Language Design
Introduction

• hacker (1985–2015)
Introduction

• hacker (1995–2015)

• currently at UvA
currently at UvA

photo credit: http://scii.nl/spaces.html
Amsterdam Subversive Center for Information Interchange
- currently
- currently at UvA
• hacker (1995-2015)
• currently at UvA
• wikipedian (2004-2015)
Introduction

  - currently at UvA
Intr

- currently at UvA
- wikipedian (2004–2012)
- animator (flash 2001–2005)
- graphic designer (2005–2008)
- font (2008), t-shirt
- mostly illustration

BibSLEIGH: http://bibtex.github.io
Introduction

• hacker (1995–2015)
  • currently at UvA
• wikipedian (2004–2015)
• animator (flash 2001, gif 2004–2007)
• graphic designer (2007–2015)
  • font (2008), t-shirts (2009–2015)
  • mostly illustrations
• indie game designer (1999–2015)
TALKING to computers
Computer?

Perhaps the professor could use your computer.

https://youtu.be/LkqiDu1BQXY?t=1m
David Monniaux, “Machine à calculer de Blaise Pascal sans sous ni deniers”,
These cards, however, have nothing to do with the regulation of the particular numerical data. They merely determine the operations* to be effected, which operations may of course be performed on an infinite variety of particular numerical values, and do not bring out any definite numerical results unless the numerical data of the problem have been impressed on the requisite portions of the train of mechanism.

In the above example, the first essential step towards an arithmetical result, would be the substitution of specific numbers for \(a\) and for the other primitive quantities which enter into the function.

Again, let us suppose that for \(F\) we put two complete equations of the fourth degree between \(x\) and \(y\). We must then express on the cards the law of elimination for such equations. The engine would follow out those laws, and would ultimately give the equation of one variable which results from such elimination. Various modes of elimination might be selected; and of course the cards must be made out accordingly. The following is one mode that might be adopted. The engine is able to multiply together any two functions of the form

\[a + bx + cx^2 + \ldots + px^n.\]

This granted, the two equations may be arranged according to the powers of \(y\), and the coefficients of the powers of \(y\) may be arranged according to powers of \(x\). The elimination of \(y\) will result from the successive multiplications and subtractions of several such functions. In this, and in all other instances, as was explained above, the particular numerical data and the numerical results are determined by means and by portions of the mechanism which act quite independently of those that regulate the operations.

In studying the action of the Analytical Engine, we find that the peculiar and independent nature of the considerations which in all mathematical analysis belong to operations, as distinguished from the objects operated upon and from the results of the operations performed upon those objects, is very strikingly defined and separated.

It is well to draw attention to this point, not only because its full appreciation is essential to the attainment of any very just and adequate general comprehension of the powers and mode of action of the Analytical Engine, but also because it is one which is perhaps too little kept in view in the study of mathematical science in general. It is, however, impossible to confound it with other considerations, either when we trace the manner in which that engine attains its results, or when we prepare the data for its attainment of those results. It was much to be desired, that when mathematical processes pass through the human brain instead of through the medium of inanimate mechanism, it were equally a necessity of things that the reasonings connected with operations should hold the same just place as a clear and well-defined branch of the subject of analysis, a fundamental but yet independent

* We do not mean to imply that the only use made of the Jacquard cards is that of regulating the algebraical operations, but we mean to explain that these cards and portions of mechanism which regulate these operations, are wholly independent of those which are used for other purposes. M. Neubrier explains that there are three classes of cards used in the engine for three distinct sets of objects, viz. Cards of the Operations, Cards of the Variables, and certain Cards of Numbers. (See pages 678 and 687.)
Adriaan J. van Wijngaarden (1916–1987)

photo credit: http://academictree.org/physics/peopleinfo.php?pid=25212
Adele Katz Goldstine (1920-1964) 1946

John von Neumann (1903-1957)
Flow Diagrams
(John von Neumann, Herman Goldstine, Adele Goldstine)
"the technique of program composition can be mechanised"

Haskell Brooks Curry (1900–1982)
Equations

00  i = 10
01  0:  y = (sqrt abs t) + 5 cube t
02  y ≤ 400  if ≤ to 1
04  i print, 'TOO LARGE' print-and-return
05  0 0  if = to 2
06  1:  i print, y print-and-return
07  2:  TO UO shift
08  i = i - 1
09  0  i  if ≤ to 0
10  stop

Coded representation

00  00  00  W0  03  Z2
10  02  07  Z5  11  TO
00  Y0  03  09  20  06
00  00  00  Y0  Z3  41
00  00  00  Z0  Z0  72
00  00  Y0  59  W0  58
00  00  00  TO  U0  99
00  W0  03  W0  01  Z1
00  00  00  Z0  W0  40
00  00  00  00  Z2  08
Grace Murray Hopper (1906-1992)
(1) READ-ITEM A(11) .
(2) VARY I 10(-1)0 SENTENCE 3 THRU 10 .
(3) J = I+1 .
(4) \( Y = SQR |A(J)| + 5*A(J)^3 \) .
(5) IF \( Y > 400 \), JUMP TO SENTENCE 8 .
(6) PRINT-OUT I, Y .
(7) JUMP TO SENTENCE 10 .
(8) Z = 999 .
(9) PRINT-OUT I, Z .
(10) IGNORE .
(11) STOP .

MATH-MATIC
(1) COMPARE PART-NUMBER (A) TO PART-NUMBER (B); IF GREATER GO TO OPERATION 13; IF EQUAL GO TO OPERATION 4; OTHERWISE GO TO OPERATION 2.

(2) READ-ITEM B; IF END OF DATA GO TO OPERATION 10.
A5000-WRONG-ANSWER SECTION.
    DISPLAY 'Question was incorrectly answered'
    DISPLAY PLAYERS(CURRENT-PLAYER)
    ' was sent to the penalty box'
    SET IN-PENALTY-BOX-YES(CURRENT-PLAYER) TO TRUE
    MOVE '1' TO DID-PLAYER-WIN
    ADD 1 TO CURRENT-PLAYER
    IF (CURRENT-PLAYER = PLAYER-COUNT) THEN
        MOVE 1 TO CURRENT-PLAYER
    END-IF
    .
Fortran

John Backus (1924-2007)
"a language so far ahead of its time that it was not only an improvement on its predecessors but also on nearly all its successors"

```plaintext
proc abs max = ([,]real a, ref real y, ref int i, k)real:
comment The absolute greatest element of the matrix a, of size ra by 2ra is transferred to y, and the subscripts of this element to i and k; comment
begin
    real y := 0; i := 1a; k := 2 1a;
    for p from 1a to ra do
        for q from 2 1a to 2 ra do
            if abs a[p, q] > y then
                y := abs a[p, q];
                i := p; k := q
            fi
        od
    od;
y
end # abs max #
```

Tony Hoare (b. 1934)
APL

\((\sim R \in R^\circ \cdot \times R) / R \leftarrow 1 \downarrow \uparrow R\)

\(\square \leftarrow \{\omega / \approx \approx \{\omega V \neq \\{\omega\} \omega \in' <>'\}t x t\)

\(life \leftarrow \{\uparrow 1 \omega V \cdot \Lambda 3 \ 4=+/-,-1 \ 0 \ 1^\circ \cdot \Theta -1 \ 0 \ 1^\circ \cdot \circ \subset \omega\}\)

Kenneth E. Iverson
(1920-2004)

APL keyboard by Rursus https://commons.wikimedia.org/wiki/File:APL-keybd2.svg CC-BY-SA
photo from http://archive.vector.org.uk/art10002990 (Michel Dumontier, Hommage à Ken Iverson)
Quote from http://www.azquotes.com/quote/1119653
Kenneth E. Iverson (1920-2004)

“"It is important to distinguish the difficulty of describing and learning a piece of notation from the difficulty of mastering its implications. [...] The very suggestiveness of a notation may make it seem harder to learn because of the many properties it suggests for exploration”

quote from http://www.azquotes.com/quote/1119653

photo from http://archive.vector.org.uk/art10002990 (Michel Dumontier, Hommage a Ken Iverson)
1967
Seymour Papert
(b. 1928)

1974
Radia Perlman
(b. 1951)
Scratch, http://scratch.mit.edu
UML

Margaret H. Hamilton (b. 1936)

http://threefingeredfox.net/?p=143, PD
SOFTWARE LANGUAGE ENGINEERING

CREATING DOMAIN-SPECIFIC LANGUAGES USING METAMODELS

ANNEKE KLEPPE

FOREWORD BY
JEAN-MARIE FAVRE
SOFTWARE LANGUAGE ARCHAEOLOGIST
AND SOFTWARE ANTHROPOLOGIST, LIG, AGONIT,
UNIVERSITY OF GRINOBLE, FRANCE

Everybody Needs Somebody To Love
Words & Music by Bert Russell, Jerry Wexler & Solomon Burke

\[ \text{\textcopyright 1968, 1983 by DMM Music, Inc.} \]

\[ \text{(See spoken intro.)} \]

\[ \text{Eb} \quad \text{Ab} \quad \text{Db} \quad \text{Ab} \quad \text{Eb} \quad \text{Ab} \quad \text{Db} \quad \text{Ab} \]

\[ \text{Eb} \quad \text{Ab} \quad \text{Db} \quad \text{Ab} \quad \text{Eb} \quad \text{Ab} \quad \text{Db} \quad \text{Ab} \]

Milestone summary

- Universal hardware + programs
- Automated code generation
- Programming with words
- Language documentation
- Domain-specific languages
- Engineer languages when needed
Part II

Domain

Ontology

Schema

Grammar
Domain

- What will the language be used for?
  - Algorithms?
  - Markup?
  - Data?
  - Constraints
  - Finance?
- Visual?
- Drawing?
- Parallel?
- Spreadsheet
- Formulae?
- Queries?
- Music?
- Dance?
- Space?
- Food?
Part II

Domain
  topic
  theme
  problems
  concerns

Ontology

Schema

Grammar
Ontology

- Fundamental entities of the domain
- Their properties
- Interrelationships
- (Could be a mindmap or mindmap-like)
Part II

Domain
- topic
- theme
- problems
- concerns

Ontology
- state
- things
- events
- concepts
- properties
- composition

Schema

Grammar
Schema

• What are “sentences”, conceptually?
• Lists? Sets? Trees? Graphs? Tables?
• Looking inside a sentence, what is there?
• Are there different kinds of sentences?
• (Explicit language modelling)
<table>
<thead>
<tr>
<th>Perspective</th>
<th>Data Composites or Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planner</td>
<td><strong>OPERATIONAL NODE</strong>&lt;br&gt;<strong>CONNECTIVITY DESCRIPTION OV-2</strong>&lt;br&gt;<strong>Level 1</strong>&lt;br&gt;<strong>OPERATIONAL INFORMATION EXCHANGE MATRIX OV-3</strong>&lt;br&gt;<strong>Information Elements at the leaf level:</strong>&lt;br&gt;• Level 3 of the OV-5 IOs&lt;br&gt;• Level 1 of the OV-2 nodes&lt;br&gt;<strong>OPERATIONAL ACTIVITY MODEL OV-5 Level 1</strong>&lt;br&gt;<strong>OPERATIONAL ACTIVITY MODEL OV-5 Level 2</strong>&lt;br&gt;<strong>OPERATIONAL ACTIVITY MODEL OV-5 Level 3</strong>&lt;br&gt;<strong>Other OV/SV products if applicable</strong></td>
</tr>
<tr>
<td>Owner</td>
<td><strong>OPERATIONAL NODE</strong>&lt;br&gt;<strong>CONNECTIVITY DESCRIPTION OV-2</strong>&lt;br&gt;<strong>Level 1</strong>&lt;br&gt;<strong>Level 2</strong>&lt;br&gt;<strong>OPERATIONAL INFORMATION EXCHANGE MATRIX OV-3</strong>&lt;br&gt;<strong>Information Elements at the leaf level:</strong>&lt;br&gt;• Level 5 of the OV-5 IOs&lt;br&gt;• Level 1 of the OV-2 nodes&lt;br&gt;<strong>OPERATIONAL ACTIVITY MODEL OV-5 Level 4</strong>&lt;br&gt;<strong>OPERATIONAL ACTIVITY MODEL OV-5 Level 5</strong>&lt;br&gt;<strong>Other OV/SV products if applicable</strong></td>
</tr>
<tr>
<td>Designer</td>
<td><strong>SYSTEMS INTERFACE</strong>&lt;br&gt;<strong>DESCRIPTION SV-1</strong>&lt;br&gt;<strong>Level 1</strong>&lt;br&gt;<strong>SYSTEMS DATA EXCHANGE MATRIX SV-4</strong>&lt;br&gt;<strong>Data Elements at the leaf level:</strong>&lt;br&gt;• Level 3 of the SV-4 data flows&lt;br&gt;• Level 1 of the SV-1 nodes/systems&lt;br&gt;<strong>SYSTEMS FUNCTIONALITY DESCRIPTION SV-4</strong>&lt;br&gt;<strong>Level 2</strong>&lt;br&gt;<strong>Level 3</strong>&lt;br&gt;<strong>Other OV/SV/TV products if applicable</strong></td>
</tr>
<tr>
<td>Builder</td>
<td><strong>SYSTEMS INTERFACE</strong>&lt;br&gt;<strong>DESCRIPTION SV-1</strong>&lt;br&gt;<strong>Level 1</strong>&lt;br&gt;<strong>Level 2</strong>&lt;br&gt;<strong>Level 3</strong>&lt;br&gt;<strong>SYSTEMS DATA EXCHANGE MATRIX SV-4</strong>&lt;br&gt;<strong>Data Elements at the leaf level:</strong>&lt;br&gt;• Level 6 of the SV-4 data flows&lt;br&gt;• Level 3 of the SV-1 nodes/systems&lt;br&gt;<strong>TECHNICAL STANDARDS PROFILE TV-1</strong>&lt;br&gt;<strong>Standards at the leaf level:</strong>&lt;br&gt;• Level 6 of the SV-4 functions/data&lt;br&gt;• Level 3 of the SV-1 systems&lt;br&gt;<strong>SYSTEMS FUNCTIONALITY DESCRIPTION SV-4</strong>&lt;br&gt;<strong>Level 4</strong>&lt;br&gt;<strong>Level 5</strong>&lt;br&gt;<strong>Level 6</strong>&lt;br&gt;<strong>Other OV/SV/TV products if applicable</strong></td>
</tr>
</tbody>
</table>

No more than 6 levels of decomposition for each type of product within a perspective.
All products within a perspective remain cohesive as to level of detail provided in each.

**DoDAF Perspectives and Decomposition Levels**

**DoDAF** = Department of Defence Architecture Framework
• Algebraic data type:

data Bool
    = tt()
    | ff()
    | conj(Boolean L, Boolean R)
    | disj(Boolean L, Boolean R)
;
Part II

**Domain**
- topic
- theme
- problems
- concerns

**Ontology**
- state
- things
- events
- concepts
- properties
- composition

**Schema**
- data types
- containment
- manipulation
- initialisation
- abstract structure

**Grammar**
Grammar

• How do you write sentences down?
• What alphabet do you use?
• How symbols are constructed in it?
• Text? Table? Diagrams? Unicode? Colours?
\[
\text{compilation} ::= \\
\quad \text{compilation_unit}^* \\
\]

\[
\text{compilation_unit} ::= \\
\quad \text{visibility_restriction?} \ "\text{\textit{separate}}"? \ \text{unit_body} \\
\]

\[
\text{visibility_restriction} ::= \\
\quad \"\text{\textit{restricted}}\" \ \text{visibility_list}? \\
\]

\[
\text{visibility_list} ::= \\
\quad \("(\" \langle\text{unit_name}\rangle:\text{name} \text{\"", \langle\text{unit_name}\rangle:\text{name}}\)\text{* }\)\"\) \\
\]

\[
\text{unit_body} ::= \\
\quad \text{subprogram_body} \\
\quad \text{module_specification} \\
\quad \text{module_body} \\
\]
Aivaloglou, Hoepelman, Hermans, A Grammar for Spreadsheet Formulas Evaluated on Two Large Datasets, SCAM 2015
wave to activate
push to move
turn to rotate
swipe to dismiss
touch to select
spread to scale
raise hand to shoot
Sources & recs


- Nathan Shedroff, Christopher Noessel, Make It So, 2012.

- Tema Ra, Aesthetics of Futuristic Interfaces, http://rhzm.ru/posts/114

Don’t Let Your Dreams Be Dreams!

**Domain**
- topic
- theme
- problems
- concerns

**Ontology**
- state
- things
- events
- concepts
- properties
- composition

**Schema**
- data types
- containment
- manipulation
- initialisation
- abstract structure

**Grammar**
- symbols
- alphabet
- sentences
Domain
- topic
- theme
- problems
- concerns

Ontology
- state
- things
- events
- concepts
- properties
- composition

Schema
- data types
- containment
- manipulation
- initialisation
- abstract structure

Grammar
- symbols
- alphabet
- sentences
PRESENT
THANK YOU

Follow @grammarware

KTHXBYE