. Ultimately, the system could allow casual computer users to create their own programs without having to master a programming language.

The system, designed by associate professor Rob Miller, grad student Tsung-Hsiang Chang, and the University of Maryland's Tom Yeh, is called Sikuli, which means "God's eye" in the language of Mexico's Huichol Indians. In a paper that won the best-student-paper award at the Association for Computing Machinery's User Interface Software and Technology conference last year, the researchers showed how Sikuli could aid in the construction of "scripts," short programs that combine or extend the functionality of other programs. Using the system requires some familiarity with the common scripting language Python. But it requires no knowledge of the code underlying the programs whose functionality is being combined or extended. When the programmer wants to invoke the functionality of one of those programs, she simply draws a box around the associated GUI, clicks the mouse to capture a screen shot, and inserts the screen shot directly into a line of Python code.

Suppose, for instance, that a Python programmer wants to write a script that automatically sends a message to her cell phone when the bus she takes to work rounds a particular corner. If the transportation authority maintains a web site that depicts the bus's progress as a moving pin on a Google map, the programmer can specify that the message should be sent when the pin enters a particular map region. Instead of using arcane terminology to describe the pin, or specifying the geographical coordinates of the map region's boundaries, the programmer can simply plug screen shots into the script: when this (the pin) gets here (the corner), send me a text.

"When I saw that, I thought, 'Oh my God, you can do that?'" says Allen Cypher, a researcher at IBM's Almaden Research Center who specializes

in human-computer interactions. “I certainly never thought that you could do anything like that. Not only do they do it; they do it well. It’s already practical. I want to use it right away to do things I couldn’t do before.”

In the same paper, the researchers also presented a Sikuli application aimed at a broader audience. A computer user hoping to learn how to use an obscure feature of a [computer program](#) could use a screen shot of a GUI — say, the button that depicts a lasso in Adobe Photoshop — to search for related content on the web. In an experiment that allowed people to use the system over the web, the researchers found that the visual approach cut in half the time it took for users to find useful content.

In the same way that a programmer using Sikuli doesn’t need to know anything about the code underlying a GUI, Sikuli doesn’t know anything about it, either. Instead, it uses computer vision algorithms to analyze what’s happening on-screen. “It’s a software agent that looks at the screen the way humans do,” Miller says. That means that without any additional modification, Sikuli can work with any program that has a graphical interface. It doesn’t have to translate between different file formats or computer languages because, like a human, it’s just looking at pixels on the screen.

In a new paper to be presented this spring at CHI, the premier conference on human-computer interactions, the researchers describe a new application of Sikuli, aimed at programmers working on large software development projects. On such projects, new code accumulates every day, and any line of it could cause a previously developed GUI to function improperly. Ideally, after a day’s work, testers would run through the entire application, clicking virtual buttons and making sure that the right windows or icons still pop up. Since that would be prohibitively time consuming, however, broken GUIs may not be

detected until the application has begun the long and costly process of quality assurance testing.

The new Sikuli application, however, lets programmers create scripts that automatically test an application's GUI components. Visually specifying both the GUI and the window it's supposed to pull up makes writing the scripts much easier; and once written, they can be run every night without further modification.

But the new application has an added feature that's particularly heartening to non-programmers. Like its predecessors, it allows users to write their scripts — in this case, GUI tests — in Python. But of course, writing scripts in Python still requires some knowledge of Python — at the very least, an understanding of how to use commands like “dragDrop” or “assertNotExist,” which describe how the GUI components should be handled.

The new application gives programmers the alternative of simply recording the series of keystrokes and mouse clicks that define the test procedure. For instance, instead of typing a line of code that includes the command “dragDrop,” the programmer can simply record the act of dragging a file. The system automatically generates the corresponding Python code, which will include a cropped screen shot of the sample file; but if she chooses, the programmer can reuse the code while plugging in screen shots of other GUIs. And that points toward a future version of Sikuli that would require knowledge neither of the code underlying particular applications nor of a scripting language like Python, giving ordinary computer users the ability to intuitively create programs that mediate between other applications.

Provided by Massachusetts Institute of Technology

Citation: Picture-driven computing (w/ Video) (2010, January 20) retrieved 13 June 2024 from <https://phys.org/news/2010-01-picture-driven-video.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.