

## Editors' Corner



John Pugh



Paul White

**W**E WRITE THIS as we watch the Olympic Games draw to a close. It is absolutely amazing to witness the accomplishments of the athletes in attendance, and to try to comprehend the dedication they each have to their sport. (We're also still reveling over the Canadian gold medal in the 4x100 relay!) The other amazing feat, from our perspective, is the advances in technology that were on display during the Games. Of course, the problems experienced were well documented and many criticized the technology as being a failure (sound familiar to conversations with your users?), but in many ways the criticisms are unfair, given what has been attempted. It's another case of unrealistic expectations coupled with an unwillingness and/or lack of opportunity to manage those expectations.

We have been told that the software applications built by IBM using VisualAge fared fairly well at the Games, but we have no confirmation of this fact. In any event, I think if the public stopped to realize just how far engineering has advanced in such a short time, they would begin to see that things are heading in the right direction. One might even draw the conclusion that we must be getting closer to building useful technology, considering people have much stronger opinions as to what is being built. In any event, hats off to everyone involved!

Last month we touched on the problems faced by many of us in managing the persistence aspect of our applications. While for most, persistence is undoubtedly the number one enemy in today's applications, there are certainly other common problems facing a large majority of us. One of the items heading this list is report generation.

Report generation has always been, and will no doubt remain, an extremely difficult problem. Deciding how to best utilize an 8-1/2 x 11 sheet of paper is an almost impossible task. Its limitations are obvious—there is a fixed-size region in which to print, and the items we wish to print do not have such limits. Features such as proportional fonts and text formats cause even more grief than what was faced by those true warriors of report generation over the past decades—the RPG programmers.

There are two aspects to the problem. The first is the technology for laying out reports, specifying the

desired behavior and the final 'look-and-feel' of the ultimate report. On this front, the news continues to get better. Both ParcPlace-Digitalk and IBM provide some sort of facilities for specifying reports, as do a handful of third-party vendors. Each of these has both strong and weak points, but the features being introduced are allowing report designers to work at a better level of abstraction for describing reports.

The second aspect of the problem comes from the application itself. Actually attempting to describe what is the desired behavior, from an application point of view, is a monumental task. Each report, of course, needs a header and a body (and perhaps a footer), but the amount of space required may vary,

depending upon the type of report and the contents contained within it. Deciding how to split a report that must run multiple pages is a real challenge. What's more, if the report contains cross references to other items within the report (as is

commonly found in insurance claims forms, for example), an automated process for describing this is extremely elusive.

In the end, one must remember that the problem being addressed is in many ways not solvable, at least not easily. The proof of this is simple: Just consider describing how to draft a report, describing how to handle every single possible configuration for the next five years, guaranteeing you haven't missed any case. If you can't solve it in English, you can't solve it using any modern computer language either. Having said this, what we need to do is provide much better tools to allow users to describe their solutions using abstractions, which make sense in terms of a report layout. We believe these tools will continue to improve.

Finally, we should note this issue represents an accomplishment we never envisioned when we began. When we started *The Smalltalk Report* in September 1991, we did it with the hope that it would help people like you to utilize Smalltalk more effectively, to serve as a place to share ideas, and hopefully, in some small way, help the Smalltalk industry grow and mature. After a full five years of it, we think we can stand up and say we've achieved this goal. And we certainly look forward to writing an editorial after our 10th year of the *Report!*

Enjoy the issue.

*Deciding how to split  
a report that must run multiple  
pages is a real challenge.*