The Man Behind REXX

z/Journal Visits IBM Fellow Mike Cowlishaw

By Gabe Goldberg

In today's team-oriented world, it's the rare project that runs on one person's inspiration and perspiration. Equally uncommon are development results that shape technology, affecting how companies, developers, application users and consumers work.

The REXX programming language was such a project, conceived and implemented by Mike Cowlishaw, now an IBM Fellow. This title is an executive-level scientist/engineer/programmer position, the technical-side pinnacle of IBM's dual-career ladder. The few dozen CEO-appointed Fellows pursue self-directed projects for renewable five-year terms; they serve as corporate consultants and mentors, and report yearly on activities.

Mike joined IBM in 1974, working on hardware and software products and interests such as the human-machine interface, programming languages and editors. Around 1980, he designed and developed REXX, "driven by the desire to make programming easy." Documented before it was developed and shaped by extensive internal user feedback, it quickly grew in popularity within IBM. Customers learning of it clamored for its release; it shipped in 1983's VM/System Product Release 3, and then was ported to MVS and VSE. Now a key resource on all IBM computing platforms, REXX is also available on most other computing platforms. Senior consultant/programmer/trainer, Chip Davis, remarks, "As soon as I installed REXX on our VM system, all [languages] but Assembler became superfluous. They have been the only two guns in my holster for the last 20 years."

Returning to his interest in editors, Mike developed Lexx, a specialized (syntax-directed color-coding) editor for Standard Generalized Markup Language (SGML) text of the *Oxford English Dictionary* (*OED*), second edition. Lexx and its derivatives are part of IBM's VisualAge products, and Lexx itself recently became available for free download. The *OED* acknowledges Mike twice—for Lexx and editorial contributions; he still consults for the *OED*. Mike led IBM's technical assessment of Java and created NetRexx, a Java platform REXX version.

Mike's books, *The REXX Language* and *The NetRexx Language*, and www2.hursley.ibm.com/rexx/ provide extensive REXX information. In this interview, Gabe Goldberg, co-editor of McGraw-Hill's *The Rexx Handbook*, talks with Mike about his eclectic and evolving interests and perspectives on technology and the industry.

***z/Journal*:** When did you realize the scale on which REXX would succeed?

Mike Cowlishaw: I suppose it was when the VM development group decided to add REXX to CMS before they had even shipped EXEC 2.

***z/Journal*:** What would you do differently if you could do REXX and NetRexx again?

MC: Well, NetRexx really gave me the chance to correct the things in REXX that I wanted to correct, such as case-independent comparison and the keyword LOOP for do-loops.

***z/Journal*:** What's the most surprising platform to which REXX has been ported?

MC: Palm/OS is perhaps the smallest platform that REXX runs on, but really none of the ports is surprising.

***z/Journal*:** REXX has been adopted throughout IBM, ported to numerous platforms, the subject of dozens of books and an annual symposium. It's made several programmer generations' work easier, more productive and enjoyable. Yet it's not as widely used as some trendy, sometimes ephemeral "languages de jour." Why hasn't the industry adopted REXX more widely?

MC: That's a difficult question to answer, as any answer must be speculation. But I wonder if REXX is perhaps almost too easy to use; not enough of a challenge! I've described it in the past as "BASIC done with hindsight" and indeed, some later BASIC variants are very REXX-like or NetRexx-like. Yet, BASIC still has a bad name in some circles. I'm confident that many of the best features of REXX (simple syntax, few notations, strong string processing, decimal arithmetic, etc.) will continue to show up in other languages.

***z/Journal*:** You've indicated that NetRexx is "designed for people, not computers." What other products—computer-related or not—do you consider to have been designed that way?

MC: Generally, it's the products that follow an existing, well-known user model. One that springs to mind is Lotus Organizer; this closely follows the "paper-based" organizer model, in both appearance and operation, and so is easy to use. A more modern example is perhaps Google.

***z/Journal*:** You've articulated the "minimal astonishment factor" as a design goal. What does this mean and where have you seen it most violated?

MC: I've described this in the past as follows: If a feature, accidentally misused, gives apparently unpredictable results, then it has a high astonishment factor and is therefore undesirable. This is another way of saying that if the program doesn't match the user's model of what's going on, then it has a design error.

***z/Journal*:** You've been quoted as hoping that human programmers won't be needed in 20 years. What might lead to that change?

MC: I was perhaps a little optimistic about the 20 years, but I still believe a computational approach derived from spiking neuron models will lead to truly intelligent systems. Those should be able to handle the tedious side of programming, allowing us to spend more time on the creative aspects.

***z/Journal*:** Spiking neuron models?

MC: Yes, that is a model of a neuron where fast learning is seen at the synapses; positive if an input spike just precedes an output spike, negative if it just follows (that is, the connection is strengthened if the input is a predictor of the output, and vice versa). This behavior has long been suspected, but it has been observed only in real neurons in the last 10 years. It's interesting because it explains a number of experimental observations, and, I think, will explain how the brain extracts patterns over time (e.g., recognizing words in a stream of speech).

***z/Journal*:** How have IBM and the computing industry changed in your nearly 30 years at IBM?

MC: The biggest change is that most of the easy things—the projects that can be done by just one person—have been done. There are plenty of "big challenges" left, though (e.g., modeling the brain).

***z/Journal*:** Could you develop REXX and have it blossom in today's IBM?

MC: I think so, but probably it wouldn't be too successful. At the time, it was a huge advance over the scripting and macro languages of the day. Nowadays, there are plenty of other languages around that can do the job reasonably well, even if they are not as easy to learn.

***z/Journal*:** Guide us through the technical interests (decimal arithmetic, REXX, electronic publishing, vision/color perception, etc.) listed on your Web page. Is there a common thread among them? One connection is clear: accelerating REXX decimal arithmetic with underlying hardware.

MC: The common thread is that they are all at the "interface" between people and computers, so there is a strong human element. Decimal arithmetic uses the base that people use; REXX makes programming fit the user's model of the world, not the computer's; vision and color perception directly affect the way we present information on displays.

In all these fields, the biological aspects mean that the answers are not necessarily obvious. For example, from a computing point of view, binary floating-point numbers have the edge over decimal ones—yet for people, the base just has to be 10. (0.1 is an infinitely recurring binary fraction, but is trivial in decimal.)

***z/Journal*:** What's the status of decimal arithmetic? How do you see it developing?

MC: It's come a long way; the IEEE 754 standard is being revised to include decimal floating-point, and my design is now fully or partially included in many different languages. And we (IBM) are putting it into hardware, too.

***z/Journal*:** How and when will native-hardware decimal arithmetic be exploited? What effects will it have on programmers and computing?

MC: IBM will be putting it into processors quite soon. This (and all the related standards) will have a similar effect on decimal arithmetic as the IEEE 754 standard had on binary floating-point: it becomes a no-brainer to use the standard formats and arithmetic in languages, rather than inventing new ones. This means we'll see native decimal formats in most languages, in time, which will make programming much more pleasant and quite a bit easier for many applications. And they'll run faster, too.

***z/Journal*:** What's planned for future exploration?

MC: The spiking neural models are a background interest at the moment, as the decimal standards and implementations are keeping me busy. But I'd like to spend more time on that.

***z/Journal*:** How do IBM Fellows influence IBM research, technology, and products? Are Fellows encouraged to focus on probable commercial research areas or can you explore where your curiosity leads you? What is work like for an IBM Fellow?

MC: There are only 53 IBM Fellows out of the 178,000 technical staff in IBM, so we're spread rather thinly and mostly work in very different fields. Those outside research (like me) tend to work closer to product development than the others, but it varies.

***z/Journal*:** Do you have a staff or do you work independently?

MC: Independently, though last year I had a part-time assistant who helped with verifying the specification for the IEEE 754 standard.

***z/Journal*:** During your career, have you had mentors or role models? Do you and other Fellows mentor IBMers, help orient them to successful technical careers?

MC: I don't think I had a specific mentor, though I am sure some senior people were helping me along. Nowadays, there's a strong focus on mentoring in IBM; I mentor five people formally and about the same number informally.

***z/Journal*:** What changes to IBM and the industry do you expect in the next 30 years?

MC: I would expect the variety of technical jobs to shrink somewhat; there's no great advantage in having 10 different machine architectures that are only a little different, for example. On the other hand, the neural network research could well lead to a whole new way of doing things. Or there could be some other breakthrough to keep life interesting.

***z/Journal*:** What would you like to accomplish technically and professionally?

MC: I'd like to be involved in that breakthrough, when it happens.

***z/Journal*:** How do you spend your non-work time?

MC: Family holidays are always a good break, and I also go to Spain for 10 days every summer (without a laptop), and do nothing but hike, explore wild caves, and generally relax. During the rest of the year, it's mostly work, though I still enjoy "recreational programming"(such as PMGlobe, a programmable world globe, www.cary.demon.co.uk/pmglobe/) and electronics (for example, my tiny caving lamp).

About the Author

Gabe Goldberg is a technology journalist and enterprise consultant, focusing on Internet and enterprise computing. A frequent conference speaker/organizer, he also volunteers with DC-area groups, helping professionals and consumers apply technology to their daily lives.
e-Mail: gabe@gabegold.com