

HOT RESEARCH

Exercises in Programming Style

Back in the 1940s, the French writer Raymond Queneau wrote an interesting book with the title *Exercises in Style* featuring 99 renditions of the exact same story, each written in a different style. It is a master piece of writing technique, as it illustrates the many different ways a story can be told. The story being fairly trivial

and always the same, the book shines the spotlight on form, rather than on content. As such, the book is of great value to all those interested in the technical aspects of the Art

of Writing, and in how the decisions we make in telling a story affect the perception of that story.

Over the years, as an instructor of many programming-intensive courses, ISR professor **Crista Lopes** noticed that many students have a hard time understanding the many different ways of writing programs and of designing systems, in general. They have been trained in one, at most two, programming languages, so they understand only the styles that are encouraged by those languages. It's not their fault. Looking at the history of programming languages, and the lack of pedagogical material on style in most CS programs, one hardly gets exposed to the issue until after an enormous amount of experience is accumulated. Even then, style is seen as this intangible property of programs that remains elusive to explain to others – and over which many technical arguments ensue.

In order to give programming styles the

RESEARCH BRIEFS

Prof. **Richard N. Taylor** delivered two keynote talks recently. The first, “The Role of Architectural Styles in Successful Software Ecosystems,” was given at the 17th International Software Product Line Conference (SPLC) in Tokyo, Japan in August. The second, “Achieving Success in Open Software Ecosystems: The Role of Architectural Styles,” was given at the 20th Asia-Pacific Software Engineering Conference (APSEC) in Bangkok, Thailand in December.

Prof. **André van der Hoek** has been named a Distinguished Scientist by the ACM. This prestigious honor is given to ACM members based on their individual contributions and their singular impacts on the field of computing.

Prof. **André van der Hoek** has been awarded \$300,000 by the National Science Foundation (NSF) for his research on Social-Technical Code Search.

The ISR website has been redesigned by programmer/analyst **Kari Nies**. Be sure to visit us at <http://isr.uci.edu> and to like ISR on Facebook, add ISR on Google+, follow ISR on Twitter, and subscribe to ISR's channel on YouTube.

More Research Briefs on page 5.

proper due, and inspired by Queneau, Lopes decided to embark on the project of writing the exact same computational task in as many styles as she has come across over the years. This project involves two artifacts: a collection of code examples currently hosted in github (<https://github.com/crista/exercises-in-programming-style>), and a textbook that explains 32 of those code examples line by line. The collection is public, and several other people have already contributed to it – among others, there is a contribution from Peter Norvig, Director of Research at Google! The textbook targets courses on advanced programming / Software Engineering, and will be available in Spring 2014.

So what is style? In Queneau's circle of intellectuals, style was nothing but the consequence of *creating under constraints*,

often based on mathematical concepts such as permutations or lipograms. These constraints were used as a means to create something new and intellectually interesting besides the story itself. This idea of associating style with constraints, however, precedes Queneau by several centuries; in all corners of the Arts, style – operationally defined by a set of constraints of what to do and what not to do – has always played a central role in creativity.

In this project, too, programming style is what results from writing programs under a set of constraints. Constraints can come from external sources or they can be self-imposed; they can capture true challenges of the environment or they can be artificial; they can come from past experiences and measurable data or they can come from personal preferences. Independent of their origin nature, constraints are the seeds of style. By honoring different constraints, we can write a variety of programs that are virtually identical in terms of *what* they do, maybe even have the exact same output for a given input, but that are radically different in terms of *how* they do it. An advanced programmer needs not just to be able to write correct programs; he/she

IN THIS ISSUE

- 2-Message from the Director
- 3-Improving Teaching through Games
- 3-6-Student News
- 4-ISR Sponsorship
- 5-Brazilian Research Team at UCI
- 5-Research Briefs
- 6-Microtasking Software Development
- 6-Technical Reports
- 8-ISR Events

MESSAGE FROM THE DIRECTOR

HealthCare.gov: the Triumph of Process over Product

Spoken to any software developers about HealthCare.gov? Everyone that I've spoken with, of whatever political persuasion, cringes when the topic comes up. It seems that every conceivable principle of software development was, and is being, violated. In a perverse way it is the ultimate source of compelling illustrations of what not to do. Whether the issues are requirements, design, testing, deployment, or management, HealthCare.gov is replete with anti-patterns. From the oldest lessons captured in *The Mythical Man-Month* to the latest best practices in website architecture, HealthCare.gov seemingly has gone the opposite direction. Few are surprised and no one is happy — after all, it is taxpayer dollars being spent.



In sifting through the documents, reports, and testimony associated with this fiasco, I came across one item that, to me, crystallized a core problem, a problem which I believe lives in many other projects having nothing to do with the Affordable Care Act or HealthCare.gov. I was, of course, interested in learning which companies were responsible for the website. One of the early and key names to come up was CGI Federal Inc. Another was the “lead” organization: CMS, The Centers for Medicare and Medicaid Services — more about them in a minute.

CGI Federal's role is an important one, important enough that Cheryl Campbell, a Senior Vice President at CGI Federal and who leads their Health and Compliance Programs Business Unit, provided testimony to Congress on September 10th regarding their work on the Federally-Facilitated Marketplace. In her testimony she said, “At this time, CGI Federal is confident that it will deliver the functionality that CMS has directed to enable qualified individuals to begin enrolling in coverage when the initial enrollment period begins on October 1, 2013.” Given how the rollout turned out, I was particularly struck when I also found a press release on CGI's website, from one year earlier, that reads, in part, as follows: “CGI Federal... announced today that its Health and Compliance Programs Business Unit (BU) has achieved the Software Engineering Institute's (SEI) Capability Maturity Model Integration (CMMI) Development Maturity Level 5, the highest rating possible. CGI Federal is the 10th company in the United States to receive the Level 5 rating for CMMI Development, version 1.3, which notes the BU's success in establishing a culture based on continuous improvement of robust, standardized, and well-defined software development processes. Level 5 is the highest level an organization can attain in the CMMI framework.”

So, a business unit assessed at the “highest rating possible” nonetheless could not realize, a month before HealthCare.gov was supposed to be operational nationwide, that it would not, in fact, “deliver the functionality that CMS has directed.” That, to me, exposes a major problem. This CMMI level 5 organization has apparently allowed their understanding of process to triumph over an appropriate focus on product. But this is an issue that transcends healthcare and the ACA. Given how influential the process assessment ethos has pervaded other organizations across the country it calls into question how many other products either are, or will be, failures while their responsible organizations nonetheless claim process sophistication.

And CMS? Two years before the October 2013 HealthCare.gov rollout, CMS was the subject of a 193-page report produced by the National Research Council of the National Academy of Sciences. “CMS turned to the National Research Council to conduct a consensus study to strategize about how to modernize CMS's business processes, practices, and information systems effectively to meet today's and tomorrow's demands, including how to build in the flexibility to deal effectively with changing requirements.” Despite the report's numerous substantive and far-reaching recommendations for change, it appears little was acted upon. But at least the report was successfully written.

Here's hoping your organization focuses on substance, and makes process subservient to product. That's our focus.

ISR Director **Richard N. Taylor** can be reached at taylor@uci.edu.

needs to be able to quickly assimilate the constraints surrounding a software project. Moreover, an advanced programmer should also be able to quickly identify certain well-known programming styles that have been devised over the course of the past 70 years.

One very tangible goal of this project is to give advanced programming students a catalog of the various ways by which they can wrap their heads around computational tasks. The vast majority of students, including graduate students, aren't aware of this multitude of ways of thinking. Even when they are, they don't

necessarily perceive the styles as something that can have a name, be deconstructed into their essential constraints, and be catalogued and used beyond the narrow scope in which they encounter them. Lopes wants to fix that, so that they are better equipped to deal with the constraints, and resulting styles, that they

will find throughout their careers.

Another goal is to make the often forgotten points that: (1) style matters; (2) programming languages are primarily instruments for suggesting/enforcing programming styles; and (3) the enforcement done by programming languages may be strong or weak, therefore one should not rely solely on the programming language for the purposes of establishing style(s).

A third goal is to catalog the various different ways by which *entire software systems* can be engineered. What can a simple 100-line program teach us about engineering Facebook, for example? Well, a lot. Because of the fractal nature of software, the various ways by which we can decompose a small computational task into conceptual units are all similar to the various ways we can decompose large systems into components. At the core, these are the ways of sub-dividing computational problems into smaller computational problems, independent of the size of the original problem. In software engineering research, decomposition at the system level is known as “software architecture,” and the styles are known as “architectural styles.” In that light, this book is a catalog of several architectural styles. The fact that it uses a trivial computational task is, in Lopes’ view, an advantage over using production systems, from a pedagogical point of view.

Software is not just an Art; there are utility functions attached to software systems. In her book Lopes tries to stand clear of judgments of good and bad. Those judgments depend heavily on the context of each project. These exercises in programming style, similarly to Queneau’s exercises in style, are exactly that: exercises. They are the arpeggios and scales of software, they aren’t the music. A piece of real software usually employs a variety of styles for the different parts of the system. All these styles can be mixed and matched, creating hybrids that are interesting in themselves.

More information on Prof. Lopes can be found at:

<http://www.ics.uci.edu/~lopes>

<http://mondego.ics.uci.edu>

Contact Prof. Lopes at lopes@ics.uci.edu.



Director Richard Taylor and Visitor Prof. Alf Inge Wang

Visitor from Norway On a Quest to Improve Teaching through Games

ISR is hosting visiting professor **Alf Inge Wang**, from August 2013 to July 2013, on invitation by his ISR faculty host Director **Richard N. Taylor**. Wang is a professor in game technology at the Norwegian University of Science and Technology (NTNU) and is also co-founder/inventor of Kahoot! – a spin-off company that focuses on game-based learning. Software engineering, learning, and game-technology have been at the heart of Wang’s teaching and research since 2006.

This is Wang’s second stint as a visitor at ISR; he was previously a visiting researcher in 2008/2009 and enjoyed both the working environment at UC Irvine and southern California so much that he and his family came back. He chose ISR/UC Irvine because he believes “this is one of the best, if not the best, research institution in the world for software engineering and game technology, especially in the work carried out by ISR Prof. **Walt Scacchi**.”

While at ISR, Wang plans to conduct experiments related to game-based learning, participate in various seminars and workshops, and write a number of research articles based on current and previous research. One of his more specific plans is to write a book chapter in an upcoming book on software engineering and games, which Prof. Walt Scacchi is editing.

“The main focus of my research will be on how to use games in an educational con-

ISR STUDENT NEWS

Maryam Khademi (C. Lopes, advisor) presented two papers in October. The first, “An Assistive Tabletop Keyboard for Stroke Rehabilitation,” at the 2013 ACM Int’l Conf.on



Khademi

Interactive Tabletops and Surfaces (ITS ‘13) is co-authored by Hossein Mousavi, Lucy Dodakian, Steven Cramer, and her advisor **Cristina Lopes**. The second,

“Multi-Perspective Multi-Layer Interaction on Mobile Device,” at the 26th ACM Symp. on User Interface Software and Technology (UIST ‘13), is co-authored by **Mingming Fan**, Hossein Mousavi Hondori, and Cristina Lopes.

Bart Knijnenburg (A. Kobsa, advisor), presented his paper “Counteracting



Knijnenburg

the Negative Effect of Form Auto-Completion on the Privacy Calculus” at the 34th International Conference on Information Systems, held in December

in Milan, Italy. The paper is co-authored by his advisor **Alfred Kobsa** and Hongxia Jin of Samsung.

Thomas Debeauvais (C. Lopes, advisor) interned last summer at Microsoft Research in Redmond,



Debeauvais

WA where he analyzed player behavior in Forza Motorsports 4, a racing game on Xbox 360. His supervisor was Tom Zimmermann.

ISR STUDENT NEWS

Nicholas DiGiuseppe (J. Jones, advisor) spent his summer as an intern for DataONE Research Lab. Working remotely, he built an open source tool that determines the domain of relevancy of a given ontology set. He blogged about



it at <https://notebooks.dataone.org/ontology-coverage/>. He also presented his paper “Automatically Describing Software Faults” in the Doctoral Symposium of the ACM International Symposium on the Foundations of Software Engineering (FSE) held in St. Petersburg, Russia in August.

Kristin Roher (D. Richardson, advisor) presented her paper



“Sustainability Requirement Patterns” at the Third International Workshop on Requirements Patterns (RePa) held in conjunction with the 21st IEEE Requirements

Engineering Conference (RE) in July at Rio de Janeiro, Brazil.

Yubo Kou (B. Nardi, advisor) spent his summer as an intern at the



Keio-NUS CUTE Center in Singapore where he did fieldwork on professional gaming in Singapore’s Garena Stadium, an entertainment center where

local video game players gathered weekly to watch eSports.

WANT TO GET INVOLVED?

Sponsoring ISR has many benefits. It enables your company to form closer ties with our faculty and students, puts you on the fast track to our leading edge research, and gives you first crack at our experimental software tools. Choose from multiple levels of sponsorship:

| Support Level | Annual Contribution | Contribution goes to: |
|-------------------------|---------------------|-------------------------------|
| Friend | \$10,000 | ISR’s general research fund. |
| Affiliate—Collaborative | \$20,000 | Collaborative efforts. |
| Affiliate—Research | \$30,000 | Designated ISR research area. |
| Affiliate—Visiting | \$40,000 | Visitor at UCI. |
| Affiliate—Grad Student | \$60,000 | Graduate student research. |
| Partner | \$100,000 or more | Large-scale research project. |

For more information about ISR Sponsorship, please contact:

Debra A. Brodbeck
brodbeck@uci.edu
(949) 824-2260

text most efficiently to motivate, engage and teach students better,” says Wang. One of his most successful projects has been Kahoot! which began in 2007. The idea behind Kahoot! was to bring a game into the classroom where the teacher as well as all the students could play together in a ‘game show’ styled quiz game using their own devices. This is very similar to existing student-response systems out there, but the Kahoot! system has been designed from the ground up as a game. Wang’s Kahoot! team has carried out several experiments with their Kahoot! system, which have been very promising – and they are planning to do more experiments here in California and more specifically at UC Irvine.

Wang’s stay has proven very fruitful so far and has opened many new connections between NTNU and UCI.

If you are interested in trying out Wang’s game-based learning system, go to getkahoot.com where you can try out Kahoot! for free.

A video of Prof. Wang’s presentation “The Use of Game-Based Learning in the Classroom” which includes a demonstration of Kahoot! in action can be found on the ISR YouTube channel at:

<http://www.youtube.com/user/ISRUCI>

Prof. Wang can be reached at alfw@idi.ntnu.no.

Become Part of the ISR Family

Rubbing elbows with ISR faculty, staff and students gives you a valuable window into the technology landscape of the future. But a relationship with ISR can be much more: Think of us as an extension of your



company — a think tank, an R&D department, a research library, a consulting firm, a training department, and an employment agency, all rolled into one. More importantly,

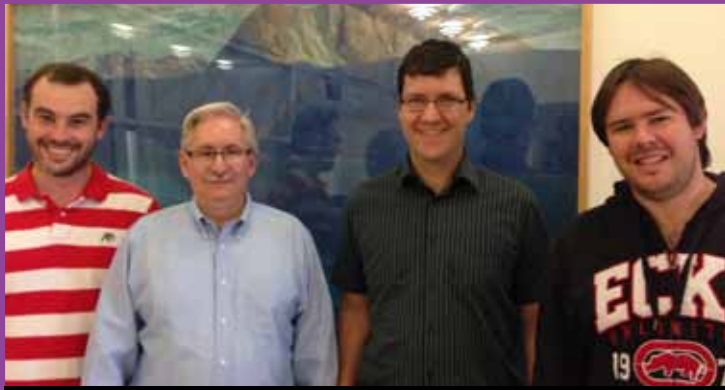
when you sponsor ISR you become part of a friendly group of folks who speak the same language and are eager to work with you to solve your current technical problems in the most cost-effective way possible.

Be part of the ISR Family – a Friend, Affiliate, or Partner.

For more information, visit:

<http://isr.uci.edu/partnerships/sponsorship/>
or contact:

Debra A. Brodbeck
Assistant Director
brodbeck@uci.edu, (949) 824-2260



Visitor Steinmacher, Prof. Redmiles, visiting Prof. Gerosa, and visitor Weiss.

Trio of Brazilian Researchers Visit Prof. Redmiles' CRADL Lab

Marco Aurelio Gerosa, an associate professor at the University of São Paulo (USP), arrived at UCI in July with two of his Ph.D. students, **Igor Steinmacher** and **Igor Wiese**, both of whom are also lecturers at the Federal University of Technology - Paraná (UTFPR), Campo Mourão. ISR Prof. **David Redmiles** is pleased to be hosting the three Brazilian researchers this year at his Collaboration Research in Action, Design, and Learning (CRADL) Laboratory. "We are already having exciting collaborations around the theme of global software engineering as a social networking enterprise," says Redmiles.

Gerosa is on sabbatical this year and opted to spend his time at UC Irvine because of its solid reputation and prominent researchers in the fields of software engineering and CSCW, and especially on the intersection of these two areas. UCI has

a long history of research bridging these two research areas.

Gerosa's research focuses on software engineering and computer supported cooperative work (CSCW), including empirical software engineering, mining software repositories, software evolution, and social

dimensions of software development. Recently he's worked on mining sociotechnical information from software repositories to improve the identification of logical dependencies between artifacts, to uncover test characteristics that indicate source code quality, and to discover gamification opportunities for virtual communities.

Igor Steinmacher's main research goal is to investigate how to facilitate the onboarding of newcomers to open source software (OSS) projects. He combines empirical methods to identify factors that hinder to newcomers. While at UCI, he is interviewing experienced open source software developers and conducting qualitative analysis on this data to identify hindering factors. The next steps of this research include (a) finding out the mechanisms – mainly awareness mechanisms – that support newcomers in overcoming the identified factors and (b) conducting a series of experiments to assess the effectiveness of these mechanisms.

Igor Wiese's current research concerns

ISR STUDENT NEWS

Hitesh Sajnani (C. Lopes, advisor) interned this summer at Microsoft Research in Redmond, WA working

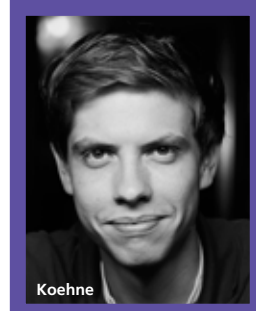


Sajnani

on a data platform tool designed for easy analysis of engineered processes. His supervisor was Dr. Rob DeLine. Sajnani has also been appointed as

the Local Arrangements Co-Chair for the International Conference on Program Comprehension (ICPC) collocated with the International Conference on Software Engineering (ICSE) to be held in Hyderabad, India in June 2014.

Benjamin Koehne (D. Redmiles, advisor) spent his summer as an intern



Koehne

at Google in New York City, NY working for the Google Drive Team. Koehne worked with designers, developers, and project managers to evaluate new versions of Google Apps

for desktop and mobile platforms at Google's own UX lab.

Xinning Gui (B. Nardi, advisor) spent her summer conducting field studies



Gui

on determining how information and communication technologies (ICTs) are used to support 'Transition Towns' transition from vulnerability to resilience in the Transition

Towns Bristol and Totnes in the UK.

RESEARCH BRIEFS

Prof. **André van der Hoek** delivered a keynote talk titled "Designers and Designs: How do we Design Software?" at CONISOFT 2013, the Mexican International Congress of Research and Innovation in Software Engineering, held in Xalapa, Veracruz, Mexico in October.

Prof. **James A. Jones**, with Stephan Diehl of the University of Trier, co-organized the New Ideas and Emerging Results (NIER) and Tool-Demo track of the first edition of the IEEE Working Conference on Software Visualization (VISSOFT) – the new union of the SOFTVIS symposium and the VISSOFT workshop. Jones also served on the Program Committee for the conference Main track. VISSOFT was held in September in Eindhoven, The Netherlands.

Prof. **Alfred Kobsa** gave talks on Personalized Privacy at three Silicon Valley companies – IBM Research Almaden, Samsung Research, and Google – in September.

ISR STUDENT NEWS

Nicholas Lopez (A. van der Hoek, advisor) interned last summer at Google, Irvine where he developed features and infrastructure for the Analytics API project. He also presented his paper “Using Topic Models to Understand Evolution of a Software Ecosystem”



at the Doctoral Symposium at the 9th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Int'l Symposium on the Foundations of Software Engineering (ESEC/FSE) held in St. Petersburg, Russia in Aug.

Francisco Servant (J. Jones, advisor) presented his paper “Supporting Bug Investigation using History Analysis” at the Doctoral



Symposium at the 28th IEEE/ACM Int'l Conference on Automated Software Engineering held in the Silicon Valley in Nov. He also presented his paper

“Chronos: Visualizing Slices of Source-Code History”, co-authored by his advisor **James A. Jones**, at the 1st IEEE Working Conf.on Software Visualization, Tool Track, in Eindhoven, Netherlands, in Sept.

Oliver Yi Wang (D. Redmiles advisor) spent his summer as an intern at IBM T.J. Watson Research Center in NY working for the Software Governance Team.



investigating to what extent social metrics can predict source code artifacts changes. He is also interested in discovering socio-technical factors that influence the results of this prediction. Preliminary results indicate that social metrics can improve the results when combined with architectural and process metrics. The goal now is to use regression models to better understand how each dimension contributes to change proneness.

All their research is connected to the NAWEB project (<http://nap.usp.br/naweb/>), an institutional multi-disciplinary research project supported by the Provost's Office for Research of the University of São Paulo. Its main goal is to develop and support collaborative environments on the web. The research team, headed by Gerosa, is interested in new international collaborations. The timing is very appropriate, since the Brazilian Government is heavily investing in attracting foreign visitors from all levels, from senior researchers to post doctorate, from short stays to long stays, with monthly subsistence payments. The program is called “Science without borders” (<http://www.cienciasemfronteiras.gov.br/web/csf-eng/>).

Prof. Gerosa is visiting UCI until July 2014; his students are staying until mid-January. They can be reached at gerosa@ime.usp.br, igorfs@utfpr.edu.br, and igor@utfpr.edu.br. Gerosa is can be found in his UCI office, 5228 DBH. For more information, visit <http://lapessc.ime.usp.br/>.



Microtasking Software Development

Crowdsourcing systems have demonstrated great success in enabling challenging tasks to be performed rapidly by massive crowds of casual workers. In 2011, players of the game Foldit were able to produce an accurate 3D model of an enzyme in just 10 days, a problem that had stumped researchers for 15 years. Over 10 million people use Duolingo to learn a language by translating small snippets of text. By aggregating these translations, Duolingo is able to rapidly produce translations of websites and other documents.

These crowdsourcing systems work by organizing work into *microtasks*. Microtasks enable work to be done rapidly by decomposing large tasks into smaller microtasks, enabling work to done in par-

ISR Technical Reports Available Online

ISR technical reports present information resulting from student and faculty research carried out under the auspices of the Institute. They showcase early results not available in print elsewhere. ISR technical reports are available in PDF on the ISR website. Recent reports include:

“RCAT: A Scalable Architecture for Massively Multiuser Online Environments”
Thomas Debeauvais, Arthur Valadares, Cristina V. Lopes
UCI-ISR-13-2, November 2013

“Motile: Reflecting an Architectural Style in a Mobile Code Language”
Michael M. Gorlick, Richard N. Taylor
UCI-ISR-13-1, June 2013

All ISR technical reports are available at:

<http://isr.uci.edu/publications/>



CrowdCode
15 lines of code
0 functions written
4 microtasks completed
George

Your score ★

10 points

Leaders ☰

| | |
|----|--------|
| 30 | Bob |
| 10 | George |

Ask the Crowd

Add a call 10 pts

The crowd found the following function for the pseudocall below:

```
/**
 * Computes the ticket amount, calculated as $10 for every mph over the speedlimit.
 *
 * @param {NUMBER} mphOver - mph over the speed limit (always > 0)
 * @return {NUMBER}
 */
function computeTicket(mphOver)
```

Can you either replace the pseudocall with a call to this function, or find a different way to do it? Feel free to update the code as necessary.

```
1 /**
2  Computes the amount of a traffic ticket for a car moving at
3  [speed] mph in a [speedLimit] zone.
4
5  @param { speed: NUMBER, speedLimit: NUMBER } input
6  @return NUMBER - ticket $ amount
7  */
8  function main(input)
9  {
10     /** check if speed and speedLimit are valid numbers
11
12     if (input.speed > input.speedLimit)
13     {
14         // compute the ticket amount
15     }
16
17     return 0;
18 }
```

Submit
Skip

Help, I don't know Javascript!

Recent Activity

👏 You earned 10 points for describing a function!

Give us feedback on CrowdCode! What do you like? What don't you like?

Send feedback

By decomposing programming tasks into microtasks, CrowdCode enables software to be created rapidly by crowds of transient workers.

allel. And by structuring work as short, self-contained tasks, many can quickly and easily contribute, increasing the potential pool of workers.

There are many situations in which there is a clear need to build software rapidly: when responding to a disaster or fixing a suddenly apparent deficiency in a key software system. A common response is *all hands on deck*, mobilizing developers across an organization or community to work until the issue is resolved. But traditional development processes are not designed for this mobilization, making it challenging to support developers in mak-

ing small contributions, identifying useful tasks, and coordinating the organized chaos of massive ad-hoc work.

To address this need, ISR postdoctoral research associate **Thomas LaToza**, together with a team of students and ISR professor **André van der Hoek**, have been designing a development process for crowd development. In crowd development, workers are presented with short, self-contained microtasks of programming work. Each microtask provides a single artifact, such as a function, and asks the worker to perform a task, such as to write pseudocode or brainstorm test cases. For

example, one worker might write a signature for function a, debug a test failure for b, and then edit the pseudocode the crowd wrote for a. To support this process, LaToza and his collaborators are designing CrowdCode, an IDE for software microtasks hosted in the cloud. CrowdCode tracks the state of each artifact, automatically generating microtasks as required. Workers login to CrowdCode, are given a microtask, and can immediately begin contributing.

Consider a simple example. Suppose the task is to calculate traffic tickets. The client first writes a user story: speeding

FALL/WINTER 2013

7



tickets should be issued in the amount of \$10 for every mph over the speed limit. To begin development, the system creates a microtask to write a main function and begin its implementation. As a worker begins this work, they might decide that the exact dollar amount of the speeding ticket should be computed in another function and write a pseudocall requesting a function to do this work. After finishing their work, they submit it to the server, which then fetches a new microtask to do next.

At this point, parallelism can begin. A microtask is created to begin enumerating test cases while, in parallel, another microtask is created to determine if a function exists with the functionality described by the pseudocall. As no such function yet exists, a microtask is spawned to write a description and signature for a new function providing the requested functionality. Next, microtasks can be spawned to implement the function, replace the pseudocall with a call to the new function, and implement each of the test cases.

Together with informatics Ph.D. student **Christian Adriano**, LaToza is currently investigating approaches for crowdsourcing debugging. One key challenge in crowdsourcing software development is to decompose tasks that traditionally require a global view of software into microtasks that are short and self-contained. Debugging tasks seem to inherently require a global view of the system, as the developer must localize the source of the bug to function(s) within the codebase. Yet, as a search process, there is also parallelism inherent in debugging: workers might investigate, in parallel, many potential locations of a bug.

As crowds of workers each change individual functions and tests in isolation, another challenge is integrating this work into a coherent whole. Working with informatics undergraduate **Lucinda Lim**, LaToza has explored an approach for support coordination across dependencies. As workers ask questions or change a function's interface, a discussion thread is created, notifying dependent artifacts and allowing a discussion to occur.

Crowdsourcing systems achieve rapid work, in part, by soliciting contributions from large crowds, which may involve workers with widely divergent skills or even malicious workers. Thus, crowd-

ISR EVENT SCHEDULE

Mark your calendars!

January 17, 2014

ISR Distinguished Speaker: Jonathan Grudin

Microsoft Research and University of Washington

"Managing Boundaries: Social Media Use by Enterprise Employees"



Terveen

April 11, 2014

ISR Distinguished Speaker: Prof. Loren Terveen

University of Minnesota

"Study, Build, Repeat: Using Online Communities as a Research Platform"

April 25, 2014

ISR Distinguished Speaker: Prof. Leon J. Osterweil

University of Massachusetts Amherst

"Reasoning about Precisely Defined Processes"



Samuelson

Friday, May 16, 2014

ISR Research Forum: Where Research Meets the Real World

Keynote: Prof. Pamela Samuelson, Univ. of California, Berkeley

We look forward to seeing you!

For more information:

<http://isr.uci.edu/isr-events/>

sourcing also requires an approach to ensure the quality of the resulting software in the face of poor quality work. Together with informatics undergraduates **Eric Chiquillo** (now at Zynga) and **Finlay Mitchell**, LaToza has investigated approaches for reviews, reputation systems, and individualized incentives. Based on this work, Chiquillo won an Informatics department award for Outstanding Contribution to Research by an Undergraduate.

While still an ongoing research project, crowd development has already begun to generate interest, leading to an active collaboration with MobileWorks, a commercial crowdsourcing platform.

More broadly, crowdsourcing is beginning to generate increasing attention within the software engineering research community.

Together with researchers at the University of Sheffield, the University of Milano, and Microsoft Research, LaToza is organizing a new workshop on Crowdsourcing in Software Engineering at the International Conference on Software Engineering (ICSE) in India in May 2014.

Contact LaToza at: tlatoya@ics.uci.edu.

ISR

To receive the *ISR Connector*, send an email request to: isr@uci.edu

ISR news, including the *ISR Connector*, is available at the ISR website: <http://isr.uci.edu/>

For more information, contact:

Debra A. Brodbeck
Assistant Director
brodbeck@uci.edu, (949) 824-2260

UNIVERSITY of
CALIFORNIA
IRVINE

SPECIAL THANKS

The UCI Institute for Software Research is generously supported by:

**Bloomberg
Northrop Grumman
The Aerospace Corporation**

