conference reports

COOTS 6

by Peter H. Salus
Our peripatetic historian
<peter@matrix.net>

How many conferences/symposia/workshops of type X are enough conferences/workshops of type X?

I got the feeling at COOTS 6 (6th USENIX Conference on Object-Oriented Technologies and Systems) in San Antonio in January that nearly 100 of us loyalists were determined to answer that riddle. Is that a conference or a workshop?

There were five USENIX Graphics Workshops, but there hasn't been one in over a decade. Tcl/Tk seems to have run its course. Etc.

Most of us use objects. Object-oriented technology is foreign to no one in the community. But how many workshops does one need? I don't really know. I enjoyed this one a lot; it was small enough that I was able to actually say hello to nearly everyone I wanted to. For the first time since Toronto, I didn't ask to be invited to the Advanced Topics Workshop.

I got a chance to talk to Ken Arnold and Deborah Zukowski, to Doug Lea and Rajendra Raj. And, once more, to hear the invariably entertaining Stu Feldman.

Stu gave the best keynote (in my opinion) at any USENIX conference – in Salt Lake City in the distant past (1984). He was the keynote at another USENIX in 1992 in San Antonio, too.

In SLC his talk was disrupted when the hi-tech slide projector failed; in San Antonio, there was a fire alarm. Stu survived.

Feldman's thrust was on two things: speed of change ("every six months a new gadget") and ubiquity. The tighter the software community, the worse the interface constraints. Do we want to be constrained to the brief messages of Japanese schoolgirls? He pointed out that there was a $300 billion investment in mobile electronic business.

In an historic aside, Stu pointed out that we had moved from the mammoth centralized server to the client server, to the Internet, and that we were becoming Web/Network-centric. This means that personalization, notification, efficiency of information, and location sensitivity are coming. He pointed out that Dick Tracy's wristwatch was already in experimental form and might well be available soon. (But, he admitted, we'd have to wear a battery pack.)

We will be operating with "layers of service" and "nets of information." There is thus no simple device model. We've been living with the PC model, but we're moving further into diversity. This will be dreadfully exciting; we'll deal with peta-everything. We need high connectivity, device capability, and integrated services, says Feldman. The problems will lie in the areas of authentication, authorization, and transfer.

"Devices are objects. User models are objects. Business models are objects. Business processes are affected by the changing state of such objects."

"Applications are the real driver," Feldman concluded.

It was a really fine talk; lots to think about. Thanks, Stu.

That afternoon, there was a guest lecture by Robert Martin of Extreme Programming. If you know nothing about XP, I suggest one of the Addison-Wesley books. Martin defined XP as "a set of simple rules from which complex behaviors arise." His form of this might be summarized as:

Due Date is #1
Due Date is frozen
The spec is never frozen (the specs change all the time)
Ad hoc management
Waterfall model

Analysis/Design/Implementation go ahead in parallel (with feedback), not sequentially.

Martin made a number of really good points:

- If you reduce quality, you slow down
- Dates can't be changed
- Adding staff makes it later (Brooks)
- You can change scope by deleting features

He also offered some "rules":

- Write tests before code
- Program in pairs
- Integrate frequently
- Rest
- Communicate with customers daily
- Follow customers' priorities
- Leave software clean/simple at the end of the day
- Adjust processes and practices to your environment

Martin was entertaining and provocative.

The next morning, Bjorn Freeman-Benson spoke about software management perspectives. Using the analogy of orienteering, he spoke about several large software projects he had worked on and their "problems." OTI, Rational, Amazon.com, and QuickSilver Tech all had different problems.

OTI built a "repository-based solution, not a file-based solution." VisualAge was what "we wanted to build, not what the customers wanted." Rational wanted to create a totally new system, not a build on top of other stuff – not really a "legacy system."

Amazon built many little programs in C, rather than objects. This made training and refactoring expensive, though maintenance costs were low. Amazon ended up with three distinct layers: db (Oracle), templates (internal scripting language, obidos, and Perl), and software (plugins to Apache, in C/C++).

Quicksilver has concentrated on using lightweight processes to adapt to business needs.

Freeman-Benson's point was: "Engineering is about the big picture *and* about the details": set a goal, plan, execute, be flexible.

Once upon a time, a decade ago and more, Smalltalk, C++, Eiffel, Java were new or nascent. COOTS is now more about the big problems than the little ones. The Stroustrup-Cargill discussions about inheritance are of little import. Ken Arnold's BoF on Jini was a big deal.

With under 100 attendees, maybe COOTS should now be shelved; something else might sprout up.

I'm pleased it's not my decision.