Revisiting the Dynabook concept *for education*

Hilaire Fernandes

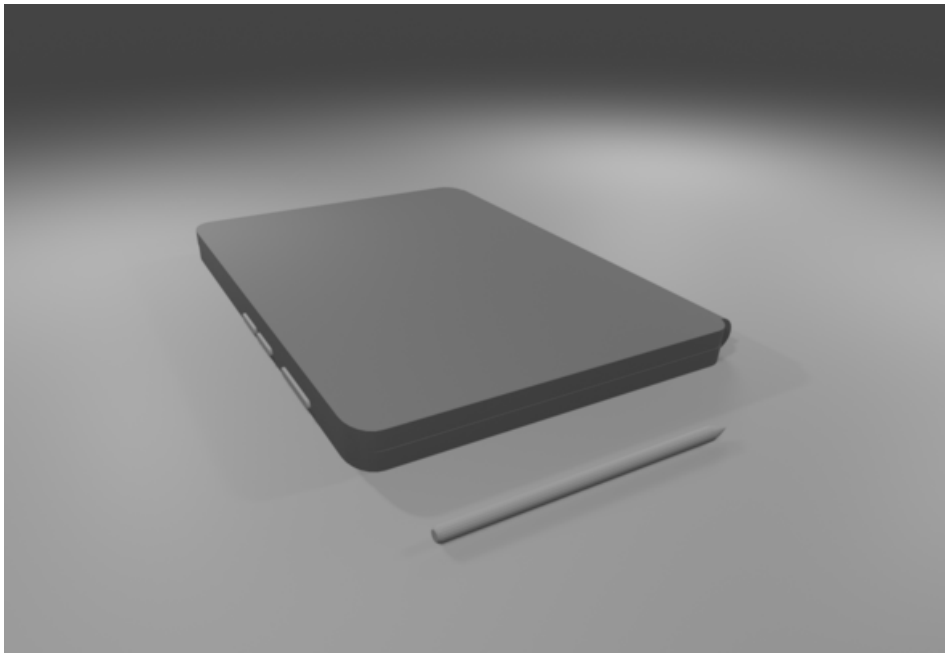
Department of Public Instruction
Geneva

November 2023

About me

- Educator in public school, Geneva, B.Math, Ma.Ed
- Computer scientist, Ma.CS, PhD.CS
- Free software enthusiast and user since 1998
- And of course, Smalltalk user since 2002

Contents



- The Dynabook in education is still mostly a concept
- In school, computerized environment used some time
  $\Rightarrow$ compare to the other sectors of the society

$\Rightarrow$ Any serious Dynabook realization should be considered as a *cash register of education*

Not this cash register

But more like this one

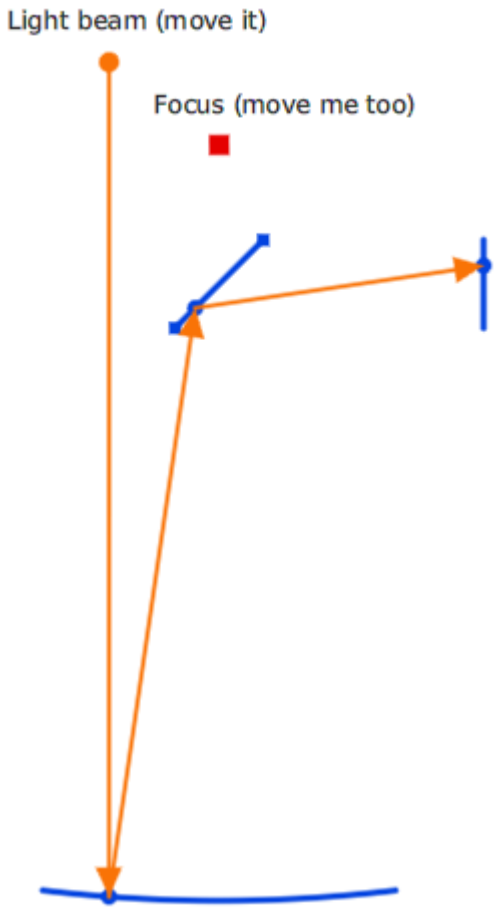
⇒ What about a dedicated hardware and software environment for a meaningful use in education

- Educator, professor
- Benefactor
- Student at university
- Econimist
- Software developer
- Project manager
- Hardware specialist
- Designer
- System administrator
- ...

A vehicle for dynamic models of knowledge
the user can design and/or operate on

Dynamic model of the *Newton Telescop*

The learner can operate a different level of knowledge light beam, focus point (change the mirror curve)



Teacher

What is a teacher?

- Yes, an educator. Manipulating, designing, sharing knowledge.
  $\Rightarrow$ Library of dynamic knowledge models, scriptable with a DSL and/or GUI (think **Dr. Geo**, **keynote Tuesday 9/11 at 11:00**)
- ... but also a manager
  - managing student
  - managing assignment
  - managing grade
  - managing meeting
  - managing parents
  - ...

Any realistic Dynabook revisit should take theses aspects in consideration.

Kid

Can we reduce the bag weight?

I have seen lightweight students assigned with mountain of materials:

- >10 binders
- >10 books
- >10 activity files
- numerous notebooks

Any realistic Dynabook revisit should take theses aspects in consideration.

Software environment

What do we need?

- Free software from the basement (OS) to the attic (end user applications)
- Rapid prototyping
- A malleable environment to develop knowledge models with state of the art visual representation
- Easy to implement DSL to script knowledge models
- Portable to different hardware architecture

$\Rightarrow$ Cuis-Smalltalk[1] to develop end user applications and knowledge models

[1] assumed biais

Free Hardware

Design there & there

Manufacture anywhere

Economic

Large scale adoption in one place, also require local economic benefits on that place:

- Software support
- Assembling/Manufacturing
- Repairing
- Training

$\Rightarrow$ Free software & hardware as prerequisites

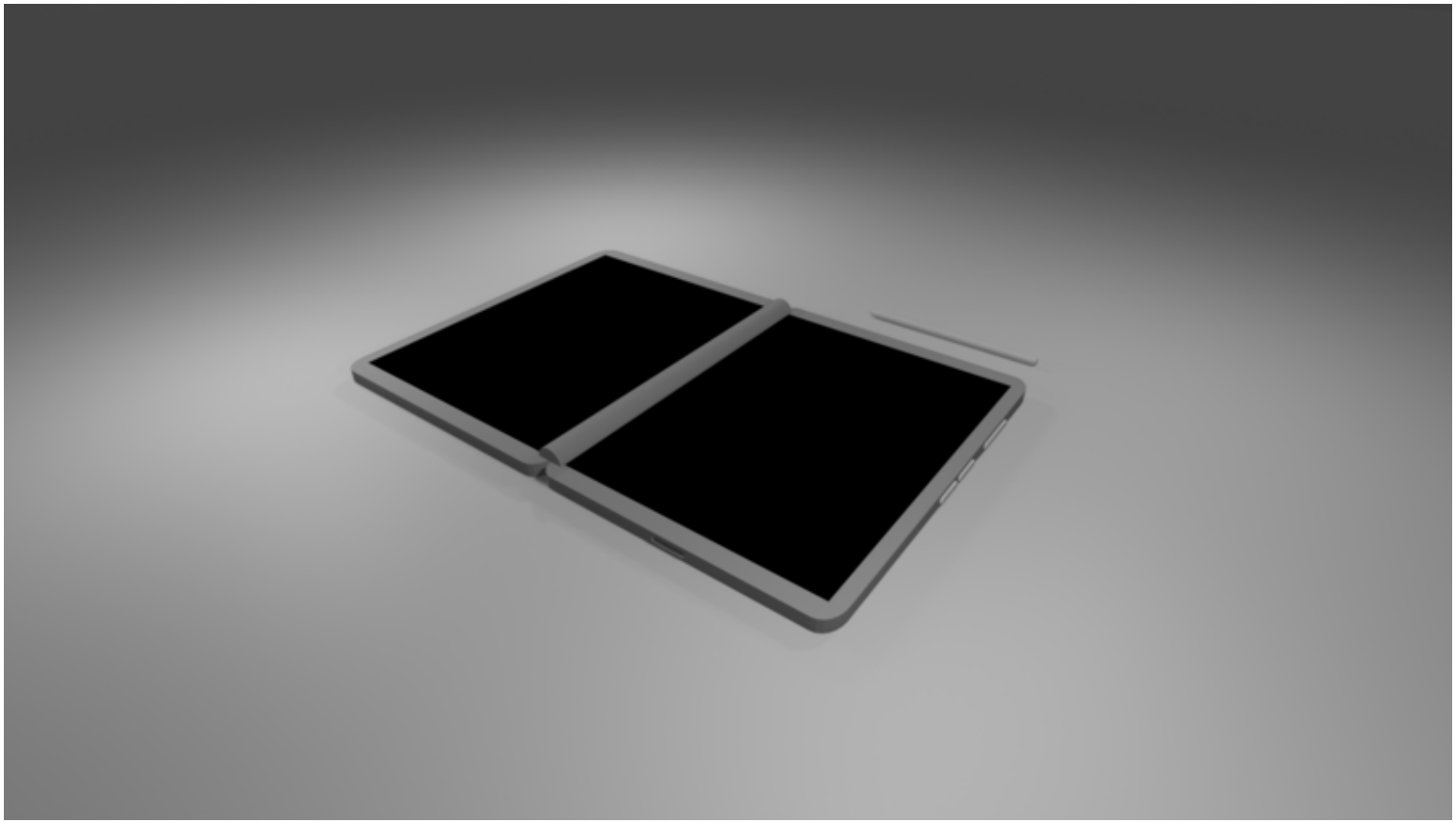
Is there any plan?

Roughly

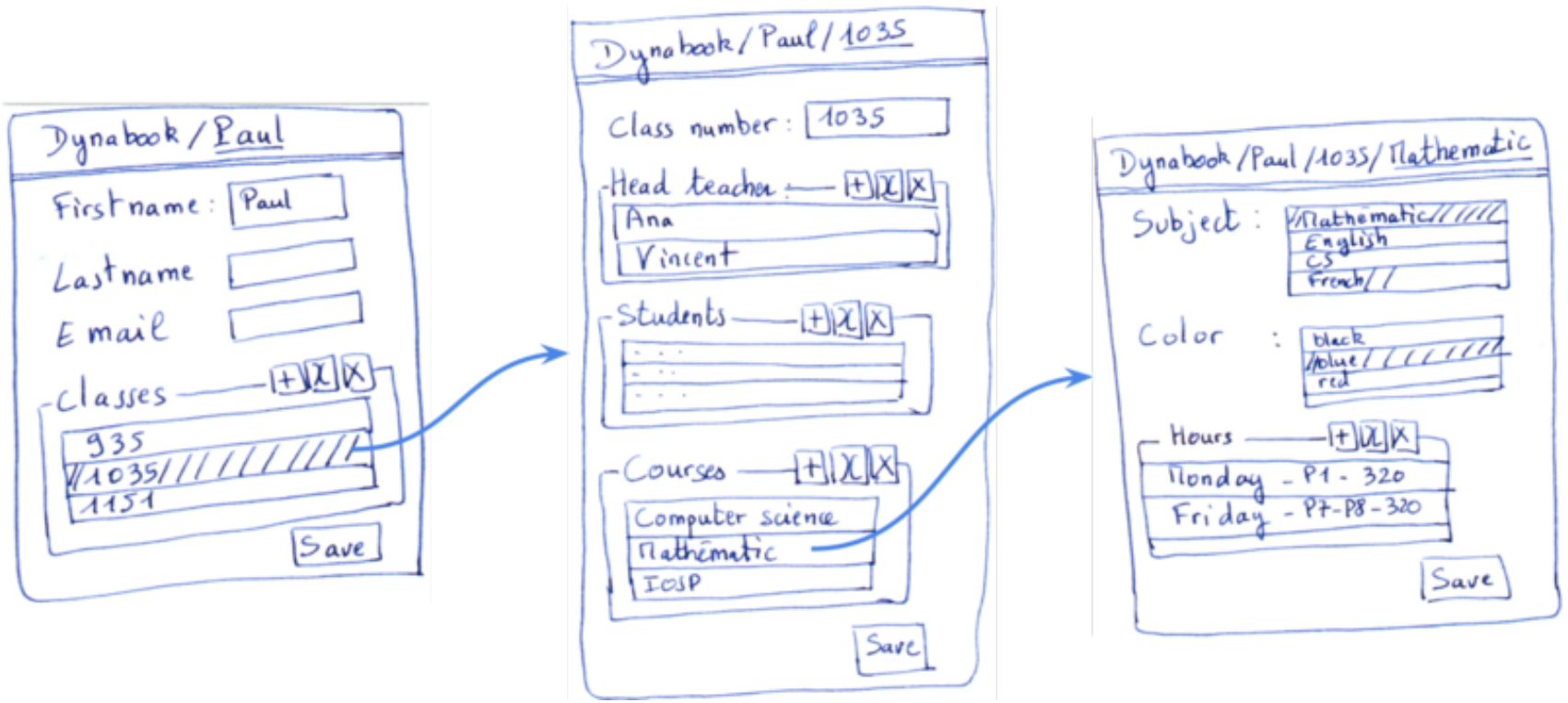
Iterations

1. **Develop the Dynabook app** (me, but join!)
2. Test Dynabook app in school and iterate with the development (1 or 2 teachers)
3. Develop hardware prototype with existing hardware
4. Develop Dynabook operating system
5. Test Dynabook app in school and iterate with the development (tenth of teachers)
6. Test Dynabook in school and iterate on the hardware and software (1 or 2 teachers/students)
7. Test Dynabook hardware and software with one classroom (30 users, students and teachers)

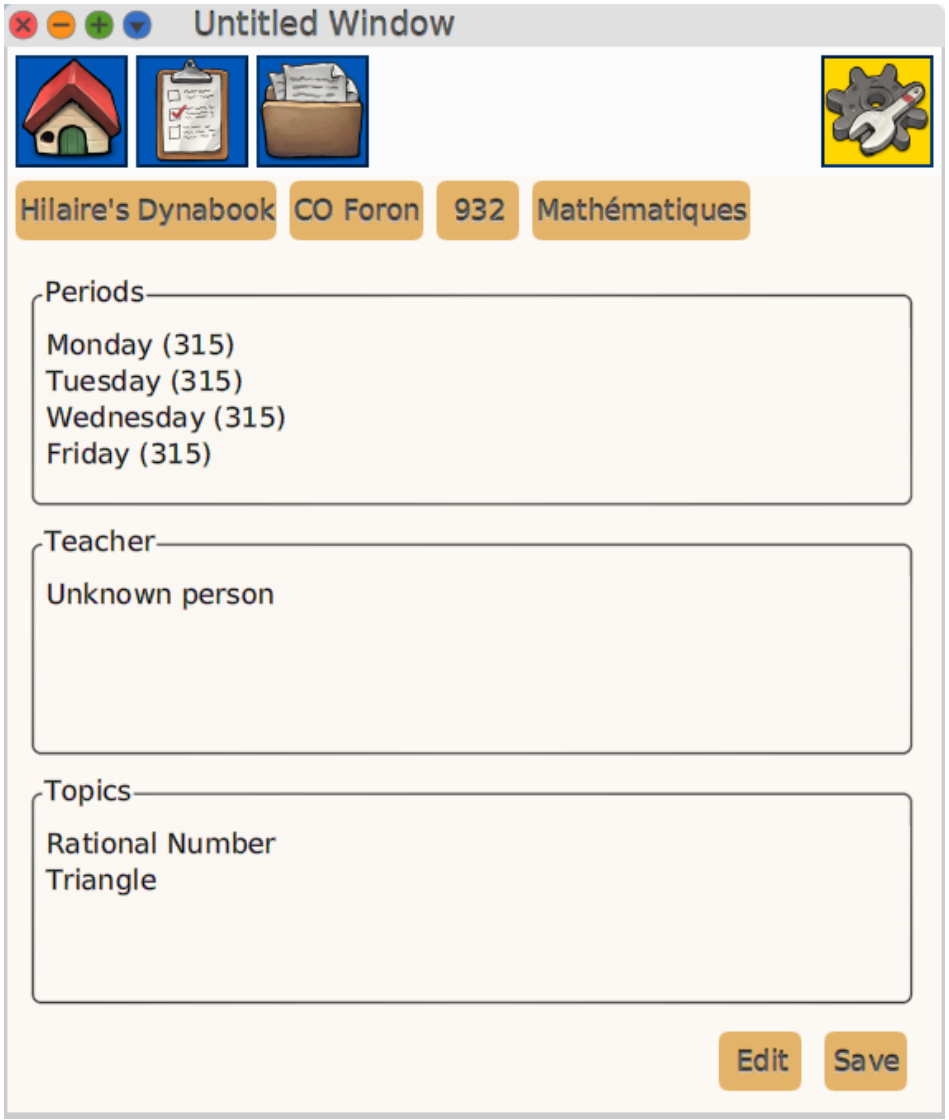
Visual Concepts



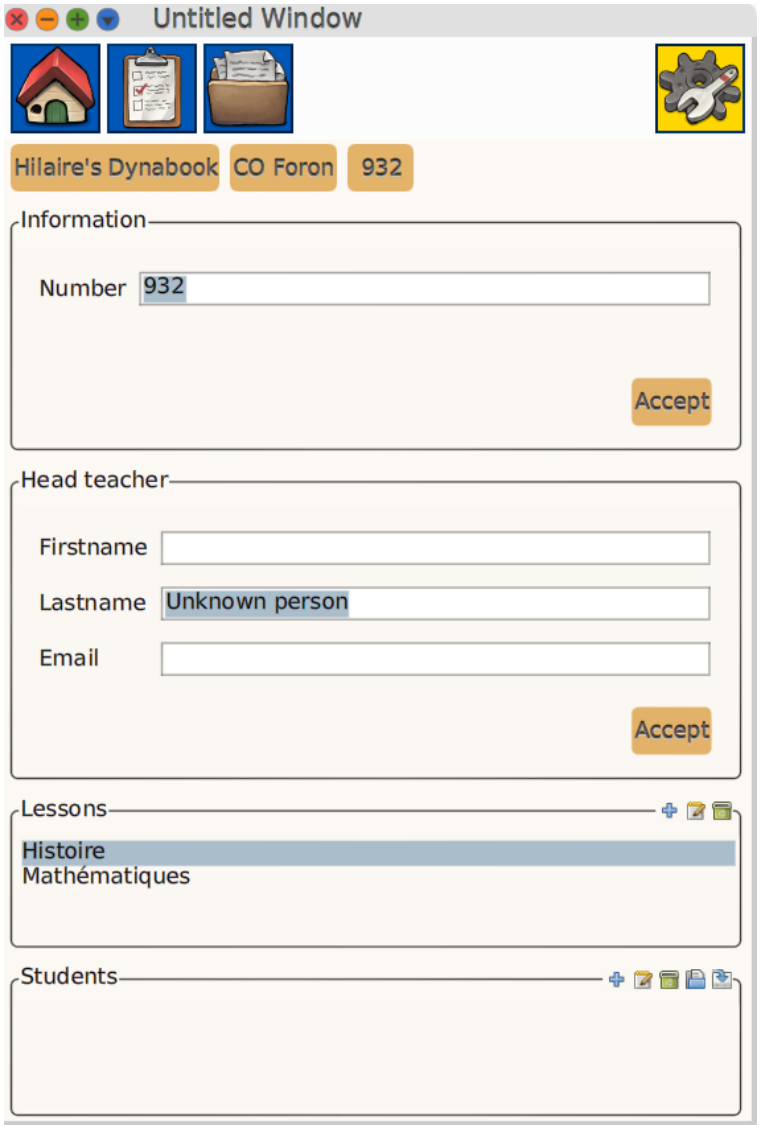
Management - Concept



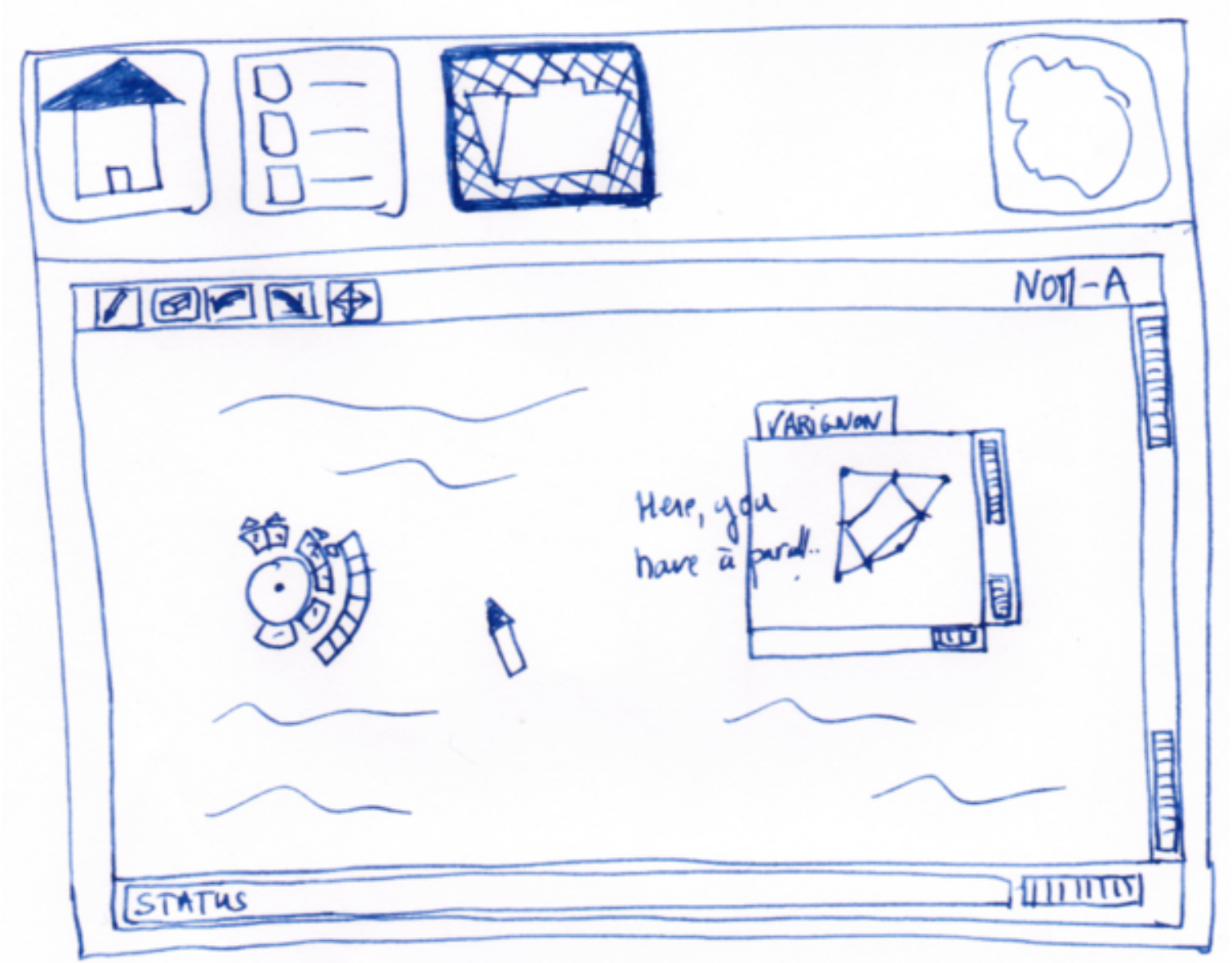
Management Viewer



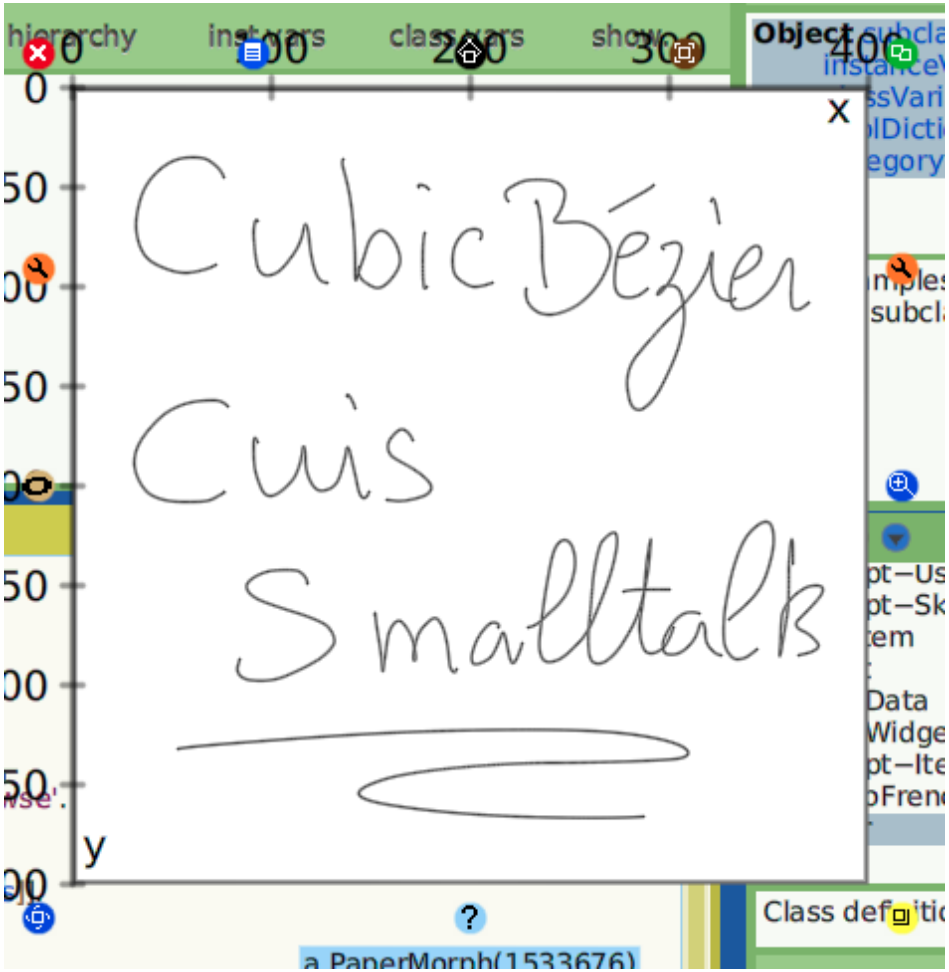
Management Editor



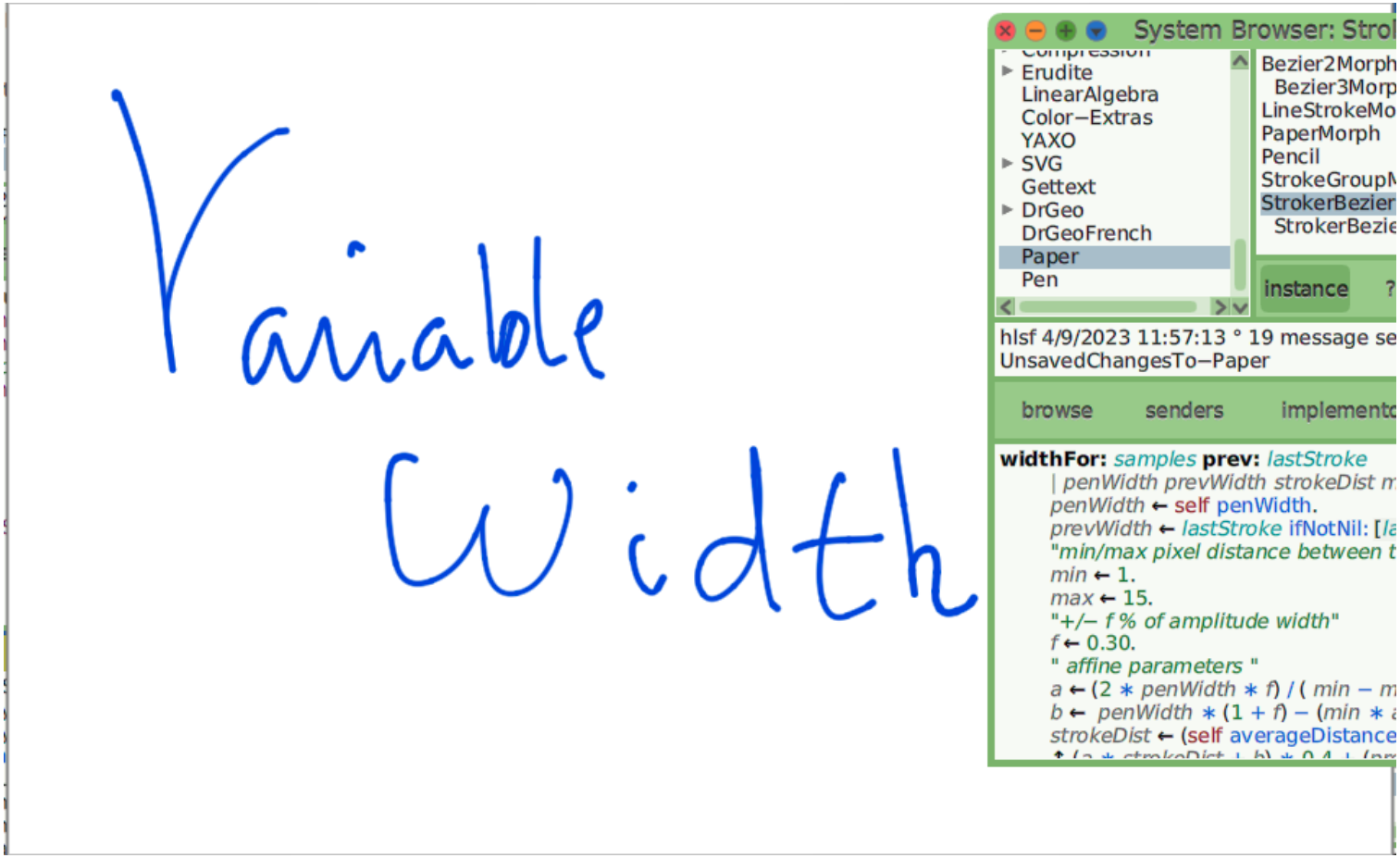
Knowledge environment



Paper Morph


hierarchy
instvars
classvars
show
Object
0
100
200
300
400
x
y
CubicBézier
Cuis
Smalltalk
Class def
a PaperMorph(1533676)

Paper Morph – Pressure emulation



1. Who & What?

- Educators
  ⇒ Write the specifications of interactive knowledge models in your domain
  ⇒ Review, compile existing pedagogical resources – under free license

- Software developers
  ⇒ Participate to the Dynabook.app design
  ⇒ Code with Cuis-Smalltalk interactive knowledge model and DSL

- Professors
  ⇒ Student projects

2. Who & What?

- Economists
  ⇒ Prospective on economic benefits
  ⇒ Environmental impacts

- Hardware specialists
  ⇒ Participate to the Dynabook hardware specification and design

- Benefactors
  ⇒ Set up a foundation to support the software and hardware specifications, design and development

If everything you try works, you aren't trying hard enough.

– Gordon Moore

`http://github.com/hilaire/dynabook`