



From the Editor in Chief...

Lessons from System Development

Robert E. Filman • RIACS/NASA Ames Research Center • filman@computer.org

Welcome to volume 8 of *IEEE Internet Computing*. This year, we plan to run theme issues on business processes on the Web, internationalizing the Web, data dissemination, device nets, measuring network performance, and homeland security. We will also be featuring two ongoing “tracks”: Agents, edited by Mike Huhns, and Middleware, edited by Doug Lea and Steve Vinoski. We welcome two new columnists who will appear in alternating issues:

- Prof. Craig Thompson, Acxiom Database Chair in Engineering at the University of Arkansas and one of the Corba object architecture’s authors, premieres Architectural Perspectives in the current issue (see p. 83). The column will provide a platform for Thompson’s ruminations on the future of the Internet and pervasive technology.
- Genevieve Bell, an anthropologist at Intel working on bringing the human component into discussions about technology, will introduce Field Notes in the March/April issue. Bell’s column will focus on the intersections of cultural practices and emerging technologies.

Potential *IC* authors are reminded that we appreciate general research paper submissions as well as manuscripts tailored to the themes, tracks, Spotlight department (tutorials and surveys), and Peer to Peer (opinion) column. Please consult www.computer.org/internet/author.htm for submission guidelines.

Mars Exploration Rovers

This month (that is, January 2004), two NASA-launched robotic vehicles – the Mars Exploration Rovers (MER, <http://mars.jpl.nasa.gov/mer>) – are set to land on and start investigating Mars. I write this in December 2003, so depending on when you’re reading this column, you are likely to have a better idea of whether the rovers survived the heat-shield, parachute-, retrorocket-, and airbag-

eased landing and went off in search of the history of Martian water. (Success is far from certain. The rovers must survive a 19,000-kilometer-per-hour deceleration over a few minutes.)

The rovers are part of a very distributed computing system. Panoramic camera images radioed to scientists on Earth present possible exploration targets. Commands sent back direct the rovers to move to particular sites and use their instruments (panoramic camera, miniature thermal-emission spectrometer, Mössbauer spectrometer, alpha particle X-ray spectrometer, magnets, microscopic imager, and rock-abrasion tool) to gather data. It takes on the order of 20 minutes for a round-trip message from the rovers to Earth, so the rovers are robots. They drive up to 40 meters a day. The planned duration of each rover’s activity is three months, but the mission will continue until they both stop working. (The expected cause of failure is dust accumulation on the solar panels, depriving the rovers of the energy they need to make it through the night. And no, putting windshield wipers on the solar panels wouldn’t make things better.)

Back on earth, a small army of about 250 scientists and engineers conducts ground operations. Most of these people are at the Jet Propulsion Laboratory in Pasadena, California, but others are scattered throughout the world. Ground operations works around the clock, analyzing the collected data, determining activities for the next day, and carefully composing the command sequences to realize these goals.

The Collaborative Information Portal

I’ve had a small part in helping to develop the Mars Exploration Rover/ Collaborative Information Portal (MER/CIP) system for facilitating MER ground operations.¹ MER/CIP provides a centralized delivery platform for integrated science and engineering data, including scheduling and schedule reminders, clocks, collaboration, broadcast

announcements, and tracking data downloads from the scientific tools. (One motivation for building MER/CIP is that the solar-powered rovers require that the mission run on Mars time. A Mars day is roughly 24 hours and 39 minutes long. In a shocking oversight, all current calendar tools seem to be limited to 24 hour days.)

MER/CIP is a three-tier system. It integrates a Java-language, multiplatform graphical user interface (GUI) fat client; middleware based on Enterprise Java Beans (EJB), XML/SOAP Web services, and Java messaging (JMS); and a back end that incorporates relational databases, relational metadatabases about a file system, search mechanism for locating interesting artifacts in that file system, and LDAP directories for user authentication and privileges. As you can see, we're up-to-date and acronym-compliant. MER/CIP is more than 130 thousand lines of Java, and it took about 25 person-years to develop. Judging from the rehearsals, the system's prospective users seem

pleased with its features and performance. MER/CIP has progressed from being a mission frill to a critical tool.

Obvious Lessons

MER/CIP is a custom-developed tool for a single customer. It was developed in the face of hard deadlines: rockets left when they did because Mars is at its closest in almost 60,000 years. The rovers will land in January, and the ground systems have to be ready. Postponing the software release to fix bugs is not an option. (Fixing bugs introduces bugs. A couple of months before landing, the operational attitude shifts from "fixing bugs" to "learning to avoid problems.")

This system's development has viscerally emphasized for me several things that are, of course, obvious.

- It might be clear what a software system has to do, but it can still take a lot of energy to make it happen. MER/CIP required only a little in the way of novel algorithm

development. It did not demand solving any unsolved computer science problems (that is, "here's where the artificial intelligence goes"). There weren't really any major surprises in system creation. Nevertheless, it took about a dozen programmers about two years to build the system.

- System development is less and less about coding than about using things and gluing them together. Knuth has observed that literate programming today is literally thumbing through 10 manuals as you code. This is a stark contrast to my Knuthian education of 30 years ago, where the heroic programmer wrung every efficiency into the smallest data structures and tightest loops. Correspondingly, the headaches in system development have moved from finding your own bugs to discovering the actual behavior and limitations of other people's products. (These days, it might be more valuable economically to be knowl-

IEEE INTERNET COMPUTING

IEEE Computer Society Publications Office
10662 Los Vaqueros Circle
Los Alamitos, CA 90720

EDITOR IN CHIEF

Robert E. Filman • filman@computer.org

ASSOCIATE EDITOR IN CHIEF

Li Gong • li.gong@sun.com

EDITORIAL BOARD

Jean Bacon • jean.bacon@cl.cam.ac.uk
Miroslav Benda • miro@amazon.com
Elisa Bertino • bertino@dsi.unimi.it
Scott Bradner • sob@harvard.edu
Siobhán Clarke • siobhan.clarke@cs.tcd.ie
Fred Douglass • f.douglass@computer.org
Stuart I. Feldman • sif@us.ibm.com
Ian Foster • foster@cs.uchicago.edu
Michael N. Huhns • huhns@sc.edu
Leonard Kleinrock • lk@cs.ucla.edu
Doug Lea • dl@altair.cs.oswego.edu
Frank Maurer • maurer@cpsc.ualgary.ca
Daniel A. Menascé • menasce@cs.gmu.edu
Chris Metz • chmetz@cisisco.com
John Mylopoulos • jm@cs.toronto.edu

Peter Norvig • peter@norvig.com

Charles J. Petrie • petrie@nrc.stanford.edu
(EIC emeritus)

Krithi Ramamritham • krithi@cse.iitb.ac.in

Ravi Sandhu • sandhu@gmu.edu

Munindar P. Singh • singh@ncsu.edu

(EIC emeritus)

Craig Thompson • cwt@uark.edu

Steve Vinoski • vinoski@ieee.org

Jim Whitehead • ejw@soe.uscs.edu

IEEE Communications Society Liaison

G.S. Kuo • gskuo@ieee.nccu.edu.tw

STAFF

Lead Editor: Steve Woods
swoods@computer.org

Group Managing Editor: Gene Smarte

Staff Editors: Scott L. Andresen,
Kathy Clark-Fisher, and Jenny Ferrero

Production Editor: Monette Velasco

Magazine Assistant: Hazel Kosky
internet@computer.org

Graphic Artists: Carmen Flores-Garvey,
Alex Torres

Contributing Editors: David Clark,
Greg Goth, Keri Schreiner, Joan Taylor

Publisher: Angela Burgess

Assistant Publisher: Dick Price

Membership/Circulation Marketing

Manager: Georgann Carter

Business Development Manager:

Sandy Brown

Advertising Supervisor: Marian Anderson

CS Magazine Operations Committee

Bill Schilit (chair), Jean Bacon, Pradip Bose,
Doris L. Carver, George Cybenko, John C. Dill,
Frank E. Ferrante, Robert E. Filman,
Forouzan Golshani, David Alan Grier,
Rajesh Gupta, Warren Harrison,
Mahadev Satyanarayanan, Nigel Shadbolt,
Francis Sullivan

CS Publications Board

Michael R. Williams (chair), Jean Bacon,
Laxmi Bhuyan, Mark Christensen,
Thomas Keefe, Deependra Moitra,
Steven L. Tanimoto, Anand Tripathi



edgeable about a product like Oracle or WebLogic than to have the skill to build a relational database or application server.)

- Building a novel system means that what you want to build changes. Create a flexible enough architecture to allow for this. In building individual components, look for ways to make them data-defined rather than code-defined. For example, the MER/CIP code that searches the file system structure for interesting new files profited by making richer descriptions of “interesting” in the configuration data. (In general, move as much as you can out of code and into configuration.)
- Building a novel system means that customers will not be able to elucidate at the start what they want; only by using the system will they be able to tell you what you should have done.
- No matter how many design meetings you had with your future end users, they will use your product differently than you had anticipated, and they will make assumptions you never expected – or even that you explicitly denied.
- System development would be easy were it not for optimization. Much evil and many faults arise not from trying to perform the desired task, but from trying to perform it more efficiently.
- System development would be easy were it not for dealing with failure. This is particularly true of distributed systems where, as Leslie Lamport has observed, the failure of a component that you didn’t know existed can cause your system to fail. In real systems, things don’t work as planned or promised. You must be prepared for that possibility.

There were also several non-obvious lessons from the development experience, but for those, you’ll have to wait for the publication of the research papers.

True Lies

Somewhat counterintuitively, some obvious things have not worked out to be true in practice.

- You don’t need extended evolution to create a usable user interface. Perhaps in credit to the user-interface developers, the overall structure of the MER/CIP interface and its implementation details converged fairly quickly to a workable (though not perfect) organization. (Of course, it might help that the users are rocket scientists).
- It is possible to create systems using a far more bottom-up process than perhaps expected. The use of protocols and interfaces allows more independent component development than classical software-engineering theory would recommend. The software development proceeded smoothly – even through changes such as going from browser-based applets to fat clients (applets take too long to load and don’t give enough control over the interface) and from Java RMI to Web services (system administration hav-

ing decided that open sessions are insecure).

- You don’t need an elaborate, formal modeling activity before coding. (I’m not sure this one quite fits into “obvious but not true,” since I never believe it anyway.) Early modeling would have turned into archived lies.

So not everything they teach you in school – even with modern software-engineering training – turns out to be the case.

Landing

MER-A (Spirit) lands in the Gusev Crater on 3 January 2004 at about 12:35 GMT. MER-B (Opportunity) lands in the Meridiani Planum on 24 January at about 13:05 GMT. In a future column, I’ll report on how the mission went, and how the portal software held up in real use. □

Reference

1. J. Walton, R. Filman, and J. Schreiner, “The Mars Exploration Rover / Collaborative Information Portal,” *Proc. 10th Int’l Conf. Human-Computer Interaction*, Lawrence Erlbaum, vol. 3, 2003, pp. 1363–1367.

How to Reach IC

Articles

We welcome submissions about Internet application technologies. For detailed instructions and information on peer review, see *IEEE Internet Computing’s* author guidelines at www.computer.org/internet/author.htm, or log on to *IC’s* author center at Manuscript Central (www.computer.org/mc/internet/author.htm).

Letters to the Editor

Read something in *IC* that you want to respond to? Please email letters, including reference to the article in question and when it appeared, to internet@computer.org.

Reuse Permission

For permission to reprint an article published in *IC*, contact William J. Hagen, IEEE Copyrights and Trademarks Manager, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08855-1331; copyright@ieee.org. Complete information is available at www.computer.org/permission.htm. To purchase reprints, see www.computer.org/author/reprint.htm.