HIDDEN MAINSTREAM: THE MAINFRAME LANGUAGES

Dr. Vadim Zaytsev

Workshop on Programming Research in Mainstream Languages
(PRiML @ LICS/ICALP 2020)
• Worked in research (CWI, VU, Koblenz)
  • software evolution
  • software languages (PL+)
  • grammars in a broad sense
• Worked in industry (Raincode, Raincode Labs)
  • legacy systems
  • software migration
  • mainframe to cloud native
• http://grammarware.net,
  http://twitter.com/grammarware,
  …
IBM S/360

In use since 1964
MAINFRAME LANGUAGES
COBOL

- Language designed by committee
  - tech advisor: Grace Murray Hopper
- Based on ideas of Grace Hopper
  - and Bob Bemer (COMTRAN)
- Extremely verbose
  - looks like English
  - surprisingly readable!
- Hard to find empirical fodder
  - little open source
  - closed source portfolios are huge
- Hard to implement
  - large grammar
  - complex semantics
MOVE INPUT-PARM OF JCL(1:LL OF JCL) TO BOOKING-DATE

MVC '49C' (8,1), '4A4' (2)

MAP STD_PARM_V OF ACC Xxx I TO CASH_ACT_UPD

D2 07 14 9C 24 A4
“The MARK IV System Engineer had just completed making the installation when a VP came in asking for a special report. The MARK IV S.E. defined existing files to MARK IV, keypunched the request, and had the report 10 minutes after it had been requested.”

1960s

"We assigned a programmer, one of our sharpest, to a job which looked like a job best done in COBOL. The programmer was told to do the job in COBOL and was given six months for completion. On his own, the programmer secretly did the job using MARK IV and completed the job in 3 weeks.”

MARK IV

FILE MANAGEMENT SYSTEM

The general purpose software product line for business data processing
Users say Pacbase worth effort

BY ALAN ALPER
CW STAFF

PEARL RIVER, N.Y. — Three users of Pacbase, an application generator developed 15 years ago by CGI-Informatique in Paris, are finding that the system saves development time and maintenance costs, even if it is somewhat difficult to learn.

Marketed in the U.S. by CGI Systems, Inc., located here, the system is based on the Merise methodology, a structured programming technique popular in France. The system runs on IBM mainframes under DOS, DOS/VSE and MVS as well as on Honeywell, Inc. and Unisys Corp. large-scale systems. It supports most teleprocessing monitors and database management systems, including IBM's DB2.

Pacbase is installed at 500 sites worldwide, including 90 in the U.S., the company notes. CGI has stepped up its U.S. presence, and its U.S. revenue is expected to exceed revenue derive from other countries this year for the first time. Worldwide revenue was $77 million last year, CGI says.

Pacbase has evolved from an
Continued on page 23

1970s
1980s

- 12 pages of COBOL
- 2 pages of Mark IV
- 1 statement in Nomad

James Martin, Applications Development Without Programmers, 1981
Users say Pacbase worth effort

BY ALAN ALPER

PEARL RIVER, N.Y. Three
users of Pacbase, an applica-
tion generator developed by
CGI-Infogrames, and last year,
 Pacbase has evolved from an
IBM VisualAge Pacbase
Continued on page 23

2020?  

STILL ALIVE  

as

CA VISION: BUILDER  
The general purpose software product line  
for business data processing

STILL ALIVE as

IBM VisualAge Pacbase
WHAT ABOUT COBOL?

• 43% of banking systems built on COBOL [Reuters 2017]
• 15% of new applications built in COBOL [Gartner 2003]
• 75% of business data processed by COBOL [Gartner 2003]
• 80% of in-person transactions run COBOL [Reuters 2017]
• 95% ATM swipes rely on COBOL code [Reuters 2017]
• 180–200 billion LOC of COBOL in use [Gartner 2003]
  • 220 billion LOC in use [Reuters 2017]
  • one codebase up to 250 MLOC [Bankia’20], 343 MLOC [NYMellon’12]
• Replacement costs at $25 per line [Tactical Strategy Group]
• 1 COBOL app costs $5M/year [Micro Focus 2003]
  • 4000 MIPS will cost $6–16M/year [Raincode 2020]

https://thenewstack.io/cobol-everywhere-will-maintain/ (2017),
EXPECTATIONS

REALITY
COBOL QUOTES:

- “it's only a matter of time before all the existing COBOL programmers die of old age” [Yourdon 1996]
- “maybe it will all be outsourced to some part of the world where COBOL maintenance programming is considered a pleasant alternative to growing rice or raising pigs” [Yourdon 1996]
- “The use of COBOL cripples the mind; its teaching should, therefore, be regarded as a criminal offence.” [Dijkstra 1975]
- However, research on COBOL is not! [Lämmel 2004]

TAKEAWAY #1:

• Legacy languages matter
• Software written in them runs your life
• 2GLs like HLASM
• 3GLs like COBOL, PL/I, CLIST, REXX, RPG, FORTRAN
• 4GLs like Pacbase, AppBuilder, IDEAL, VISION:BUILDER
LANGUAGE FEATURES TO BE AFRAID OF
Language features to be afraid of:
C
C++
C#
Java
JavaScript
...

https://www.ioccc.org/2019/dogon/prog.c
[Dogon, IOCCC 2019]
\[
f(x) = \text{let } a = w(x) \\text{ in if } \text{cond1}(x) \\text{ then } a \\text{ else if } \text{cond2}(x) \\text{ then } g(a) \\text{ else } f(h(x), a)
\]

**Align Homogeneous Parts**

- Haskell
- Kotlin

- [Landin 1966]


```python
i = 1
while i < 6:
    print(i)
    i += 1
else:
    print("error")
```

**Align all blocks**

- Python
MACRO
LCLC &KAF
LCLA &COS
LCLA &TZT

&N1 MOVE &T,&F
AIF (T'&T NE T'&F).END

&KAF SETC 'fHUx'.(3)'X'.'th8LFqnqiexQvPD12RwUA'.7PVUPZABNmQwOCkr'*
(9,13)

&TZT SETA 872+354/469
AIF (T'&T NE 'F').END

&COS SETA &TZT

&N2 ST 2,SAVEAREA
L 2,&F
LCLB &EOT
ST 2,&T
L 2,SAVEAREA
ANOP

.END MEND
LINE CONTINUATIONS IN COBOL

000001              "AAAAAAAAAAABBBBBABBBCCCCCCCCCCDDDDDDDDDEEEEEEEEEE
                      "GGGGGGGGGGGGHHHHHHHHHHHHHHHHHHHHIIIIIIIIIIIIJJJJJJJJJJJJJJJJJJKKKKKKK
                      "LLLLLLLLLLMMMMMMMMMM"

000003              N"AAAAAAAAAAABBBBBABBBCCCCCCCCCCDDDDDDDDDEEEEEEEEEE
                      "GGGGGGGGGG"

000005              "AAAAAAAAAAABBBBBABBBCCCCCCCCCCDDDDDDDDDEEEEEEEEEE
                      "GGGGGGGGGGGHHHHHHHHHHHHHHHHHHIIIIIIIIIIIIJJJJJJJJJJJJJJJJJJKKKKKKK
                      "LLLLLLLLLLMMMMMMMMMM"

000010              "AAAAAAAAAAABBBBBABBBCCCCCCCCCCDDDDDDDDDEEEEEEEEEE"
                      "GGGGGGGGGGGGHHHHHHHHHHHHHHHHHHIIIIIIIIIIIIJJJJJJJJJJJJJJJJJJKKKKKKK
                      "LLLLLLLLLLMMMMMMMMMM"
TAKEAWAY #2:

- Parsing is “solved”
- Indentation is solved ad hoc
- There is no line continuation calculus

Example @ GPCE 2017 ➔
Language features to be afraid of:

- INDENTATION
- NAMING
Typed names — implicit typing in FORTRAN

```
IMPLICIT REAL (A-Z)
IMPLICIT INTEGER (I-N)
ADJUSTMENT = 1.0E5
NAME = 42
```
hole. = "empty"
hole.9 = "full"
hole.rat = "full"
rat = "cheese"
drop hole.rat
say hole.1 hole.mouse hole.9 hole.rat

empty empty full HOLE.cheese
NAMES ARE NOT UNIQUE IN APPBUILDER
IF THEN = ELSE
THEN ELSE = IF
ELSE IF = THEN
END;
CONTRACTIONS IN COBOL

MOVE 42 TO X OF Y

IF X > 0 AND Y = Z OR X = Z THEN DISPLAY X END.

IF X > 0 AND X = Y OR X = Z THEN DISPLAY X END.
Language features to be afraid of:

- INDENTATION
- NAMING
- LEXICAL IMPORTS
Imports in Haskell

import Data.Maybe
import qualifying Data.Maybe
import Data.Maybe hiding maybeToList
**Imports in Python**

```python
import os.path
import lxml.etree as ET
from library import *
```
IMPORTS IN COBOL

COPY PRIMLLI
REPLACING == STD-FSSH-INIT == BY == STD-FSSHH-INIT ==.
COPY PRIMLLI
REPLACING == STD-FSSH-CHKPT == BY == STD-FSSHH-CHKPT ==.
COPY PRIMLLI
REPLACING == STD-FSSH-END == BY == STD-FSSHH-END ==.
COPY PRIMLLI
REPLACING == STD-FSSH-ROLLBACK == BY == STD-FSSHH-ROLLBACK ==.
COPY PRIMLLI
REPLACING == STD-FSSH-COMMIT == BY == STD-FSSHH-COMMIT ==.
Language features to be afraid of:

- Indentation
- Naming
- Lexical imports
- Transfer of control
GO TO considered harmful? TRY ALTER!
PROCEDURE DIVISION.

...

EXIT-ON-ERROR.
  GO TO EXIT-UPDATE-RECORD.
EXIT-UPDATE-RECORD.
  ...
EXIT-ROLLBACK-RECORD.
  ...
PROCEDURE DIVISION.

ALTER EXIT-ON-ERROR TO EXIT-ROLLBACK-RECORD.

EXIT-ON-ERROR.
    GO TO EXIT-ROLLBACK-RECORD.
EXIT-UPDATE-RECORD.
    ...
EXIT-ROLLBACK-RECORD.
    ...
Want more?

- Semi-structured transfer of control
  - `GO TO + ALTER` (COBOL)
  - `GO TO (...) , X` (FORTRAN)
  - `EX` (HLASM)

- Structured transfer of control
  - `NEXT SENTENCE` (COBOL)
  - `PERFORM THRU` (COBOL)
  - `DO` (AppBuilder)
  - `SIGNAL` (REXX)
TAKEAWAY #3:

- Read the documentation carefully!
- Expect the unexpected
  - and still be surprised
- More new challenges = good!
- Example @ PX/17.2
CONCLUSION

• Your life is run by COBOL
  • banking, booking, ordering, …
• BabyCobol is a WIP
  • http://slebok.github.io/baby
• Mainframe languages are great
  • hide many challenges
• Follow @grammarware!