

April 2007 \$9.95 www.StickyMinds.com

BETTER SOFTWARE

REQUIREMENTS SOS
Don't let poor specs
sink your project

LIGHTEN UP
Test more with less

The Print Companion to

StickyMinds.com

THE ★ RIGHT STUFF

*Four Small Steps for Testers
One Giant Leap for Risk Mitigation*

The Magic 8 Ball of Testing

by Michael Bolton

Over the years, there has been some progress on automating parts of the development and testing process. The NUnit family of test automation frameworks has caused a revolution in unit testing and has made the world safe for refactoring. FIT and FitNesse help with integration-level tests, and Web automation tools, such as Perl's WWW::Mechanize, Watir, and Selenium, show a lot of promise in aiding the testing effort at the highest levels of the application. All of these tools provide support for tests that we've thought of already, but none of them helps with the job of defining and refining requirements such that programs are automatically more testable.

To my surprise, this spring I was introduced to such a tool. "It's a little joke—can you spot the anagram?" asked Dr. Ralf Piolo. "It's called oClear. That's an anagram for 'oracle'—an oracle is a principle or mechanism for recognizing a problem, so a really clear requirement statement is one of the best oracles." Dr. Piolo had invited me to his lab at the University of Bala in Ontario, Canada, to see a demonstration of oClear. The program is a high-level requirements tool that he has been working on with several of his graduate students. The tool accepts requirements in natural language, parses them, and provides a complete analysis and a suite of improvement suggestions as output, also in natural language.

The interface of the prototype that Dr. Piolo showed me was



very lean and clean—a single dialogue, with tabs for the various analysis features. The tool recognizes variables, ambiguities, equivalence classes and boundary values, and project-costing problems. It also suggests test script ideas and identifies excep-



2006 SCOTT SHEARER

tations as to where bugs will be found.

Dr. Piolo encouraged me to submit a sample requirement. I proposed a requirement that I thought might fool the system: "Send a reminder notice to every customer who has not paid in thirty days." I entered it into the requirements parser. "Good," said Dr. Piolo. "Let's start by clicking the tab labeled Disambiguate." In the output window, the application reported:

Actor Ambiguity: oClear suggests that only those customers who owe money should receive reminder notices.

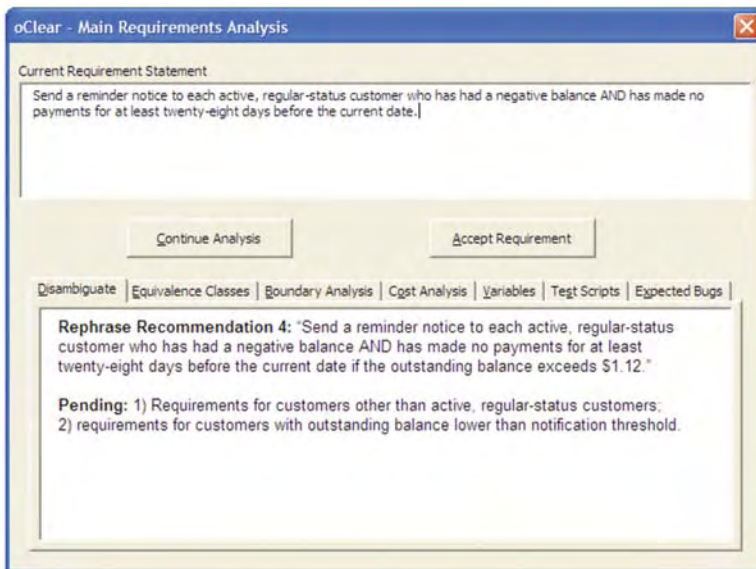
Rephrase Recommendation 1: "Send a reminder notice to every customer who has had an outstanding balance AND has made no payment for at least thirty days."

I was astonished. Not only had oClear identified an ambiguity but it also had proposed a reasonable solution for getting around it—and had put the solution into a well-structured sentence to boot.

"Press Continue Analysis, and then press Disambiguate again," said Dr. Piolo. I did.

Actor Ambiguity: oClear suggests that the term "every" is ambiguous, as some customers will fall into exceptional categories, such as "privileged," "inactive," or "bankrupt."

Rephrase Recommendation 2: "Send a reminder notice to each active, regular-status customer who has had a negative balance AND has made no payments for at least thirty days."



Warning: Customer special-status categories must be defined.

This was really impressive. I wanted to press Disambiguate again, but Dr. Piolo wanted to show me more of the application. "Try pressing Boundary Analysis," he suggested.

Date Ambiguity: oClear suggests that assessment periods of more than twenty-eight days fail over February/March boundary.

Date Recommendation 1: Alter requirement to calculate payment due based on the []th day of every month (or first business day thereafter).

Date Recommendation 2: Alter requirement to calculate payment due based on four-week period. (Choose day of week [Monday] and start date [yyyy/mm/dd]).

Date Recommendation 3: Alter requirement to calculate payment due based on [day of week] in 4/4/5 week intervals. (Choose start date: [yyyy/mm/dd]).

"I notice that oClear put the year first," I said.

"Yes," said Dr. Piolo. "A lot of our research funding comes from international companies. Here in Canada, dates are formatted day/month/year, and in the States, they are formatted month/day/year. OClear expresses the requirement in a way that reduces the possibility of confusion, so that products can be internationalized easily. Try accepting the second one." I did. The requirement now said:

Rephrase Recommendation 3: "Send a reminder notice to each active, regular-status customer who has had a negative balance AND has made no payments for at least twenty-eight days before the current date."

Pending: 1) Requirements for customers other than active, regular-status customers

I was impressed. "OK," said Dr. Piolo. "Now click the Cost Analysis button."

Cost Analysis: [Send notice] When sending notifi-

cations, ensure that the value of sending notification exceeds the burdened cost.

Burdened cost of email: \$0.04

Burdened cost of surface mail (based on bulk rate): \$1.12

"Where do those figures come from?" I asked.

"That's probably from the Direct Marketing Association survey data," replied Dr. Piolo. "We have a fair pool of data from which to draw. If it turns out to be more or less expensive for a given organization, you can change it. Just use this for now."

Rephrase Recommendation 4: "Send a reminder notice to each active, regular-status customer who has had a negative balance AND has made no payments for at least twenty-eight days before the current date if the outstanding balance exceeds \$1.12."

Pending: 1) Requirements for customers other than active, regular-status customers; 2) requirements for customers with outstanding balance lower than notification threshold

"This is amazing," I said. "I'd love to show this around. Can I do that?"

"Sure," said Dr. Piolo. "OClear is Web-enabled. You can use it from any browser."

"Great!" I said. "I'm having dinner with one of my consulting clients next Sunday night. How do I get into the system?"

Dr. Piolo frowned. "Sunday night? That's likely to be a problem. Can it wait until Monday?"

"Umm . . . probably," I said, "but why?"

"My students aren't in on Sundays; the lab is closed."

"Why does that make a difference?"

"OClear needs someone at the other end to make the suggestions that clarify the requirements."

"It does?"

"Well, of course it does," said Dr. Piolo. "You didn't think that this sort of analysis could be done by a computer, did you?"

So I reluctantly rescheduled my appointment for April 2. **{end}**

Michael Bolton lives in Toronto and teaches heuristics and exploratory testing in Canada, the United States, and other countries as part of James Bach's Rapid Software Testing course. Michael is also program chair for the Toronto Association of System and Software Quality. He is a regular contributor to Better Software magazine. Contact Michael at mb@developsense.com.



Have you ever been taken in (for a while, at least) by a product claim that was too good to be true?

▼

Follow the link on the StickyMinds.com homepage to join the conversation.