ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Babele è stata davvero una maledizione?

La molteplicità dei linguaggi di programmazione

Simone Martini

Dipartimento di Informatica-Scienza e Ingegneria

Accademia delle Scienze dell’Istituto di Bologna

Matematica, Fisica e Informatica nel secolo XX:
L’ossessione della totalità
Accademia delle Scienze di Torino, 19 marzo 2024

What is a programming language?

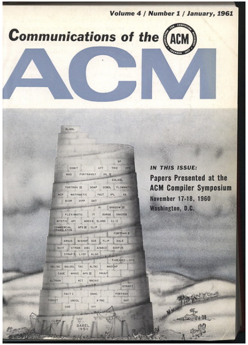
An artificial language used to write instructions that can be translated into machine language and then executed by a computer.

[THE AMERICAN HERITAGE SCIENCE DICTIONARY, ©2011]

How many programming languages are there?

Thousands... Several dozen currently in use

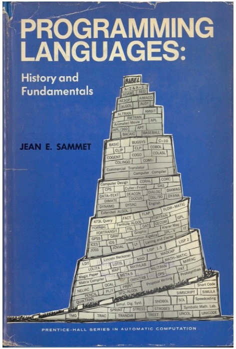
The Babel of programming languages


Volume 4 / Number 1 / January, 1961
Communications of the ACM
ACM
IN THIS ISSUE:
Papers Presented at the ACM Compiler Symposium
November 17-18, 1960
Washington, D.C.
BABEL 1960

1961

IFIP
WORKING CONFERENCE VIENNA 1964
FORMAL LANGUAGE DESCRIPTION LANGUAGES

1964



1969

Babel vs Universal: Saul Gorn (1912-1992)

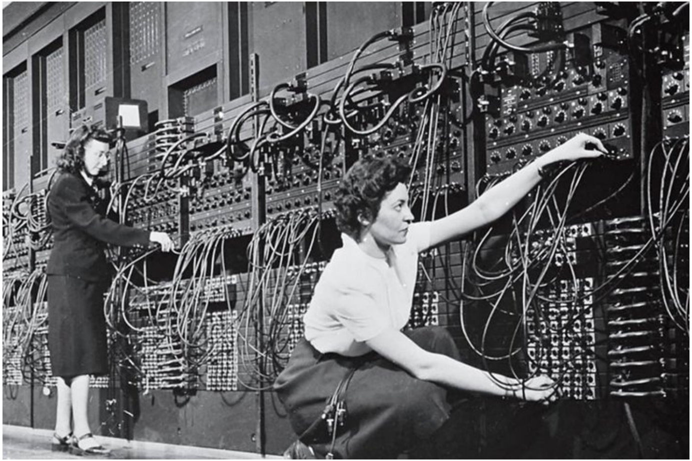
A simple enough "universal code" to be used by "computers, data processers, production engineers, traffic controllers, or administrators of large companies."

[Planning universal semi-automatic coding, 1954]

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

The first proliferation: machines

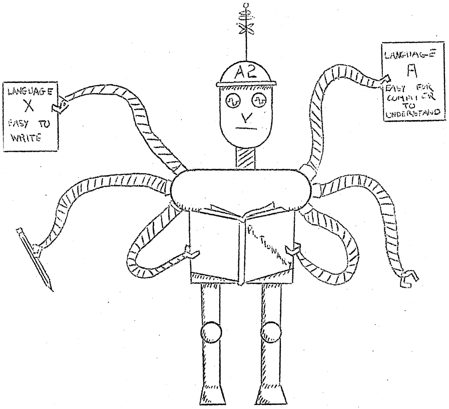
ENIAC: 1947



One machine, one code

Technology, bound to the specific machine

Translation: Grace M. Hopper

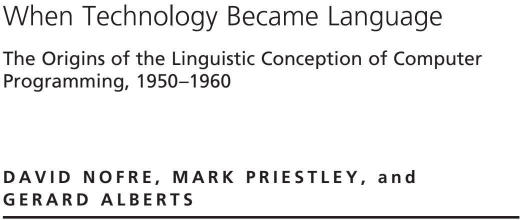


TRANSLATION PHASE

HS 10/23/53

[Digital computer advanced coding techniques, ©MIT, 1954]

Towards the language metaphor



When Technology Became Language

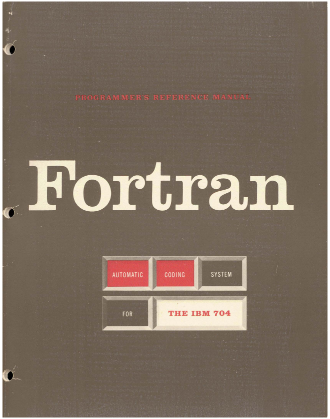
The Origins of the Linguistic Conception of Computer Programming, 1950–1960

DAVID NOFRE, MARK PRIESTLEY, and GERARD ALBERTS

Technology and Culture, Vol 55, January 2014.

The second proliferation: domains

Algebraic expressions: FORTRAN



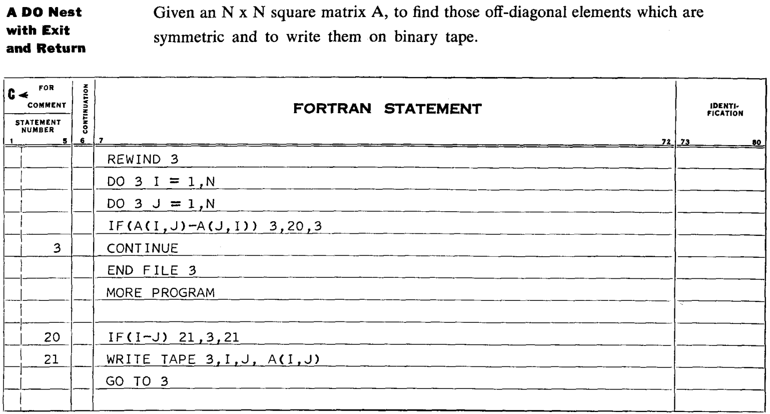
©IBM, 1956

Algebraic expressions, e.g.

D=(A+B)*C-sin(A*C+2)

translated into efficient object programs

Simple control structures: FORTRAN



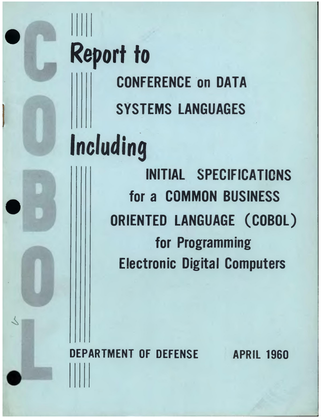
A DO Nest with Exit and Return

Given an N x N square matrix A, to find those off-diagonal elements which are symmetric and to write them on binary tape.

C ← FOR COMMENT STATEMENT NUMBER 1 5	CONTINUATION 6	FORTRAN STATEMENT 7 72	IDENTIFICATION 73 80
		REWIND 3	
		DO 3 I = 1,N	
		DO 3 J = 1,N	
		IF(A(I,J)-A(J,I)) 3,20,3	
3		CONTINUE	
		END FILE 3	
		MORE PROGRAM	
20		IF(I-J) 21,3,21	
21		WRITE TAPE 3,I,J, A(I,J)	
		GO TO 3	

©IBM, 1956

Records and English: COBOL



COBOL

Report to

CONFERENCE on DATA
SYSTEMS LANGUAGES

Including

INITIAL SPECIFICATIONS
for a COMMON BUSINESS
ORIENTED LANGUAGE (COBOL)
for Programming
Electronic Digital Computers

DEPARTMENT OF DEFENSE APRIL 1960

©US Dept of Defense, 1960

Collections of non-numerical data

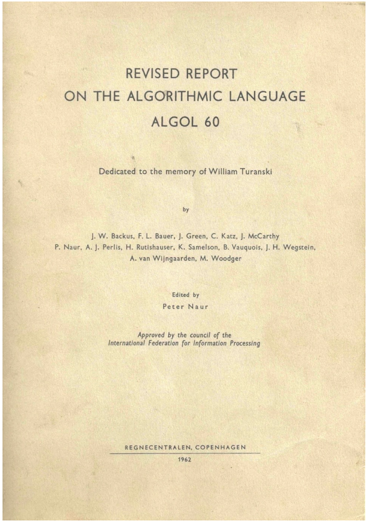
English words

Records and English: COBOL

```
FD  CUSTOMER-FILE
    RECORD CONTAINS 45 CHARACTERS.
01  CUSTOMER-RECORD.
    05  CUSTOMER-NAME.
        10 LAST-NAME        PIC X(17).
        10 FILLER           PIC X.
        10 INITIALS         PIC XX.
    05  PART-ORDER.
        10 PART-NAME        PIC X(15).
        10 PART-COLOR       PIC X(10).
```

```
IF LAST-NAME = PART-NAME GO TO PARAGRAPH 1 ELSE
MOVE PART-NAME TO LAST-NAME
```

Algorithms: ALGOL



REVISED REPORT
ON THE ALGORITHMIC LANGUAGE
ALGOL 60

Dedicated to the memory of William Turanski

by

J. W. Backus, F. L. Bauer, J. Green, C. Katz, J. McCarthy
P. Naur, A. J. Perlis, H. Rutishauser, K. Samelson, B. Vauquois, J. H. Wegstein,
A. van Wijngaarden, M. Woodger

Edited by
Peter Naur

Approved by the council of the
International Federation for Information Processing

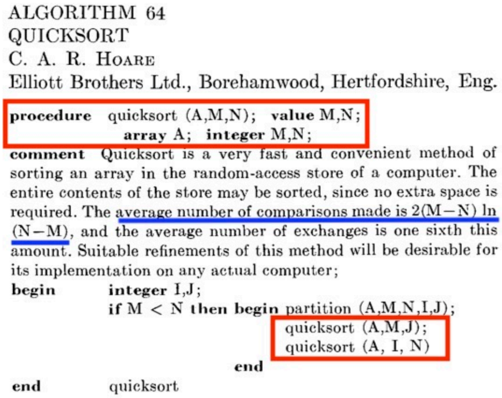
REGNECENTRALEN, COPENHAGEN
1962

©IFIP, 1962

Universal language for algorithm exchange

International committee

Algorithms: ALGOL



ALGORITHM 64
QUICKSORT
C. A. R. Hoare
Elliott Brothers Ltd., Borehamwood, Hertfordshire, Eng.

```
procedure   quicksort (A,M,N);  value M,N;
                 array A;  integer M,N;
comment  Quicksort is a very fast and convenient method of
sorting an array in the random-access store of a computer. The
entire contents of the store may be sorted, since no extra space is
required. The average number of comparisons made is 2(M−N) ln
(N−M), and the average number of exchanges is one sixth this
amount. Suitable refinements of this method will be desirable for
its implementation on any actual computer;
begin       integer I,J;
            if M < N then begin partition (A,M,N,I,J);
                                 quicksort (A,M,J);
                                 quicksort (A, I, N)
                           end
end         quicksort
```

Rich data types: Tony Hoare

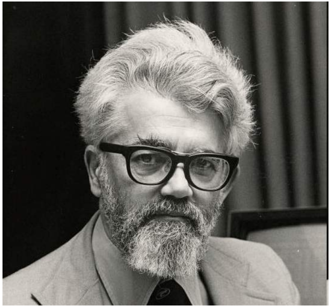
C.A.R. Hoare, 1934-

Modelling tool:

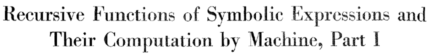
In the simulation of complex situations in the real world, it is necessary to construct in the computer analogues of the objects of the real world

[Hoare, Record handling, 1965]

Symbolic structures: LISP

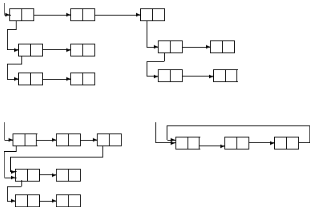


John McCarthy, 1927-2011



Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I

Communications of the ACM Volume 3, April 1960



Symbolic structures: LISP

```
eval[e; a] = [
 atom[e] → cdr[assoc[e;a]];
 atom[car[e]] → [eq[car[e]; QUOTE] → cadr[e]];
                 eq[car[e]; COND] → evcon[cdr[e];a];
                 T → apply[car[e]; evlis[cdr[e];a];a]];
 T → apply[car[e]; evlis[cdr[e];a];a]]
```

Different domains, different machines

Scientific: FORTRAN, on IBM the 7090 and the IBM 1620

Business: COBOL, on the IBM 7080 and the IBM 1401

Real-time: JOVIAL, on the IBM 7750 and IBM 7950 (Harvest)

Abstraction over the machine: Hoare

It was a firm principle of our implementation that the results of any program, even erroneous, should be comprehensible without knowing anything about the machine or its storage layout.

[Hoare, personal communication, 2014]

Taking stock, 1

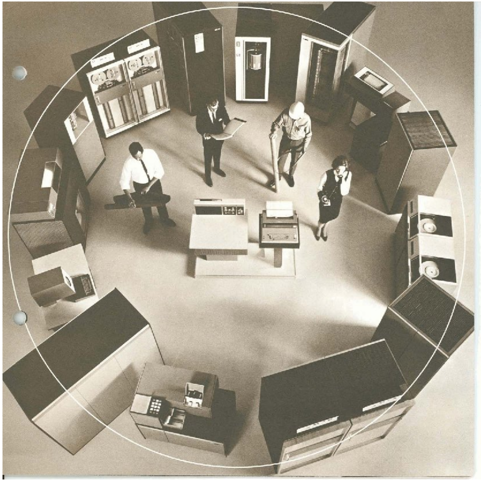
1. PLs do not give instructions to the (physical) machine: they hide it.

2. PLs are sets of abstraction mechanisms,
   - over control (structured control, procedures)
   - and data (data types).

3. Programs are abstract, computational models of "the real world" (cf Hoare).

The myth of the total language

IBM System/360, 1964 ff



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

The ultimate language

One machine, one language: for all

- users need to learn only one language
- only one compiler to be maintained
- programs could be easily shared

"A universal programming language that would meld and displace FORTRAN and COBOL"

[Brooks and Shustek, 2015]

PL/I: Some design choices

- Anything goes

  "If a particular combination of symbols has
  a reasonably sensible meaning,
  that meaning will be made official"
- Full access to machine and operating system
- Cater to the novice

[G. Radin, H.P. Rogoway. NPL: Highlights of A New Programming Language. CACM 8(1), 1965]

PL/I: An inconsistent model

PL/I: An inconsistent model

PL/I: defeat

Different domains raise
different classes of problems that require
different sets of representations.

PL/I was designed in order to forget about such peculiarities.

Other driving forces: correctness

Algol's research programme:

a (Kuhn) paradigm for programming language design, and

correct software development.

A language for the new science

Ada: 1977 ff

Designed *for* the US Department of Defence:

- concurrency
- real-time
- embedded computing
- life-critical applications
- reliability
- formal definition
- simplicity

Ada, the last total language

Jean Ichbiah (Ada’s main designer):

In ten years from now [scil. 1979-80]*, only two programming languages will remain: Ada and Lisp.*

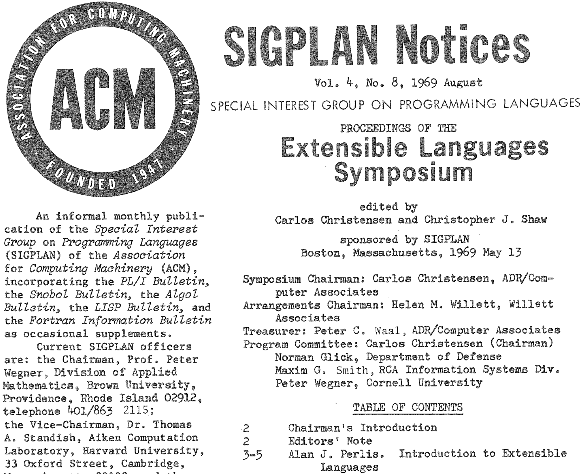
[according to Rosen, The Ada paradox(es), Ada Letters 24, 2009]

Another attempt to universality

Total language

Total language vs. Extensible language

Extensible languages



An informal monthly publication of the *Special Interest Group* on *Programming Languages* (SIGPLAN) of the *Association for Computing Machinery* (ACM), incorporating the *PL/I Bulletin*, the *Snobol Bulletin*, the *Algol Bulletin*, the *LISP Bulletin*, and the *Fortran Information Bulletin* as occasional supplements.

Current SIGPLAN officers are: the Chairman, Prof. Peter Wegner, Division of Applied Mathematics, Brown University, Providence, Rhode Island 02912, telephone 401/863 2115; the Vice-Chairman, Dr. Thomas A. Standish, Aiken Computation Laboratory, Harvard University, 33 Oxford Street, Cambridge,

SIGPLAN Notices

Vol. 4, No. 8, 1969 August

SPECIAL INTEREST GROUP ON PROGRAMMING LANGUAGES

PROCEEDINGS OF THE

Extensible Languages Symposium

edited by
Carlos Christensen and Christopher J. Shaw

sponsored by SIGPLAN
Boston, Massachusetts, 1969 May 13

Symposium Chairman: Carlos Christensen, ADR/Computer Associates
Arrangements Chairman: Helen M. Willett, Willett Associates
Treasurer: Peter C. Waal, ADR/Computer Associates
Program Committee: Carlos Christensen (Chairman)
Norman Glick, Department of Defense
Maxim G. Smith, RCA Information Systems Div.
Peter Wegner, Cornell University

TABLE OF CONTENTS

An assessment on extensible languages, 1975

*Extending a simple base results often in long,
thin extension cascades that are often ugly and inefficient.*

[Standish, Extensibility in programming language design. AFIPS 1975]

An assessment on extensible languages, 1975

*Extending a simple base results often in long,
thin extension cascades that are often ugly and inefficient.*

[Standish, Extensibility in programming language design. AFIPS 1975]

Buy instead of build

Nice try, though

Total languages are *closed* (technical) objects

Extensible languages are (more) *open* (technical) objects

Taking stock, again

Taking stock, 1 (again)

1. PLs do not give instructions to the (physical) machine: they hide it.

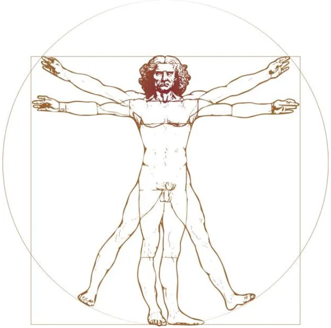
2. PLs are sets of abstraction mechanisms,
   - over control (structured control, procedures)
   - and data (data types).

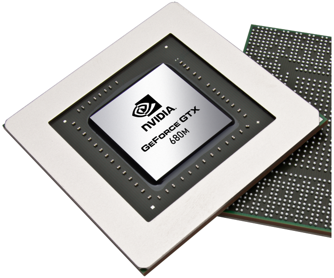
3. Programs are abstract, computational models of "the real world" (cf Hoare).

Taking stock, 2

4. Each PL has its own “embedded” model
5. Different domains need different models
6. External drivers

An interesting pluralism: Languages as mediators





Complete relativism?

No criteria for discernment?

Let’s take the “language metaphor” seriously:
a PL is a medium for dialogue with the machine

Let’s take the “language metaphor” seriously:
a PL is a medium for dialogue with the machine

The machine is a source of alienation

The criteria: reduce alienation

Gilbert Simondon

Gilbert Simondon, 1924-1989

Les objets techniques qui produisent le plus d'aliénation sont ceux qui sont destinés à des utilisateurs ignorants.

[Du mode d'existence des objets techniques, 1958]

Open and closed technical objects

Closed technical object

- its user does not understand how and why it works
- it cannot be repaired
- it is unmodifiable
- it evokes the sacred, the untouchable

Open and closed technical objects

Closed technical object

- its user does not understand how and why it works
- it cannot be repaired
- it is unmodifiable
- it evokes the sacred, the untouchable

Open technical object

- its user knows how it works, and how it could be repaired
- “to be” instead of “to appearing” (être et ne pas paraître)
- it shows the traces of its own evolution

“Open” programming languages

Let everyone be allowed to use a language that suits them

A language that reveals and mediates the machine within the limits, aspirations, and competences of that user

Such a language can reduce their alienation

Languages as extensible systems

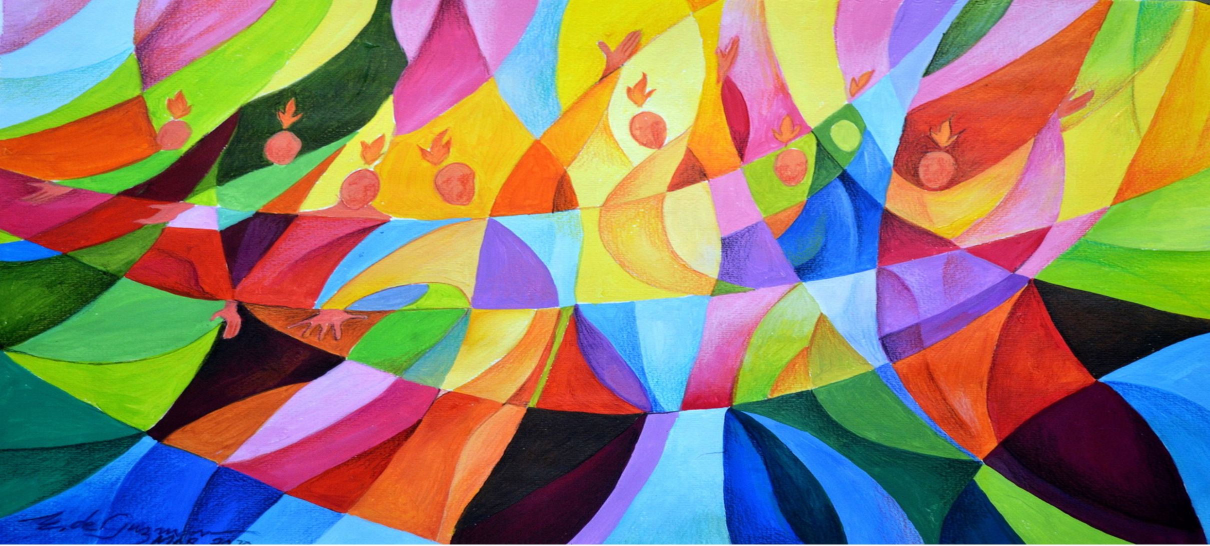
Logo: S. Papert et al., 1967. BBN, MIT

Smalltalk: A. Kay et al., 1975. Xerox PARC

Per conoscere il mondo bisogna costruirlo

Cesare Pavese, Il mestiere di vivere. 1952

Babel?



©E. De Guzman, 2014

ἤκουον εἷς ἕκαστος τῇ ἰδίᾳ διαλέκτῳ λαλούντων αὐτῶν·

[At. 2, 6]

Babel was the contrary of a curse.
The gift of tongues is precisely that;
a gift and benediction beyond reckoning.

[G. Steiner. Errata. 1998 (p. 99)]

Babel was the contrary of a curse.
The gift of tongues is precisely that;
a gift and benediction beyond reckoning.

[G. Steiner. Errata. 1998 (p. 99)]

How monotone must love-making have been in Paradise.

[(p. 102)]

Babel was the contrary of a curse.
The gift of tongues is precisely that;
a gift and benediction beyond reckoning.

[G. Steiner. Errata. 1998 (p. 99)]

How monotone must ~~love-making~~ have been in Paradise.
programming

[(p. 102)]