Conference overview:

Smalltalk Solutions '96: Progress and new challenges

David Carr

THE SMALLTALK FAITHFUL reconvened in New York this March after a year of dramatic change. Since last year's Smalltalk Solutions, ParcPlace and Digitalk merged, Easel became part of VMark, and OTI became a wholly-owned subsidiary of IBM. Meanwhile, the Web emerged as a potentially significant application development tool, propelling Java into position to challenge Smalltalk. These changes and challenges figured in the keynote talks of IBM's VisualAge Marketing Manager Skip McGaughey and STIC's Executive Director Reed Phillips, while ParcPlace founder Adele Goldberg discussed Smalltalk as a teaching tool for college students and computer professionals.

The study STIC released last year found that companies adopting Smalltalk were more likely to have followed a formal process in choosing a programming language.

Smalltalk is gaining worldwide market acceptance as a solution for serious business problems, according to McGaughey. No other technology is as scalable and robust, "But we've got to translate this rich technology into business value."

Global economic commerce is accelerating the pace of change enormously. The trend is most obvious on the World Wide Web, where fortunes are being made overnight. McGaughey feels the value of that opportunity is "almost unimaginable. It's going to radically alter our management, and how we live, how we work, and how we play. Look what happened to Netscape." The explosion of interest in the Web, in turn, has transformed Java into a serious contender. "Java is going after not just market; it's going after mindshare," McGaughey said; Smalltalk needs to keep moving to stay competitive.

There are great opportunities overseas, McGaughey said, citing projects in Turkey, Brazil, China, and the former Soviet Union. Another huge opportunity is a vast number of legacy systems. In one bank alone, IBM found 54,000 date routines that need to be changed. "We estimate there is $400 billion in code that needs to be rewritten between now and the year 2,000. Here is the challenge: why not rewrite them in Smalltalk?"

"IBM is absolutely, fundamentally, wholeheartedly committed to Smalltalk,"McGaughey asserted. About OTI's acquisition: "We've set it up as a wholly-owned subsidiary, but all the Smalltalk people in IBM now report to (OTI President) Dave Thomas. So maybe it's the other way around —that OTI acquired IBM."

IBM recognizes that it needs more successful partnerships like the one with OTI, which produced IBM Smalltalk. "We want to have 1,000 vendors out there building parts and creating components, and they have to know where we're going. For instance, Object Share is taking out our visual programming environment and putting in their own on top of IBM Smalltalk.We have to enable our competitors because they provide value."

Smalltalk is going through a period of transition, enjoying unprecedented success at the same time that it is "being attacked from the bottom by Visual Basic, Delphi, and Java," he said. Different strategies are required now that Smalltalk is coming into the mainstream. "We need to send different kinds of messages to make different kinds of sales. Instead of selling competitive advantage, we need to sell, 'This is safe, this will scale.'"

"We're beginning to sell to a different kind of clientele," Reed Phillips agreed. "They don't really like technology, and they don't like taking risks." Those promoting Smalltalk for corporate systems now point to prominent success stories and provide more references, he said.

Still, misperceptions persist. Outsiders see Smalltalk's performance as a major weakness, while only 2% of Smalltalkers agree, he said. And many corporate programmers know so little about the language that it seems too exotic to be a practical choice, "So a lot of people decide, 'Why don't we use something less complicated, like C++.'"

The study STIC released last year found that companies adopting Smalltalk were more likely to have followed a formal process in choosing a programming language. "If we can get people to do real comparisons, then Smalltalk has a significant advantage," Phillips concluded. "Smalltalk seems to have to fight its way into an organization, but once it's there, it does pretty well." Smalltalk projects also were twice as likely to achieve their expected goals. "The Smalltalk industry has the opportunity to grow and prosper be cause of the successes that are there. It's a matter of getting the word out," Phillips said.

To Adele Goldberg, the issue is not just teaching Smalltalk, but teaching systems building as opposed to programming. "Too many university computer science curriculums stop at teaching data structures and algorithms," she said. It's not surprising it so hard to recruit people capable of building extensible, adaptable systems. "The most significant part about a system is that once we start it up, there's a maintenance issue. You want it to run indefinitely." And while people can learn the syntax for programming in Smalltalk in an afternoon, "they don't get the systems building part," Goldberg said.

Her solution is LearningWorks, a modified version of the Smalltalk implementations she used to teach programming to 12-year-olds. Its interface is organized into a neat binder of several "books" used for system planning, experimentation, and development, and it feeds students the modern Smalltalk class library a little at a time. Using the internet as a medium for distributing this free tool, she plans to have Open University students collaborate on building LearningWorks systems as class projects.

Students can start by experimenting with rehearsal worlds that illustrate key concepts and provide a context for exercises in organizing behaviors and allocating responsibilities, Goldberg said. Businesses could train their employees by having them create LearningWorks books that represent the essence of the company's framework.

Reg Krock of the Ontario manufacturing firm Maksteel was one of the people who approached Goldberg after her talk to express interest in obtaining a copy of LearningWorks. "One reason is that we have a 67-year-old president of our company. I could give that to him, and he would actually play with it."

Computer systems are the only part of the business that Maksteel's president doesn't fully understand, which makes it harder for him to manage, Krock said. "There's always been a language gap between the CEO and the CIO. What I'd like to do is take some of the mystique out of it."

David Carr is a freelance writer specializing in the object-oriented programming industry. He can be reached at davecarr@pcnet.com.

GETTING REAL

continued from page 22

virtual machine. Each session process has two caches in which to access objects, in addition to the shared page cache. One cache, called the temporary object cache, is where new objects are created. As the execution of server Smalltalk code causes new objects to be created, they are created in a section of memory carved out just for that purpose. This area of memory is garbage collected by generational scavenging techniques, since many newly created objects die early and can be garbage collected soon after their creation. If this cache should become full, then some objects from it must be written to disk, where garbage collection is more expensive. To determine the appropriate size for the temporary object cache, a system designer must consider the total size of all new objects created during a single transaction.

The other cache utilized by the session is the private page cache. This cache is a private area in which to read and write pages of objects. This cache is usually small, since the session primarily uses the shared page cache to read and write objects. If the system is configured not to allocate a shared page cache on the machine where a particular session is executing, then its private page cache size should be increased accordingly.

A session's process can get a variety of information about itself. To monitor garbage collection activity in the temporary object cache, a session can get the #TimeInScavenges statistic to find out the CPU time spent performing in-memory garbage collection, #NumberOfScavenges to find out the number of times the in-memory garbage collector has been executed, or #NumberOfMakeRoomInOldSpace to find out the number of times the oldest generational space filled up (a large number may indicate that the session's temporary object cache size is too small). A session can also find out how well it is using the shared page cache. It can get the #NumberAttached statistic to find out the number of pages that the process is currently using in the shared page cache, and #LocalPageCacheHits and #LocalPageCacheMisses to find out how many times a page was found or not in either the shared page cache or the private page cache. A session can measure its transaction activity by looking at the #NewObjsCommitted statistic to find out the number of newly created objects committed by the most recent transaction, and #NumberOfCommits and #NumberOfFailedCommits to get a cumulative number of successful or failed transactions since the session began.

The statistics described above are but a sampling of the kinds of information to look at when configuring a multiuser Smalltalk system. The key to successfully configuring and tuning such systems is understanding the multiprocess nature of clients and servers, and how different memory spaces and caches are used. Fortunately, tools are available to gather these statistics over long periods of time and then graph the results to analyze overall system performance. Without the ability to gather statistics about each process in system, tuning is a shot in the dark.