Programming Languages as Designed Objects

4S / EASST
Rotterdam, Netherlands

stuart mawler
2008/08/22

Multitudes...

- There are literally hundreds of different languages
- Why so many?
- Why one versus another?
- Is it language “evolution”?
- How does Science & Technology Studies approach programming language design and selection?

Hello World!

- A common introduction to a new language
- A simple program that displays the text “Hello World!”
  - Possibly the most trivial example of programming...

Java

```
class HelloWorld {
        public static void main (String args[]) {
        for (;;) {
                 System.out.print("Hello World ");
                 }
        }
}
```

- All code samples from:
  - The ACM Hello World Project. Available online at http://www2.latech.edu/~acm/HelloWorld.shtml, accessed 2008/03/31.

Practitioners & Partisans

Ada

```
with Text_Io; use Text_Io;

   procdure hello is
   begin
      put ("Hello world!");
   end hello;
```

Algol

```
BEGIN
FILE F (KIND=REMOTE);
EBCDIC ARRAY E [0:11];
REPLACE E BY "HELLO WORLD!";
WHILE TRUE DO
    BEGIN
    WRITE (F, *, E);
    END;
END.
```

Occam

```
-- occam
PROC write.string(CHAN output, VALUE string[])=
  SEQ character.number = [1 FOR string[BYTE 0]]
    output ! string[BYTE character.number]

write.string(terminal.screen, "Hello World!")
```

Practitioners & Partisans

- IT Trade Literature
  - “General-purpose languages [...] are usually created either to address existing languages' inadequacies, to fill some business need, or both” (Mark Johnson, JavaWorld.com, 2000)
    - Technical objectives and market forces
    - Hidden debate
      - Language categories
      - Definition of “inadequacies”
      - Organizational concerns
- Fan Literature
  - Proliferation alone shows partisanship
    - Corrections to the Hello World! entry for Algol

Computer Science

- Technical literature downplays the social
  - Even “comprehensibility” is technical, rather than social
- Authoritative Reminiscences
  - Place social motivations outside the scope of designers (e.g., Sun & Java)
- Actual histories (internalist)
  - Case of C++
  - Goals for the maintenance of C++: to “keep the language from fragmenting into dialects” and to “keep the language and its community from stagnating” (Bjarne Stroustrup , HOPL III, 2007)
  - However, retains active denial of the social—success is purely technical

Computer Science – Critics & Visionaries

- Elegance
  - Analogy with engineering (Bruce MacLennan, 1997)
    - *The Tower and the Bridge*, David P. Billington
  - Promote a “set of values” (MacLennan)
  - The exception, rather than the rule

Hello World!

BASIC

```
10 print"Hello World!"
20 goto 10
```

Lisp

```
; LISP
(DEFUN HELLO-WORLD ()
                    (PRINT (LIST 'HELLO 'WORLD)))
```

C++

```
#include <iostream>

int main()
{
    std::cout << "Hello, world!\n";
}
```

Dijkstra on BASIC

“It is practically impossible to teach good programming to students that have had prior exposure to BASIC: as potential programmers they are mentally mutilated beyond hope of regeneration.”

Dijkstra, Edsger, “How do we tell truths that might hurt?” ACM Technical Note EWD498 (18 June 1975)

Hello World!

COBOL

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID.     HELLOWORLD.
000300 DATE-WRITTEN.   02/05/96        21:04.
000400*       AUTHOR    BRIAN COLLINS
000500 ENVIRONMENT DIVISION.
000600 CONFIGURATION SECTION.
000700 SOURCE-COMPUTER. RM-COBOL.
000800 OBJECT-COMPUTER. RM-COBOL.
000900
001000 DATA DIVISION.
001100 FILE SECTION.
001200
100000 PROCEDURE DIVISION.
100100
100200 MAIN-LOGIC SECTION.
100300 BEGIN.
100400     DISPLAY " " LINE 1 POSITION 1 ERASE EOS.
100500     DISPLAY "HELLO, WORLD." LINE 15 POSITION 10.
100600     STOP RUN.
100700 MAIN-LOGIC-EXIT.
100800     EXIT.
```

Dijkstra on COBOL

“The use of COBOL cripples the mind; its teaching should, therefore, be regarded as a criminal offense.”

Dijkstra, Edsger, “How do we tell truths that might hurt?” ACM Technical Note EWD498 (18 June 1975)

Hello World!

Pascal

```
Program Hello (Input, Output);

Begin
  Writeln ('Hello World!');
End.
```

Fortran

```
c
c   Hello, world.
c
      Program Hello

      implicit none
      logical DONE

      DO while (.NOT. DONE)
        write(*,10)
      END DO
   10 format('Hello, world.')
      END
```

Dijkstra on Fortran

“‘The infantile disorder’—, by now nearly 20 years old, is hopelessly inadequate for whatever computer application you have in mind today: it is now too clumsy, too risky, and too expensive to use.”

Dijkstra, Edsger, “How do we tell truths that might hurt?” ACM Technical Note EWD498 (18 June 1975)

Eye of the Beholder: Style & Elegance

Smalltalk

```
Transcript show:'Hello World';cr
```

Lisp

```
; LISP
(DEFUN HELLO-WORLD ()
                  (PRINT (LIST 'HELLO 'WORLD)))
```

Eiffel

```
class HELLO_WORLD

creation make

feature

   make is
      do
         io.put_string("Hello World in Eiffel.%N")
      end  -- make

end -- class HELLO_WORLD
```

Eye of the Beholder: “Amateurish”

BASIC

```
10 print"Hello World!"
20 goto 10
```

Perl

```
print "Hello, World!\n" while (1);
```

Social Science Literature

- Social Science Literature
  - Methodological / theoretical pieces, but no direct application to programming language design
    - Craft to Architectural tradition in shipbuilding – as a metaphor for IT and programming language design specifically (David McGee)
  - Comparing non-programmers to the IT industry (a la Turkle and Papert, 1990)
  - Control as the primary factor (a la Turkle)
    - Control of what, though?
      - The computer?
      - The functions?
      - The knowledge?
      - The economy?

Why?

- Technical considerations are critical
  - Platforms (Assembler, C)
  - Specific applications (algorithms, speed - Fortran)
- Languages do not “evolve”, they are constructed
  - Control (Ada, Algol, COBOL, C#, Assembler)
  - Power/Knowledge (Forth, Assembler)
  - Style (Smalltalk, Algol)
  - Philosophy (Haskell, Algol, Forth)
  - Market Forces (Java, C#, J)
  - Organizational Considerations (COBOL, Ada)

How?

- Not at all...
  - Too little Science & Technology focus
  - Languages are a black box
    - Not analyzed in parallel means
  - Tools, methods, theories exist
    - Craft Tradition
    - Knowledge / power relationships
    - Political Economics
- Future Directions
  - Analyzing practitioner texts
    - Comments in Code
    - USENET groups