REXX 1995 – The Growth of a Language

M. F. Cowlishaw
IBM Fellow

Pages 8-32

Rexx 1995

The Growth of a Language

Mike Cowlishaw

IBM UK Laboratories
Hursley, England

Outline

- The first year
  - Background and context
  - Initial specification, refinement, and evolution
  - Retrospective
- 1980-1995

Reference:

The Early History of Rexx, Mike Cowlishaw
IEEE Annals of the History of Computing,
Vol 16, No. 4, 1994

Whence Rexx?

Rexx grew from two concepts:

1. A *single* macro language for *many* applications (first expounded by Stephenson in 1973)
2. A language designed for the benefit of the *user* (programmer), not the *language implementer*

Traditional macro languages

Macro languages assumed that most of the content of a program would be literal data:

```
&IF &NODE&J ¬= &LOCAL &USER = &STRING OF
&USER&J AT &NODE&J
```

By 1979, programs existed where more than 50% of the tokens began with "&".

The solution:

```
if node.j¬=local then user=user.j 'AT'
node.j
```

March 20-29, 1979

Discussion with EXEC 2 people [March 22]

"... I'm thinking of implementing an experimental EXEC processor to handle a more ... PL/I-like language. ... This is of course the *dual* of the EXEC/EXEC 2 languages, in that literals are identified, rather than variables/control words, but ... EXECs nowadays often seem as complex as programs ... and that therefore literals are often a very small percent of the tokens in an EXEC".

→ first specification for **REX** [March 29]

First specification (1)

- 5 pages of introduction and rationale
- 10-page language description
- 4 pages of examples
- Eleven instructions (IF, DO WHILE/UNTIL, SELECT, QUEUE, PUSH, PULL, SAY, EXIT, RETURN, TRACE ON/ERROR, ERROR)—plus a proposal for REX (INTERPRET)
- 8 special variables (BLANKS, DATE, N, NL, Q, RC, RETCODE, TIME); DATE, Queued, and TIME became functions.

First specification (2)

- There were three example programs (including bugs). For example:

```
/* Send file to a local user */
Pull name fn ft fm;
CP SPOOL PUN name CLASS A;
if rc¬=0 then do; /* check it worked */
  say name is not a valid userid;
  exit 102; end;
PUNCH Fn Ft Fm;
CP SPOOL PUN * CLASS A;
```

etc.

Refinement

- Hundreds of pieces of mail refined the initial specification
- Arguments such as DO...END versus IF...ENDIF
- Version 0.01 to Les Koehler and Ray Mansell [May 21]
- Initial specification had evolved to 30-page reference manual [by June]
- Rapid growth of features, following suggestions (better tracing, hex strings, nested comments, *etc.*)

Key features

- ✦ Control structures
- ✦ Parsing—PULL and decompose into words
- ✦ Fluidity of symbols (multiple uses)
- ✦ Concatenation with blank
- ✦ Alternative quotes for literals
- ✦ Lack of “boilerplate”
- ✦ Case-insensitive comparisons (later removed)
- ✦ Case-preservation for literals (later removed)
- ✦ Tracing

Performance

- ✦ Comparisons with EXEC, EXEC 2, and PL/I
- ✦ Test loop: 3.31 seconds (on S/370 model 155):

```
i=0
do 2000
   i=i+2
   end
```

1995:

- ✦ 0.19s on a 486/33MHz PC

A typical week— the first of 1980

- Requests for a more PL/I-like DO instruction, with the ability to step a control variable
- Requests for subscripts (rejected because, among other things, "... the obvious syntax, using square brackets, is not practical because so few people have brackets on their keyboards")
- A user contributed a draft quick-reference card
- Positive feedback:

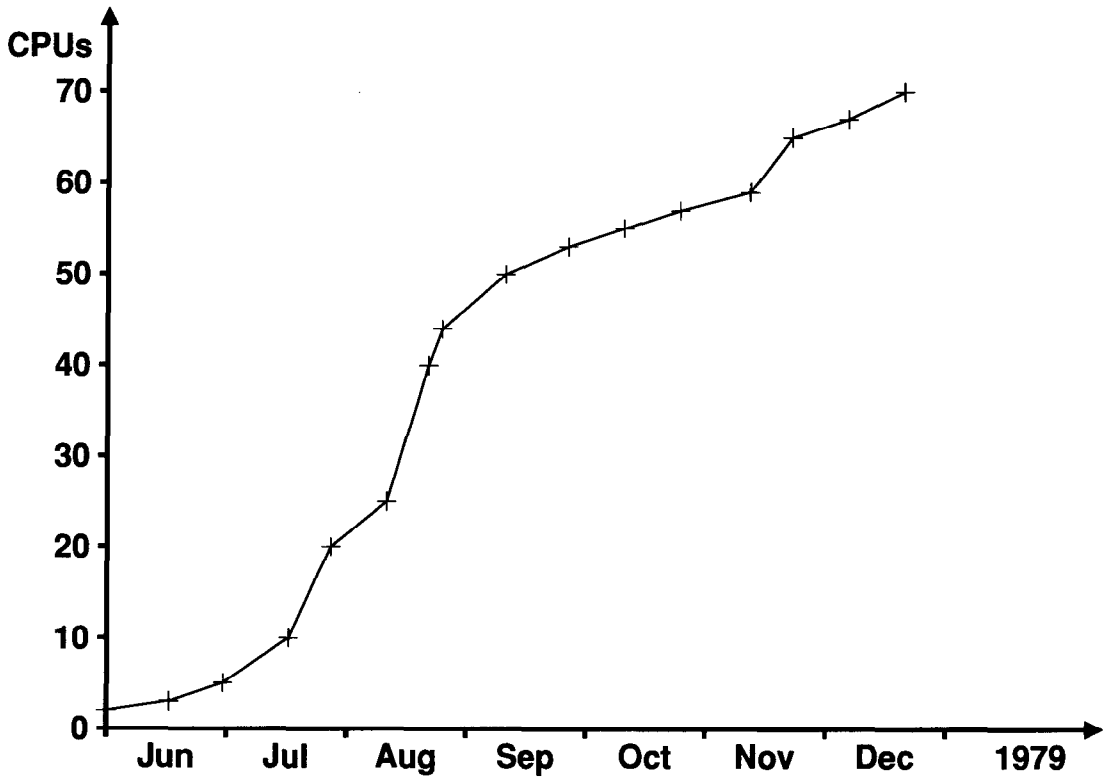
  "REX is getting some really good press around here. People really sit up and take notice, but wonder why someone didn't do it 30 years ago"

Development and usage report

"The value of this communication with other programmers and users cannot be underestimated. Without the communications provided by the network, REX would never have been developed."

- ✦ 10,000 lines of assembler, 5,000 of documentation
- ✦ 27 man-weeks (1000 hours)
- ✦ Only evenings and weekends—when response time was good and interruptions were few.

Growth chart



Retrospective—design errors

- ✦ Comparison should have been case-insensitive
- ✦ DO should have been split into DO...END and LOOP...END
- ✦ Too much emphasis in the External Data Queue
- ✦ Parsing is something of a compromise

Retrospective—successes

- ✦ Deliberate minimizing of "boilerplate" and punctuation, and notations in general
- ✦ Hardware independence and robustness
- ✦ Upgradeable language (keywords only reserved in context)
- ✦ String support (especially "blank operator")
- ✦ Associative arrays (stems)
- ✦ Decimal arithmetic
- ✦ Use of the electronic network for rapid design evolution

1980-1984

- 30 internal releases
- Customers, led by **SLAC**, ask for REX
- Name changed to REXX
- VM/SP 3, with REXX, announced and shipped world-wide (1983)

Help!

There are some **omissions** in the following.

Please let me know of them (and any corrections)—I'll incorporate in a WWW page soon.

2.5

1985-1988

- First non-IBM implementation (Charles Daney, 1985)
- *The Rexx Language* published (1985)
- First Unix implementation (Andy Pierce, IBM, 1985)
- Experimental OS/2 implementation (1986)
- Rexx for VMS VAX (Charles Daney, 1986?)
- IBM SAA has Rexx as "Procedures Language" (1987)
- Amiga Rexx (AREXX, Bill Hawes, 1987)
- Rexx in MVS and TSO/E (1988)
- T-REXX for Tandem (Keith Watts, 1988?)

1989-1990

- IBM and Microsoft agree Rexx is the best scripting language for OS/2 (1989)
- Rexx compiler for VM (IBM Haifa and Vienna, 1989)
- uni-Rexx (The Workstation Group, 1989)
- Rexx 4.00 published (1990)
- First Rexx Symposium (SLAC, 1990)
- Rexx in AS/400 (1990)
- Rexx in OS/2 (1990)

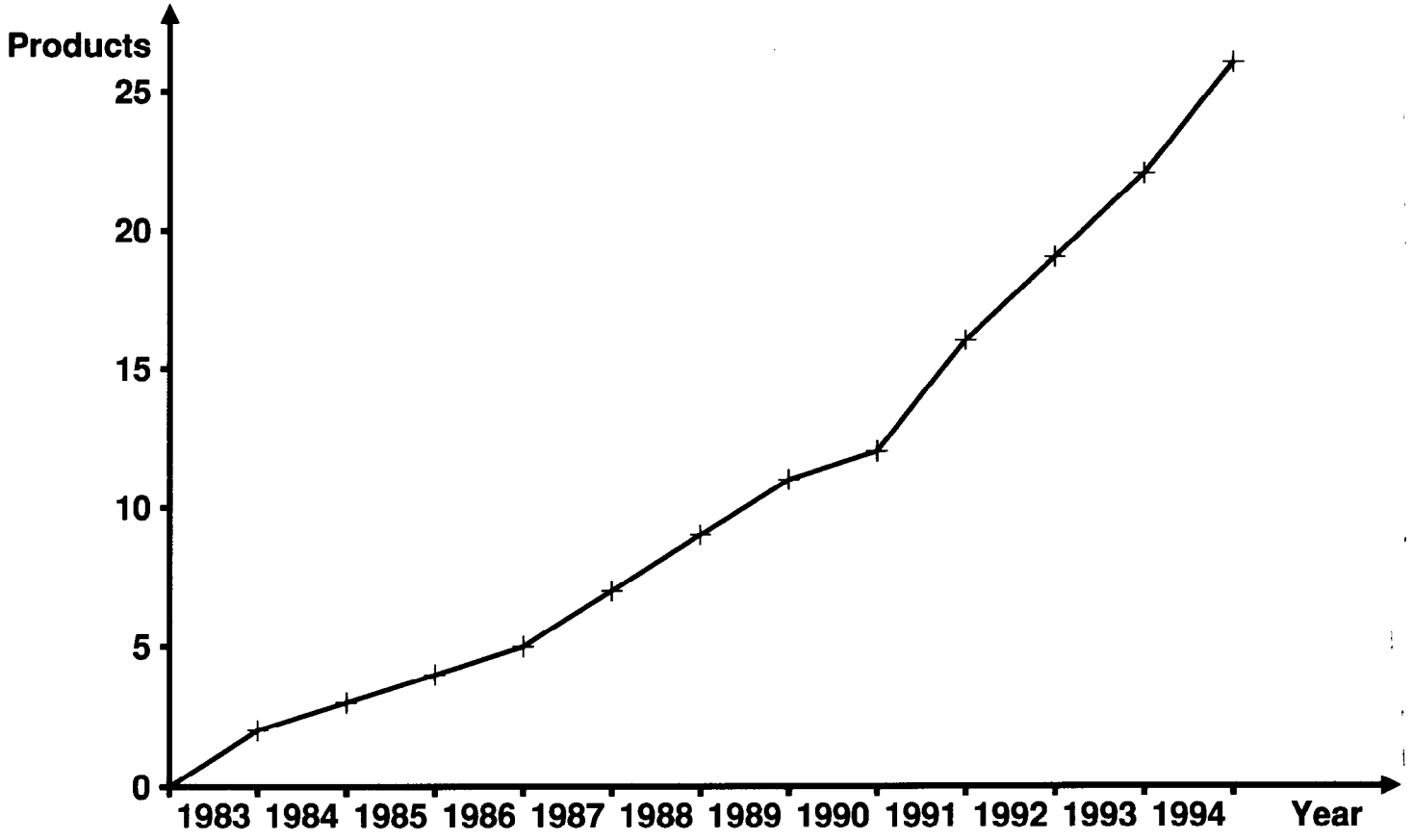
1991-1994

- Work on ANSI standard for Rexx starts (1991)
- Rexx/imc (Ian Collier, 1992)
- Regina Rexx (Anders Christensen, 1993)
- Rexx for VSE (1993)
- Rexx for AIX/6000 (1993)
- Rexx Language Association formed (1994)
- Rexx for Novell NetWare (1994)
- Simware Rexx; Windows, Macintosh, NetWare (1994)
- Rexx for CICS/ESA (1994)

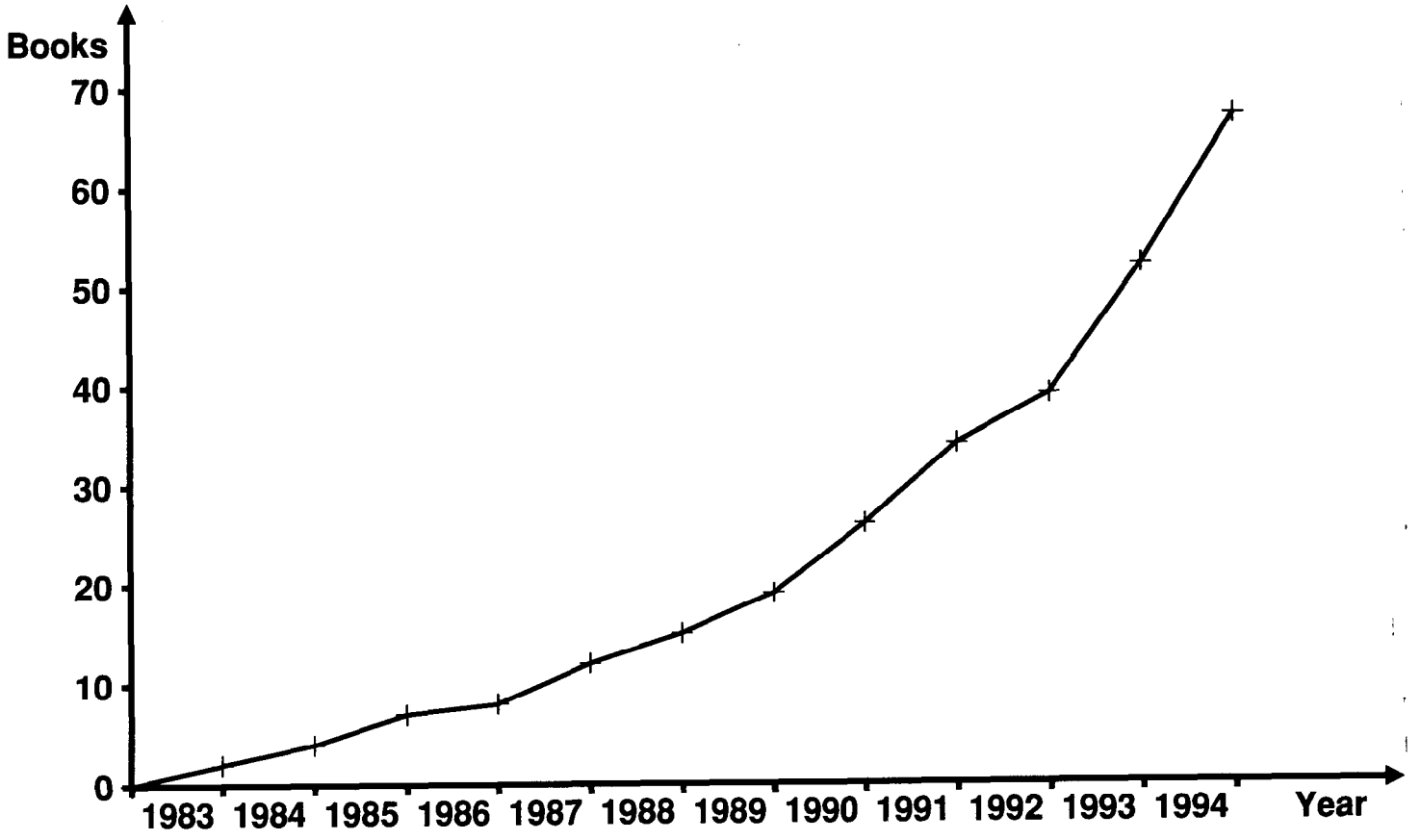
1995

- Rexx in PC-DOS 7, as the "programming language of choice"
- World-Wide Web pages for Rexx; start at:
  - → `http://rexx.hursley.ibm.com/rexx/`
- Object Rexx public beta
- ...and more...

REXX Language Products Available



REXX Books and Manuals



Summary

- ✦ Rexx is a carefully designed, purpose-built scripting language
- ✦ Steady growth over 15 years, especially rapid in last 2-3 years
- ✦ Rexx is installed on 15-25 million users' machines
- ✦ Well over 2 million Rexx programmers
- ✦ It wouldn't have been possible without **people**.