Lego Programming

Simone Martini

Alma mater studiorum • Università di Bologna
and
INRIA FoCUS – Sophia / Bologna

and

Collegium - Institut d'études avancées de Lyon 2018-2019

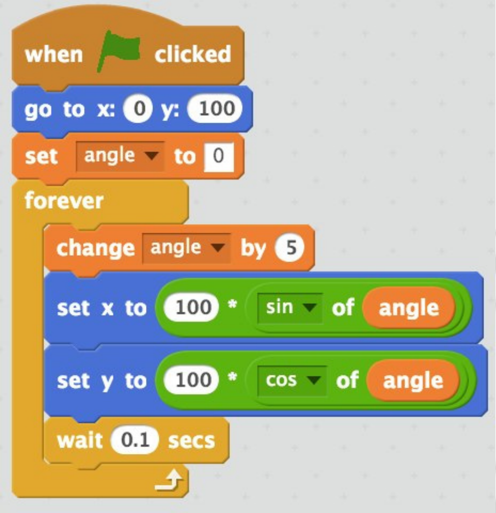
COLLEGIUM
DE LYON
Université de Lyon

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA

informatiques mathématiques

Very preliminary
material on
non-verbal programming *experiences*

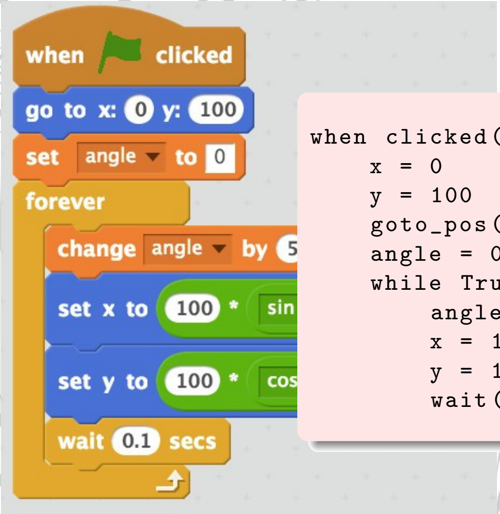
Scratch



```
when [flag] clicked
go to x: 0 y: 100
set angle to 0
forever
    change angle by 5
    set x to 100 * sin of angle
    set y to 100 * cos of angle
    wait 0.1 secs
```

A project of the Lifelong Kindergarten Group at the MIT Media Lab

Scratch

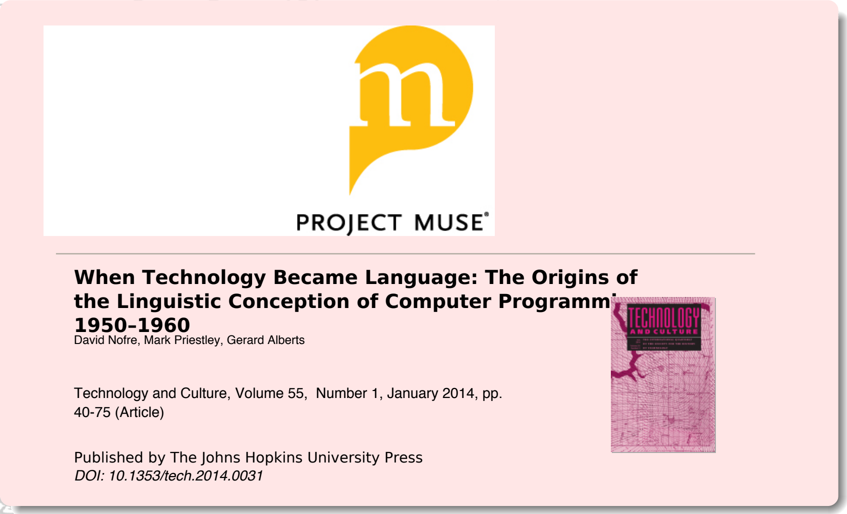


```
when clicked
go to x: 0 y: 100
set angle to 0
forever
    change angle by 5
    set x to 100 * sin
    set y to 100 * cos
    wait 0.1 secs
```

```
when clicked(flag):
    x = 0
    y = 100
    goto_pos(x,y)
    angle = 0
    while True:
        angle += 5
        x = 100*sin(angle)
        y = 100*cos(angle)
        wait(0.1)
```

A project of the Lifelong Kindergarten Group at the MIT Media Lab

The linguistic metaphor



PROJECT MUSE®

When Technology Became Language: The Origins of the Linguistic Conception of Computer Programmi 1950–1960

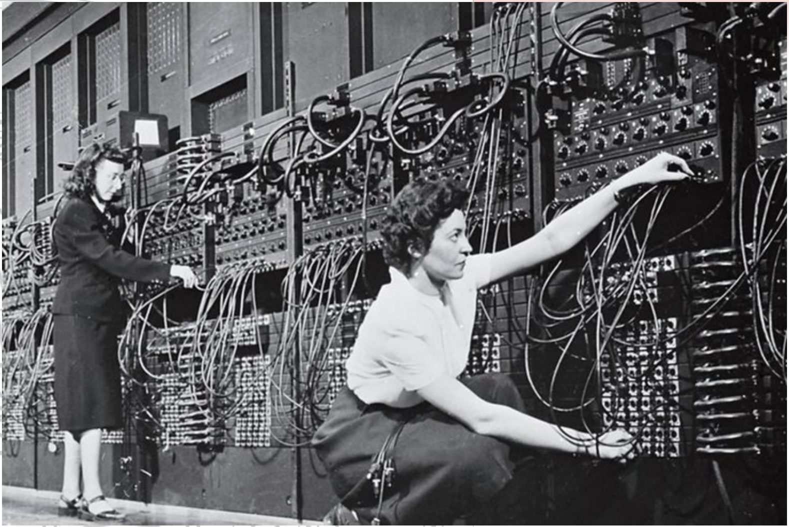
David Nofre, Mark Priestley, Gerard Alberts

Technology and Culture, Volume 55, Number 1, January 2014, pp. 40-75 (Article)

Published by The Johns Hopkins University Press
DOI: 10.1353/tech.2014.0031

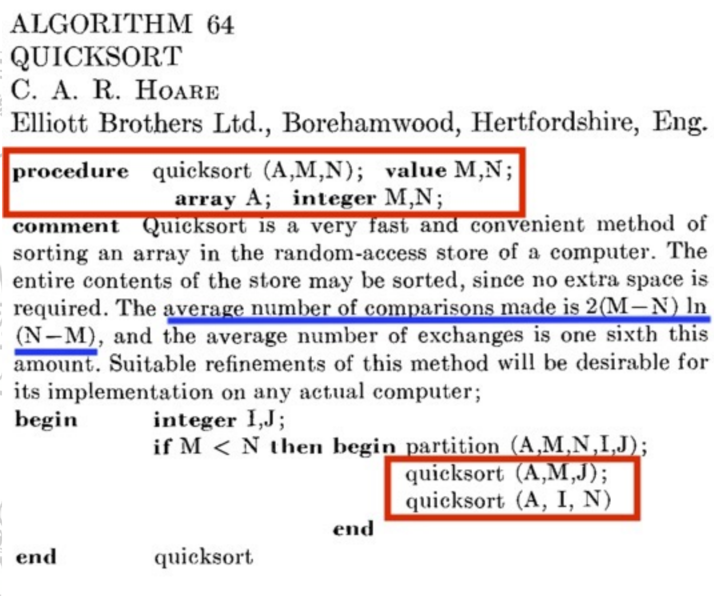
Programming languages

- tool
- object of study
- meta-language:
  algorithms published in Algol on the *Communications of ACM*
- programs are “immutable mobiles” (B. Latour; see also J. Goguen)



Programming ENIAC: 1945-46

CACM 1961



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
```

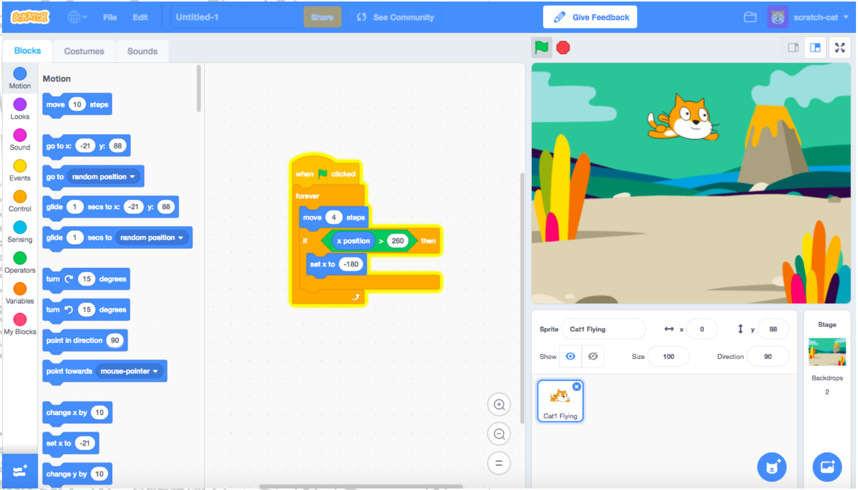
H. Bergson:
l'illusion rétrospective du vrai

Par le seul fait de s'accomplir, la réalité projette derrière elle son ombre dans le passé indéfiniment lointain ; elle paraît ainsi avoir préexisté, sous forme de possible, à sa propre réalisation

[H. Bergson, La pensée et le mouvant, 1934]

also: le mouvement rétrospectif/rétrograde du vrai

Scratch, in use



```
when 🏁 clicked
forever
    move 4 steps
    if x position > 260 then
        set x to -180
```

Bricks, not languages. . .

From Scratch's official distribution

- a *graphical* programming *language*
- children can *drag and combine* code blocks to make a range of programs
- it's a bit like the *programming equivalent of LEGO!*
- learn coding concepts [. . .] without needing to learn a text-based programming language
- they won't be slowed down by their keyboard skills or the *ability to remember complex code*

From freeCodeCamp

On Scratch:

- Isn't even a proper language
- It's more reminiscent of LEGO than real engineering
- Every student over the age of 12 agrees with my diagnosis

- It turns out, typing is overrated
- Programming isn't like English
  There aren't a million different words and sentence structures
- Creating blocks saves time
- [Scratch] enables to focus 100% on the design and logic of the programs, not the semantics.

From freeCodeCamp

On Scratch:

- Isn’t even a proper language
- It’s more reminiscent of LEGO than real engineering
- Every student over the age of 12 agrees with my diagnosis

- It turns out, typing is overrated
- Programming isn’t like English
  There aren’t a million different words and sentence structures
- Creating blocks saves time
- [Scratch] enables to focus 100% on the design and logic of the programs, not the semantics.

©Steve Krouse

Lego programming

Non linguistics components

Brick game

Continuous interaction and feedback

Visual aspect more important than the linguistic one

A long history

Programming is interacting with the *executor*, the machine

- Logo, 1969:

  W. Feurzeig and S. Papert. Programming languages as a conceptual framework for teaching mathematics. Final report on the first fifteen months of the Logo Project. TR 1889. BBN, Cambridge, MA.

- Smalltalk, 1972

  Alan Key, XEROX PARC

Interaction is mediated by powerful metaphors:

turtle

object

A long history

Programming is interacting with the *executor*, the machine

- Logo, 1969:

  W. Feurzeig and S. Papert. Programming languages as a conceptual framework for teaching mathematics. Final report on the first fifteen months of the Logo Project. TR 1889. BBN, Cambridge, MA.

- Smalltalk, 1972

  Alan Key, XEROX PARC

Interaction is mediated by powerful metaphors:

turtle

object

Per conoscere il mondo bisogna costruirlo

To know the world we must construct it

Cesare Pavese, Il mestiere di vivere (Engl.: This business of living). 1952

(we must re-construct it in a story — or as a story)

In other words, we make not just to have, but to know.
But the having can happen without most of the knowing taking place.

Alan Kay, The early history of Smalltalk. 1993

The problem:

How, then, can we construct for knowing and not just for having?

Per conoscere il mondo bisogna costruirlo

To know the world we must construct it

Cesare Pavese, Il mestiere di vivere (Engl.: This business of living). 1952

(we must re-construct it in a story — or as a story)

In other words, we make not just to have, but to know.
But the having can happen without most of the knowing taking place.

Alan Kay, The early history of Smalltalk. 1993

The problem:

How, then, can we construct for knowing and not just for having?

Per conoscere il mondo bisogna costruirlo

To know the world we must construct it

Cesare Pavese, Il mestiere di vivere (Engl.: This business of living). 1952

(we must re-construct it in a story — or as a story)

In other words, we make not just to have, but to know.
But the having can happen without most of the knowing taking place.

Alan Kay, The early history of Smalltalk. 1993

The problem:

How, then, can we construct for knowing and not just for having?

Constructing for knowing:

Seymour Papert:

we must construct meaningful objects and relations

Papert

Central for Papert:

not computer science, or a programming language, or programming, *per se*

but construction, with computational means,
of *concrete* versions of *abstract* mathematical concepts

We understand what we construct

Constructivism:

Jean Piaget

Computational “environments” are one of the most effective and economic ways to obtain such models in an autonomous manner.

Papert

Central for Papert:

not computer science, or a programming language, or programming, *per se*

but construction, with computational means,
of *concrete* versions of *abstract* mathematical concepts

We understand what we construct

Constructivism:

Jean Piaget

Computational “environments” are one of the most effective and economic ways to obtain such models in an autonomous manner.

Computational thinking

Seymour Papert, 1980

Mindstorm: Children, Computers, And Powerful Ideas

Their visions of how to integrate computational thinking into everyday life was insufficiently developed.

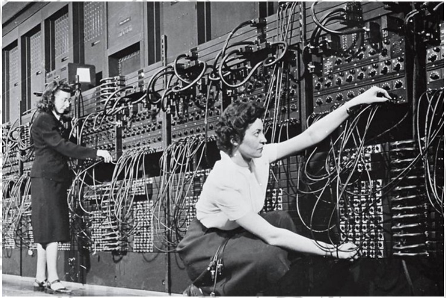
Their = people using computers for offering computationally rich activities

Papert's constructionism

Constructivism + Meaningfulness

We build concrete versions of abstract concepts
and we enter into a relationship with these concrete objects

Too tempting



Papert

The use of some programming languages is one of the most effective and economic ways for children to obtain such models in an autonomous manner.

But:

The modality of interaction with the computational media is as (and probably *more*) important than its contents.

The try and correct cycle:
Feedback from the computational objects more than static semantics.

The context for the “computational thinking” citation

Samba schools for computation

In the next few years we shall see the formation of some computational environments that deserve to be called “samba schools for computation.”

There have already been attempts in this direction [but] their visions of how to integrate computational thinking into everyday life was insufficiently developed.

Samba schools, in Rio

Samba schools, in Rio

- clubs ranging from hundreds to thousands of people, from children to their grandparents, from novices to professionals
- they gather every weekend to dance and to meet with friends
- all of them dance: the novice learns, the expert teaches and practices for harder moves
- a great social cohesion, a great sense of belonging, a strong idea of having a "common purpose."
- learning is spontaneous and natural, it is also *deliberate*

In samba schools for computation:

- no knowledge is transmitted
- pupils will learn because are immersed in an environment
- activities are both "rich of computational principles" and meaningful for the community

Affective relation

Building "objects to think with"

oxymoron:
the abstract is obtained using the concrete

In the choice of such objects:

there is not only a cognitive aspect

there is always a fundamental affective component

Papert: "I was in love with gears!"

Early Smalltalk

Alan Kay:

- 1966-1969: in graduate school at University of Utah
- Summer 1967: learns Papert's ideas from Minsky
- Winter 1968: meets Papert and his group

> *This encounter finally hit me with what the destiny of personal computing really was going to be: [. . .] a personal dynamic medium [which] had to extend into the world of childhood.*

> *All came together to form an image of what a personal computer really should be. It had to be no larger than a notebook, and needed an interface as friendly as JOSS', GRAIL's, and LOGO's, but with the reach of Simula and FLEX.*

Early Smalltalk

Alan Kay:

- 1966-1969: in graduate school at University of Utah
- Summer 1967: learns Papert's ideas from Minsky
- Winter 1968: meets Papert and his group

This encounter finally hit me with what the destiny of personal computing really was going to be: [. . .] a personal dynamic medium [which] had to extend into the world of childhood.

All came together to form an image of what a personal computer really should be. It had to be no larger than a notebook, and needed an interface as friendly as JOSS', GRAIL's, and LOGO's, but with the reach of Simula and FLEX.

Early Smalltalk

Alan Kay:

- 1966-1969: in graduate school at University of Utah
- Summer 1967: learns Papert's ideas from Minsky
- Winter 1968: meets Papert and his group

This encounter finally hit me with what the destiny of personal computing really was going to be: [. . .] a personal dynamic medium [which] had to extend into the world of childhood.

All came together to form an image of what a personal computer really should be. It had to be no larger than a notebook, and needed an interface as friendly as JOSS', GRAIL's, and LOGO's, but with the reach of Simula and FLEX.

Smalltalk

It isn't enough to just learn to read and write. There is also a literature that renders ideas. Language is used to read and write about them, but at some point the organization of ideas starts to dominate mere language abilities.

And it helps greatly to have some powerful ideas under one's belt to better acquire more powerful ideas [Papert 70s]. So, we decided we should teach design.

Smalltalk

> *Adele [Goldberg] decided that what was needed was an intermediary between the vague ideas about the problem and the very detailed writing and debugging that had to be done to get it to run in Smalltalk. She called the intermediary forms design templates.*
>
> *Using these the children could look at a situation [. . .] and decompose it into classes and messages without having to worry just how a method would work.*
>
> *We wanted more, and started to push on the inheritance idea as a way to let novices build on frameworks that could only be designed by experts.*

Smalltalk

> *Adele [Goldberg] decided that what was needed was an intermediary between the vague ideas about the problem and the very detailed writing and debugging that had to be done to get it to run in Smalltalk. She called the intermediary forms design templates.*
>
> *Using these the children could look at a situation [. . .] and decompose it into classes and messages without having to worry just how a method would work.*
>
> *We wanted more, and started to push on the inheritance idea as a way to let novices build on frameworks that could only be designed by experts.*

Smalltalk

Adele [Goldberg] decided that what was needed was an intermediary between the vague ideas about the problem and the very detailed writing and debugging that had to be done to get it to run in Smalltalk. She called the intermediary forms design templates.

Using these the children could look at a situation [. . .] and decompose it into classes and messages without having to worry just how a method would work.

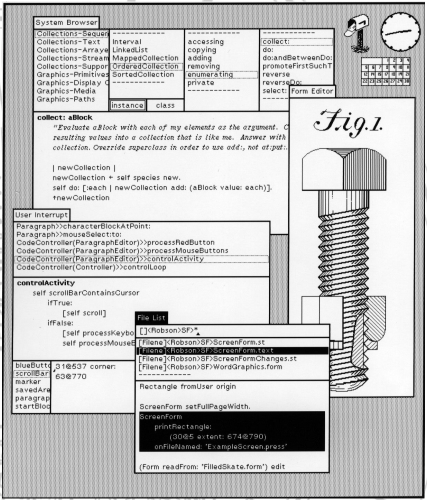
We wanted more, and started to push on the inheritance idea as a way to let novices build on frameworks that could only be designed by experts.

Smalltalk

From the objects and classes of Simula

to an ecosystem of interacting objects

Smalltalk for Alto



```
Collections-Sequen
Collections-Text
Collections-Arraye
Collections-Stream
Collections-Suppor
Graphics-Primitives
Graphics-Display C
Graphics-Media
Graphics-Paths
```

```
------------
Interval
LinkedList
MappedCollection
OrderedCollection
SortedCollection
------------
```

```
------------
accessing
copying
adding
removing
enumerating
private
------------
```

```
------------
collect:
do:
do:andBetweenDo:
promoteFirstSuchT
reverse
reverseDo:
select:
------
```

```
collect: aBlock
    "Evaluate aBlock with each of my elements as the argument.  C
    resulting values into a collection that is like me.  Answer with
    collection. Override superclass in order to use add:, not at:put:.

    | newCollection |
    newCollection ← self species new.
    self do: [:each | newCollection add: (aBlock value: each)].
    ↑newCollection
```

```
Paragraph>>characterBlockAtPoint:
Paragraph>>mouseSelect:to:
CodeController(ParagraphEditor)>>processRedButton
CodeController(ParagraphEditor)>>processMouseButtons
CodeController(ParagraphEditor)>>controlActivity
CodeController(Controller)>>controlLoop
```

```
controlActivity
    self scrollBarContainsCursor
        ifTrue:
            [self scroll]
        ifFalse:
            [self processKeybo
            self processMouseB
```

```
blueButto
scrollBar
marker
savedAre
paragrap
startBloc
```

```
31@537 corner:
63@770
```

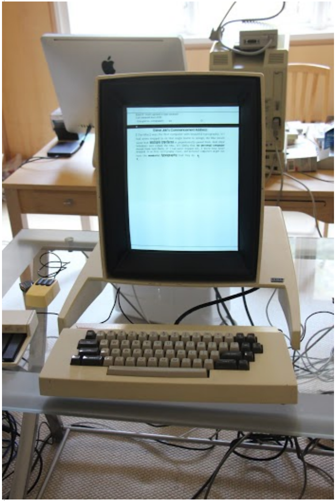
```
[]<Robson>SF>*
```

```
[Filene]<Robson>SF>ScreenForm.st
[Filene]<Robson>SF>ScreenForm.text
[Filene]<Robson>SF>ScreenFormChanges.st
[Filene]<Robson>SF>WordGraphics.form
------------
```

```
Rectangle fromUser origin

ScreenForm setFullPageWidth.
ScreenForm
    printRectangle:
        (30@5 extent: 674@790)
    onFileNamed: 'ExampleScreen.press'

(Form readFrom: 'FilledSkate.form') edit
```



Smalltalk

From the objects and classes of Simula

to an ecosystem of interacting objects

```
Smalltalk is NOT only its syntax or the class
library, it is not even about classes.  I'm sorry
that I long ago coined the term "objects" for this
topic because it gets many people to focus on the
lesser idea.

The big idea is "messaging" [...]  The Japanese have
a small word -- ma -- for "that which is in between".
```

A. Kay, message to the Squeak-dev mailing list. Sat Oct 10 1998

Smalltalk

From the objects and classes of Simula

to an ecosystem of interacting objects

> Smalltalk is NOT only its syntax or the class library, it is not even about classes. I’m sorry that I long ago coined the term "objects" for this topic because it gets many people to focus on the lesser idea.
>
> The big idea is "messaging" [...] The Japanese have a small word -- ma -- for "that which is in between".

A. Kay, message to the Squeak-dev mailing list. Sat Oct 10 1998

Smalltalk

And this is reflected into Smalltalk itself:

```
when ST hit the larger world, it was pretty much
taken as "something just to be learned", as though it
were Pascal or Algol.
```

while it is something one should fiddle about, tinker with:

```
at PARC we changed Smalltalk constantly, treating it
always as a work in progress
```

Smalltalk

And this is reflected into Smalltalk itself:

> when ST hit the larger world, it was pretty much taken as "something just to be learned", as though it were Pascal or Algol.

while it is something one should fiddle about, tinker with:

> at PARC we changed Smalltalk constantly, treating it always as a work in progress

Smalltalk

And this is reflected into Smalltalk itself:

when ST hit the larger world, it was pretty much taken as "something just to be learned", as though it were Pascal or Algol.

while it is something one should fiddle about, tinker with:

at PARC we changed Smalltalk constantly, treating it always as a work in progress

Concluding

Am I saying that...

- Scratch is not a programming *language*?
- Programming in Scratch is not a linguistic activity?

But programming in these visual languages is experienced, and often explicitly proposed, primarily as non-linguistic

This view has ancient and well established roots in some standard, linguistic ancestor

Concluding

Am I saying that. . .

- Scratch is not a programming *language*?
- Programming in Scratch is not a linguistic activity?

But programming in these visual languages is experienced, and often explicitly proposed, primarily as non-linguistic

This view has ancient and well established roots in some standard, linguistic ancestor

After all. . .

We interact with our smartphones
that is: we program them
exploiting less and less the linguistic metaphor

We should try to make sense of this

After all. . .

We interact with our smartphones
that is: we program them
exploiting less and less the linguistic metaphor

We should try to make sense of this