Fundamentals of Situated Interaction

Wendy Mackay & Michel Beaudouin-Lafon

13 October 2017

Seminar format

- Discovering the principles of situated interaction:
  - Instrumental Interaction
    - Reification
    - Polymorphism
    - Reuse
  - Substrates

  - Human-computer partnerships
    - (Reciprocal co-adaptation)

Exercise #7 Finding Structures

Group exercise

1. Pick three activities.
2. Identify at least five structured ways to organize information for that activity.
3. Identify the type of computer structure and what it is for.

Example:	*Activity: Cooking*	
Shopping list	*paired list*	*specify amounts of ingredients*

Homework: send typed exercises to:

To: *mackay@lri.fr; mbl@lri.fr*

Subject: *Group # FSI: Ex #7: Finding Structures*

Exercise #8 Discovering properties

Individual exercise

Discovering physical affordances

- Choose a physical object (not a pencil !)
- List "hidden" properties that make it a tool
- How did you think of them? discover them?

Discovering digital affordances

- Choose a simple digital application
- List the hidden capabilities that you know of
- How did you think of them? discover them?

Homework: send typed exercises to:

To: *mackay@lri.fr; mbl@lri.fr*
Subject: *Group # FSI: Ex #7: Finding Structures*

Exercise #9 Expressivity

Choose a quotation (2-3 sentences)

Each person should:

1. Copy it with your best hand-writing
2. Write it while walking across the room
3. Type it on your phone – send to designated group member
4. Type it while walking – send to designated group member

Shuffle the physical writing samples

1. Can you tell which were written by the same person? How or why not?
2. Can you tell which were written in each condition? How or why not?

Look at the typed writing samples (text only)

3. Can you tell which text was written by whom? How or why not?
4. Can you tell which text was written in which condition? How or why not?

To: *mackay@lri.fr; mbl@lri.fr*

Subject: *Group # FSI: Ex #7: Finding Structures*

Readings

Readings

Beaudouin-Lafon, M. (2000). Instrumental Interaction: an Interaction Model for Designing Post-WIMP User Interfaces. *Proc. ACM Human Factors in Computing Systems*, CHI 2000, The Hague (The Netherlands), CHI Letters 2(1):446-453, ACM Press.

Beaudouin-Lafon, M. & Mackay, W. (2000). Reification, Polymorphism and Reuse: Three Principles for Designing Visual Interfaces. *Proc. Advanced Visual Interfaces*, AVI 2000, Palermo (Italie), ACM Press, pp 102-109.

Mackay, W. (2000) Responding to cognitive overload: Co-adaptation between users and technology. *Intellectica*. Vol. 30 (1), pp. 177-193.

Prepare Presentations

Presentations

1. Prepare a 15-minute talk
   - choose three papers
   - describe each technique (show video, if possible)
   - compare them and analyze:
     - how do they represent instruments? or not?
     - how are they co-adaptive? or not?

2. Prepare a 10-minute class discussion
   Suggest ideas for revising the techniques
   or creating a new co-adaptive instrument

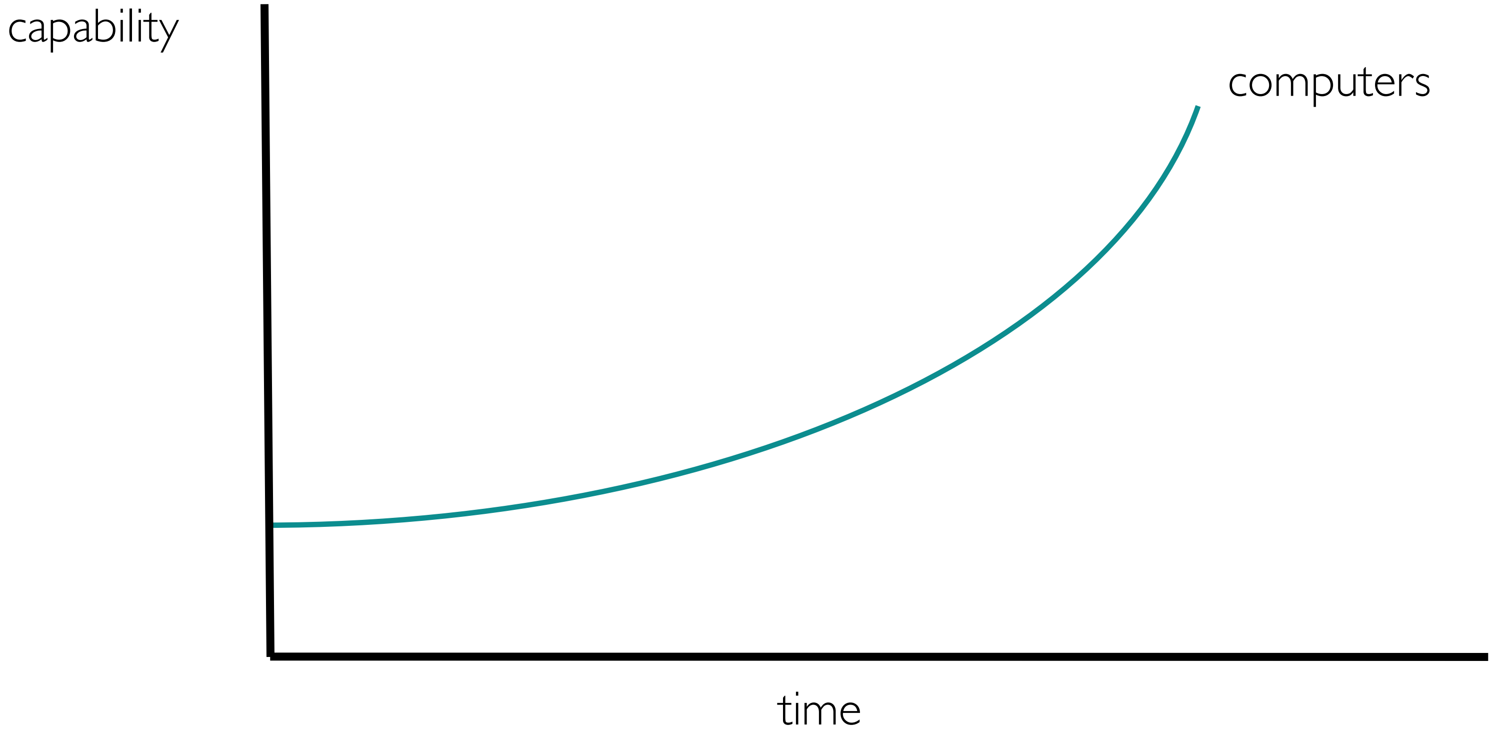
Human-Computer Partnerships

or

Co-Adaptive Instruments

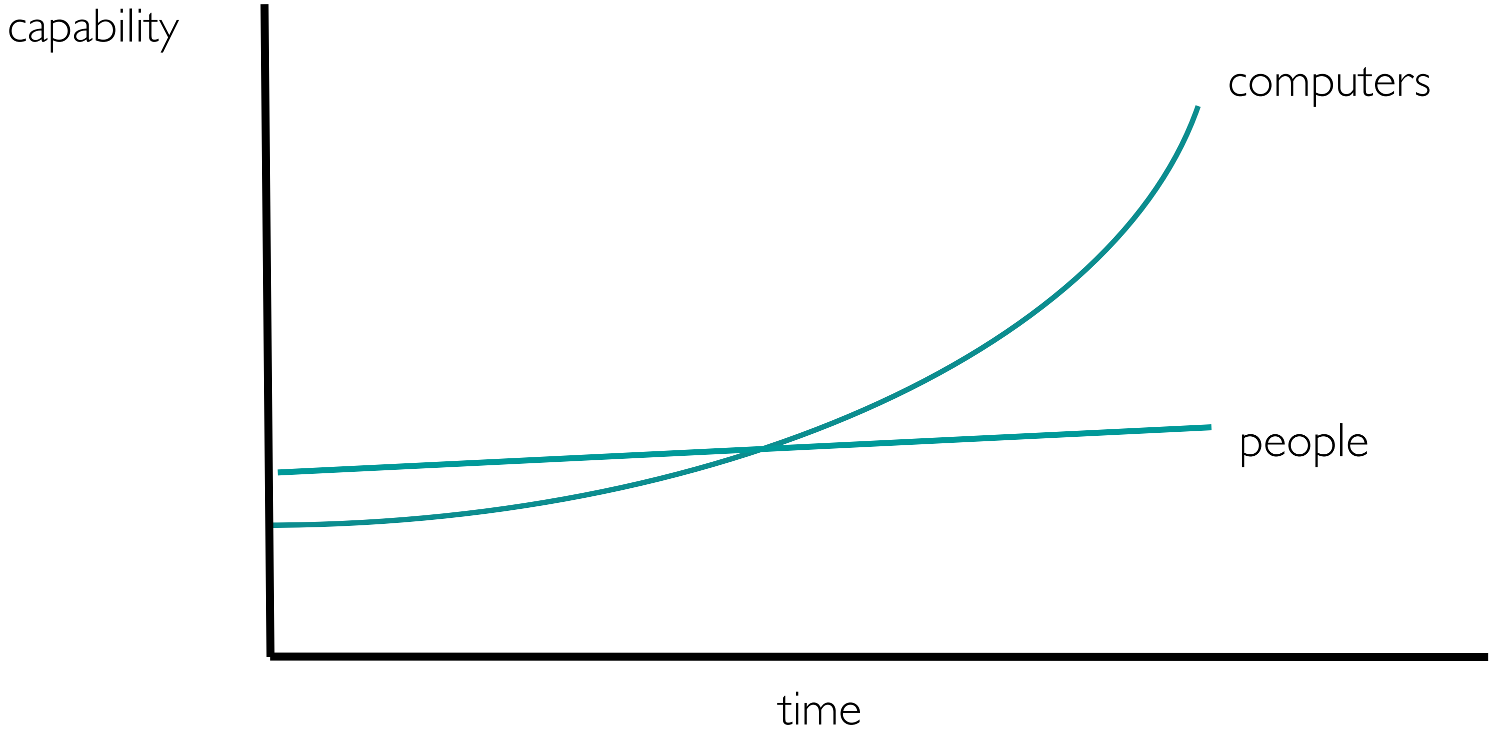
Computer hardware
has changed dramatically over the past
40 years …

Computer capabilities are exploding

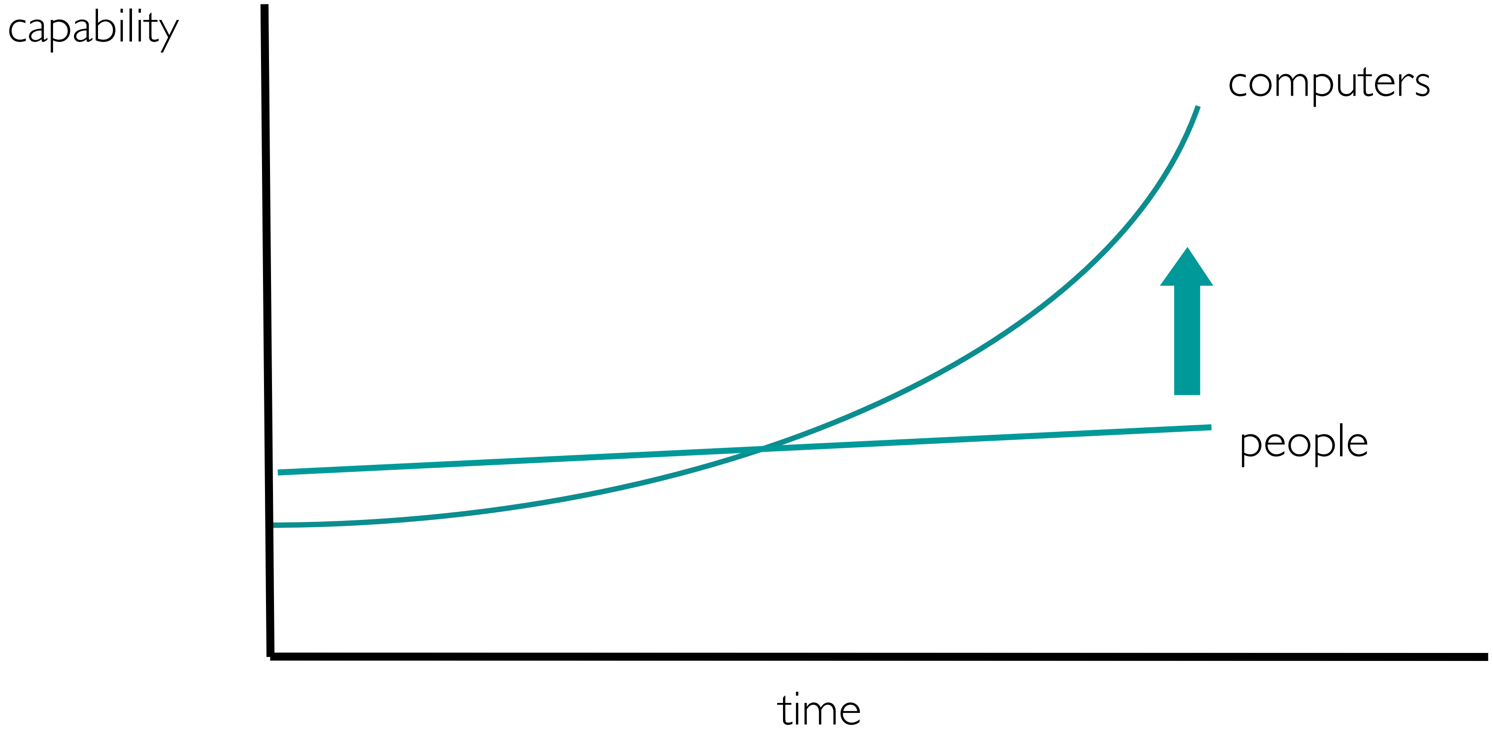


but human capabilities are not…

Bill Buxton:
'Moore's law vs. God's law'



Can computers augment human capabilities?



Key Challenge

How can we improve interactive systems, given today's ever-increasingly complex computational environment?

We have multiple relationships with computers

Computer as a ***tool***

I accomplish the task myself

Computer as a ***servant***

It accomplishes the task for me

Computer as a ***medium***

It lets me communicate
with other people

Graphical User Interfaces

Designed for executive secretaries to process documents
in a completely different technology environment

Dates back to the 1970s to:
copy hand-written notes
check for mistakes
format on letterhead

Problem:
Brilliant then,
out-moded today

GUIs are a vindication … and a challenge

Human-Computer Interaction research
    fought hard to make interfaces easier to use

Today, novices easily accomplish simple tasks

GUIs are a vindication … and a challenge

Human-Computer Interaction research
fought hard to make interfaces easier to use

Today, novices easily accomplish simple tasks

Yet …
advanced research in interaction techniques
is rarely adopted in commercial systems

Today, experts use ineffecient techniques and are
constantly forced to change their behavior

Desktops, the web and apps …

Require constant relearning:

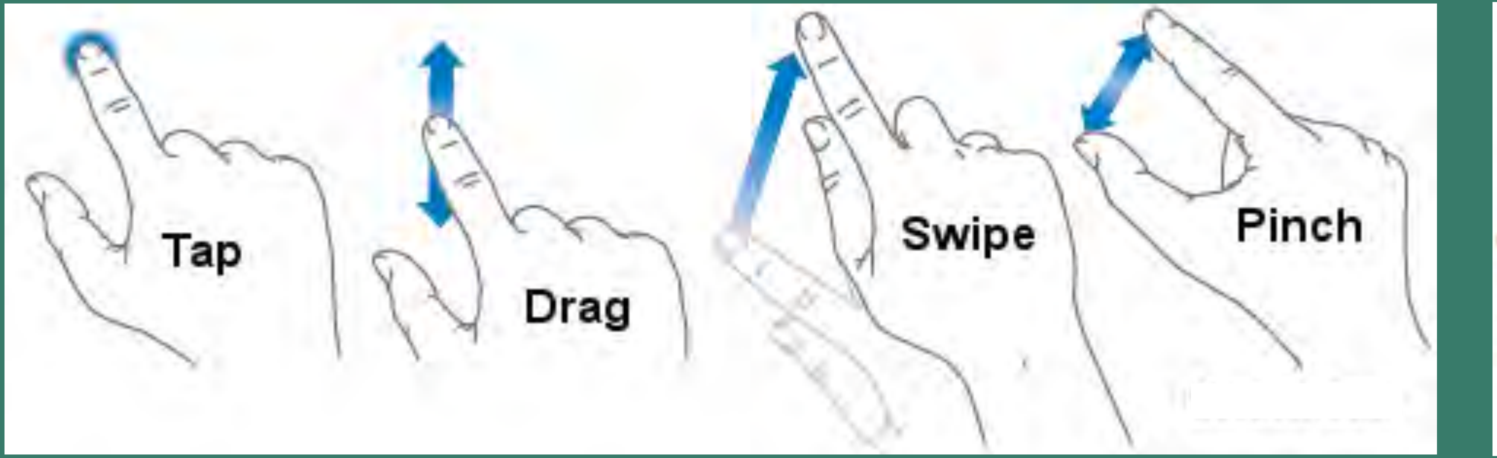
- each new version introduces arbitrary changes
- each system requires slightly different interaction

Require high visual attention

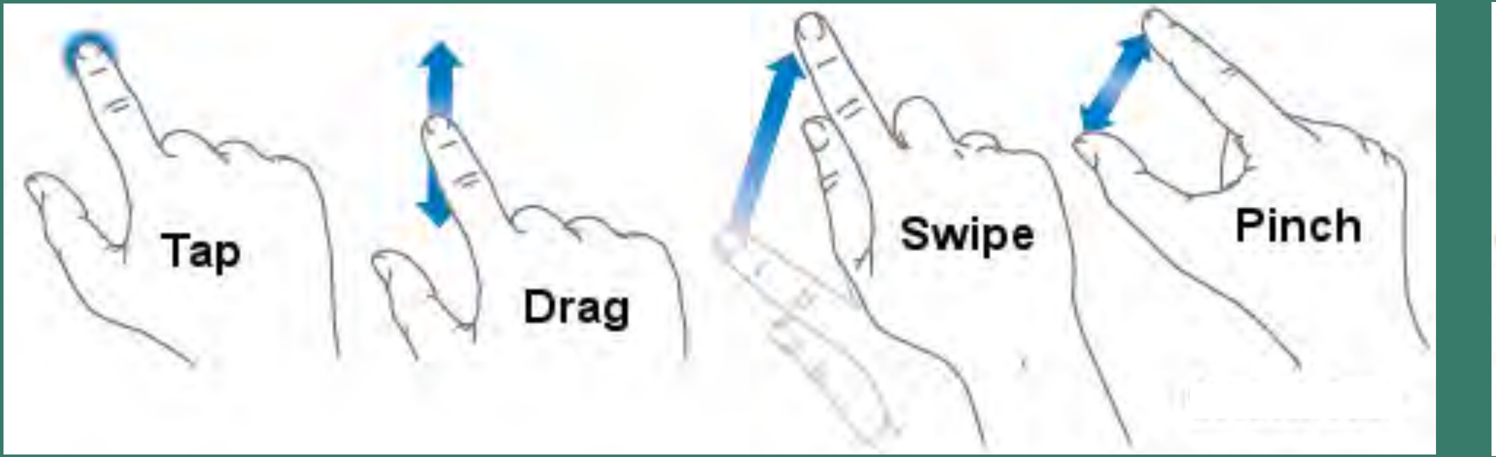
Do not scale

Depend on specific devices

Smartphones are easy … but not powerful



Smartphones are easy … but not powerful



What about creativity and expression?

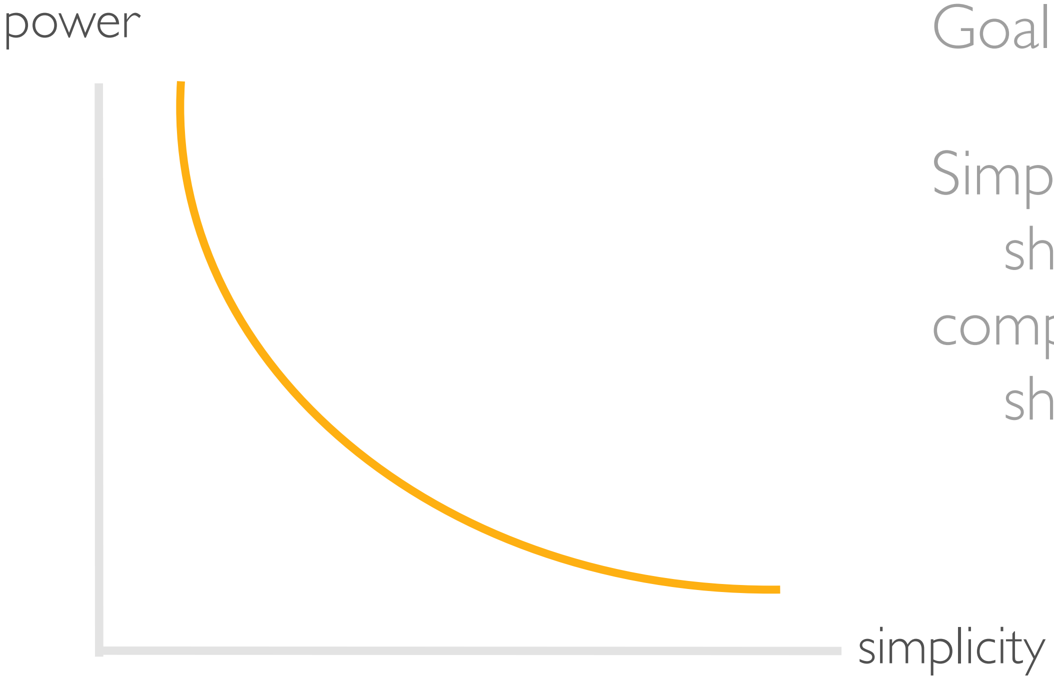
We need to reassess human-computer interaction

Early assumptions about graphical user interfaces
no longer hold

Everyone, not just experts
manages increasing quantities of data
faces information overload
constantly relearns the details of interaction

Redefine what we mean by "computer literacy"

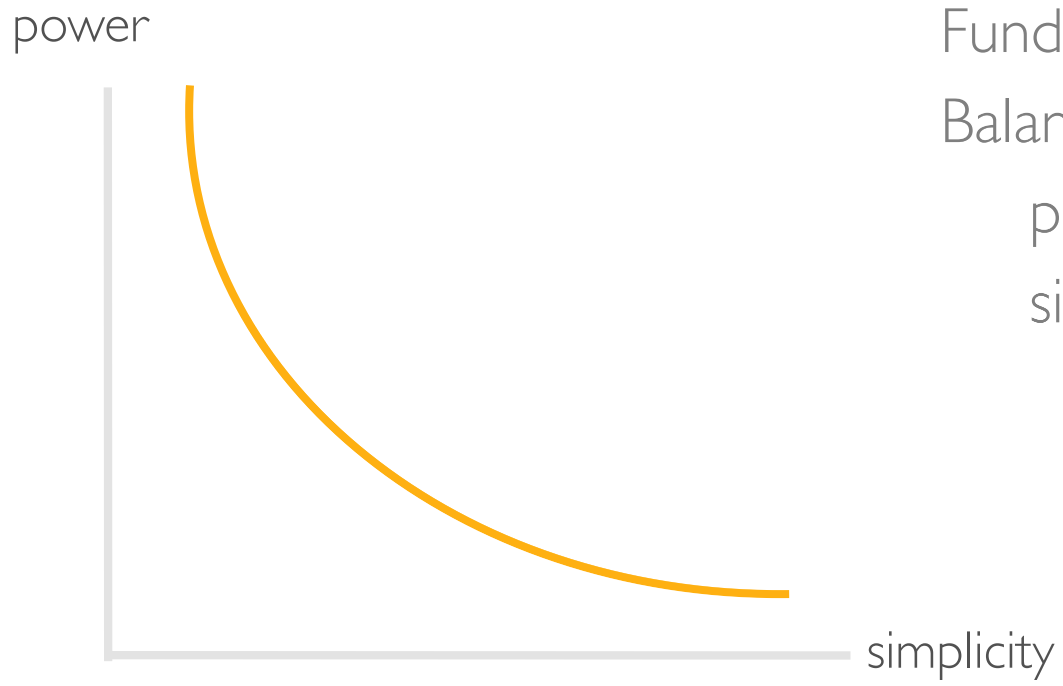
Design Trade-offs



Goal:

Simple things
should be simple …
complex things
should be possible

Design Trade-offs

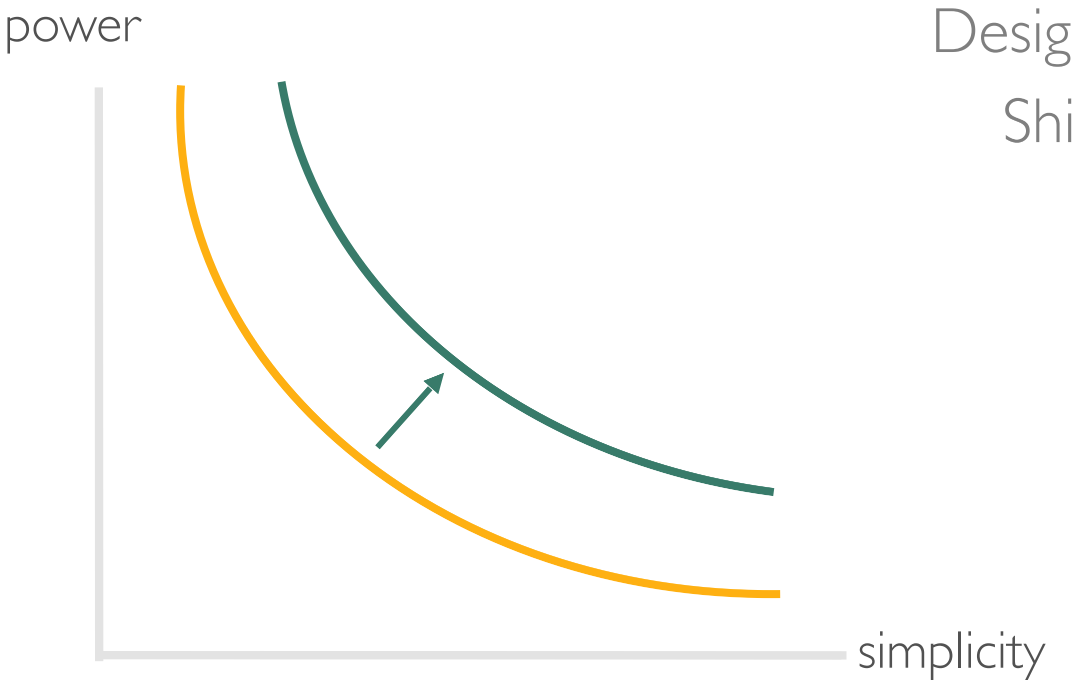


Fundamental challenge:

Balance trade-off between:

- power of expression
- simplicity of execution

Design Trade-offs



Design challenge:

Shift the curve

Strategy: Combine two key concepts

Instrumental interaction
(Michel Beaudouin-Lafon)

and

Co-adaptive phenomena
(Wendy Mackay)

Human-Computer Relationships

Between people and physical tools:

- follow well-known physical principles
- users can learn them
- users can appropriate them

Human-Computer Relationships

Between people and physical tools:

- follow well-known physical principles
- users can learn them
- users can appropriate them

Between people and computer tools:

- follow arbitrary constantly changing rules
- users must learn, and relearn, and relearn them
- users break them when they try to appropriate them

Focus on interaction, not interfaces

How can we let users control interaction
in a flexible, reusable way,
developing expertise without constantly relearning skills?

Solution: Co-adaptive Instruments
Separate interaction from data and functionality
Interaction becomes a first-class object

Co-adaptive phenomena

Inspired by co-evolution in biology
Organisms create their environment
even as they adapt to it

Anaerobic bacteria change the atmosphere
making it possible for aerobic bacteria to emerge

Users change spreadsheets from an addition tool
to a tool for exploring 'what if' scenarios

Key phenomenon: *Co-adaptation*

Users *adapt* to a new system

- they learn to use it

Users *adapt* the new system to their own needs

- they appropriate and change it

Co-adaptive instruments

Creative activities require both

- especially when integrating physical and digital information

Create digital tools that are as intuitive, and learnable,

- as physical tools

Co-adaptive Instruments

Worthwhile spending time and energy learning them

Complex tools become accessible
- can learn cognitive and sensori-motor skills
- can adapt to new situations

Move beyond
- graphical user interfaces
- to expert instruments

To do this:
- Extract widgets from applications
- to create personal instruments

Reciprocal Co-adaptation

People adapt their behavior to technology

… they learn it

People adapt the technology for their own purposes

… they appropriate it

Computers adapt their behavior to people

… machine learning

Computers adapt human behavior

… training

Human-Computer Partnerships

People have rich cognitive and
sensory motor capabilities

increasingly,
so do computers

Why is the interface so limited?

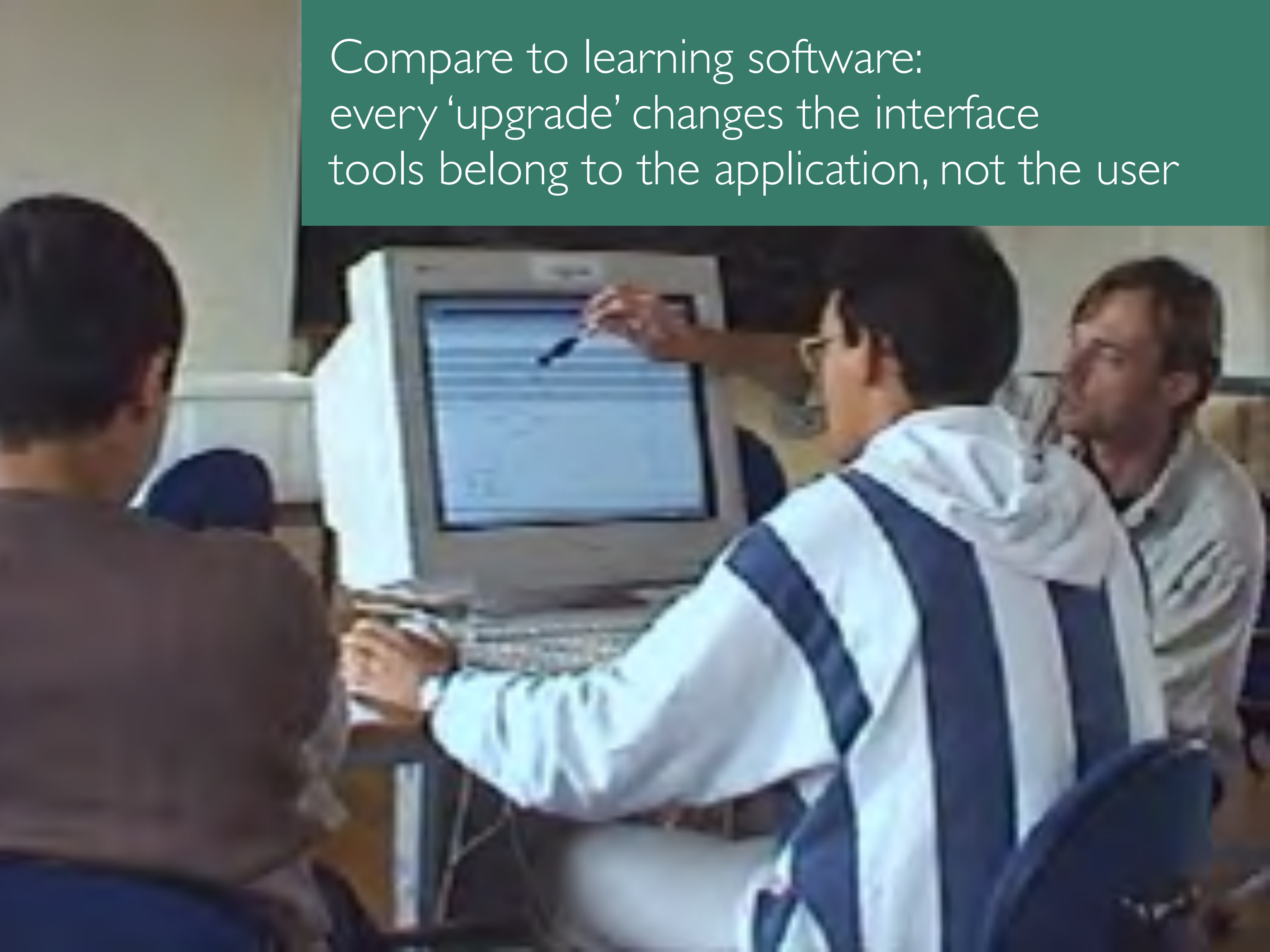
Physical tools

- follow the laws of physics
- we learn them
- we appropriate them

Computer tools

- follow the whims of programmers
- we learn, and relearn and relearn
- and then we break them!

Learning to play a musical instrument
—from novice to virtuoso—
the instrument becomes part of the body



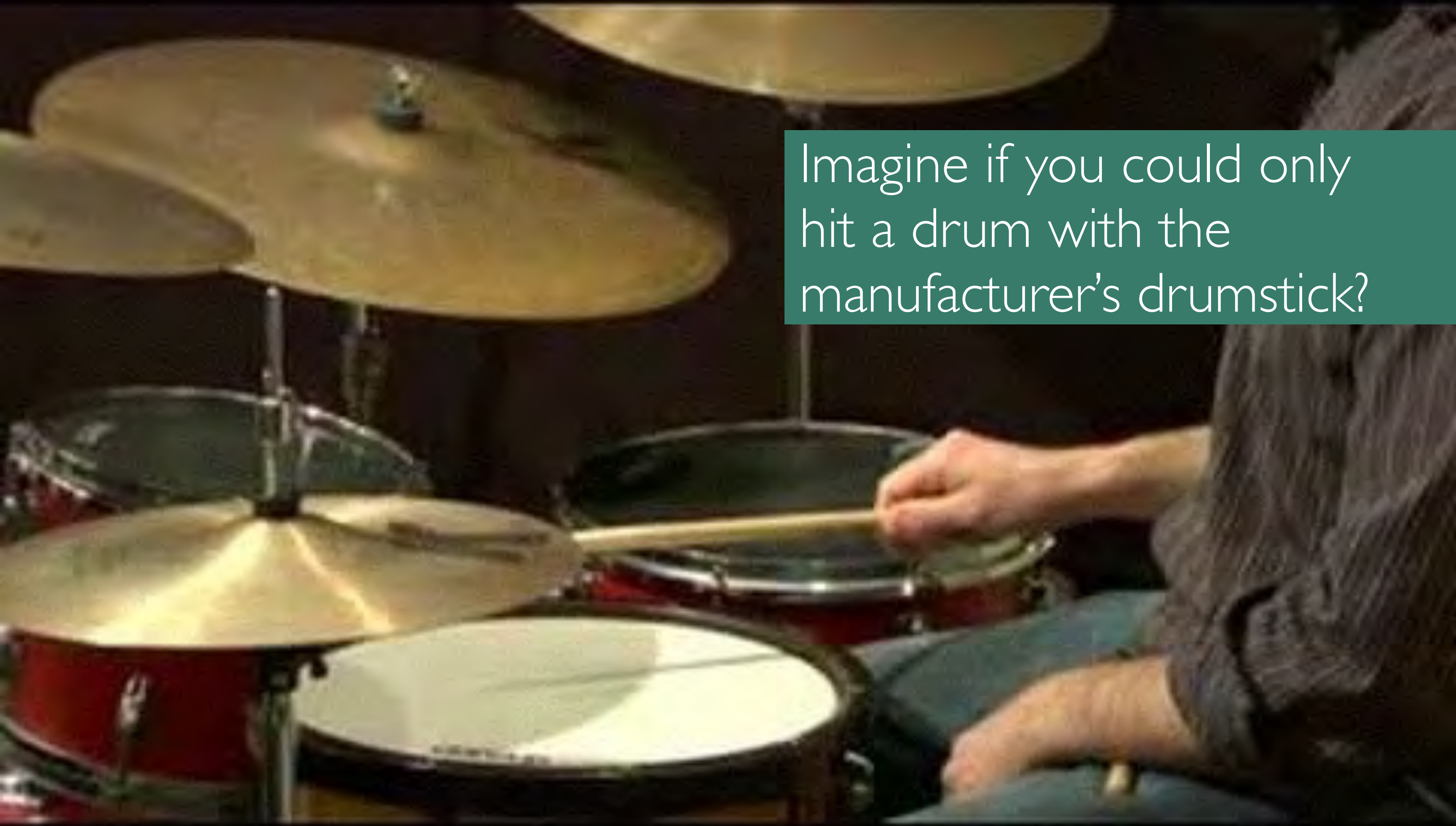
Compare to learning software:
every ‘upgrade’ changes the interface
tools belong to the application, not the user

Physical tools are defined through use

Some tools are designed for a specific task

Some tools are designed for a specific task
but we also improvise

People redefine physical tools … why not software?



Imagine if you could only hit a drum with the manufacturer's drumstick?

Our vision:

Software tools should be incrementally learnable

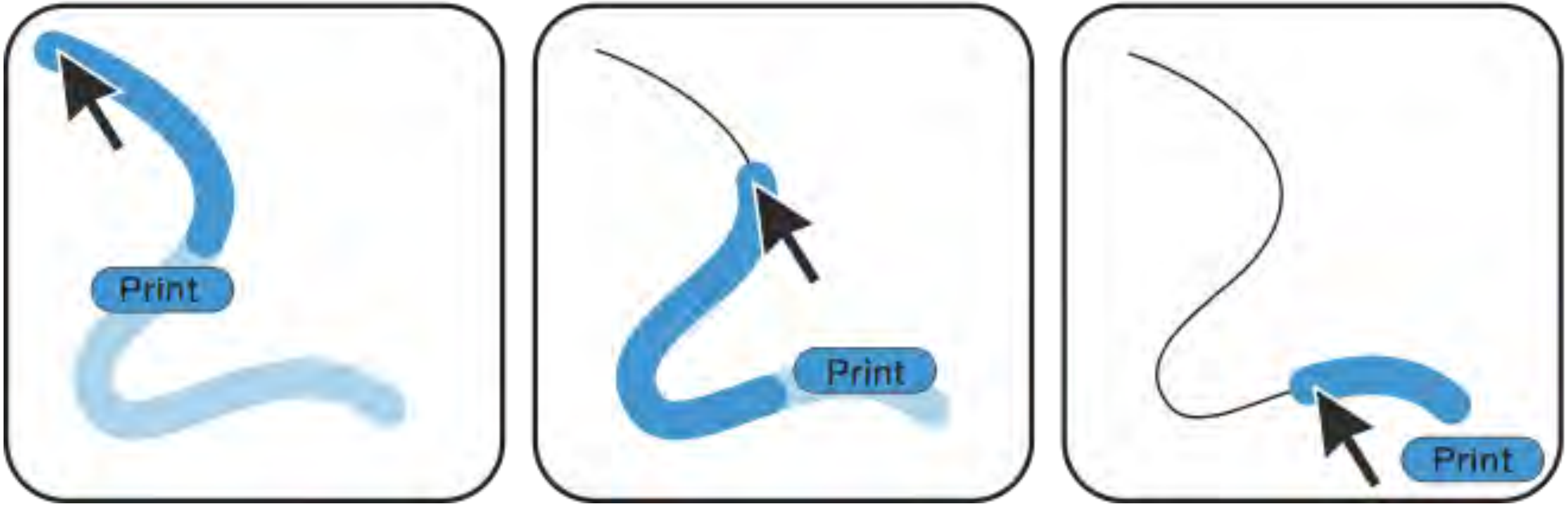
People should choose and control their own tools

Software tools should be easy to appropriate

Octopocus: Learning complex gestures

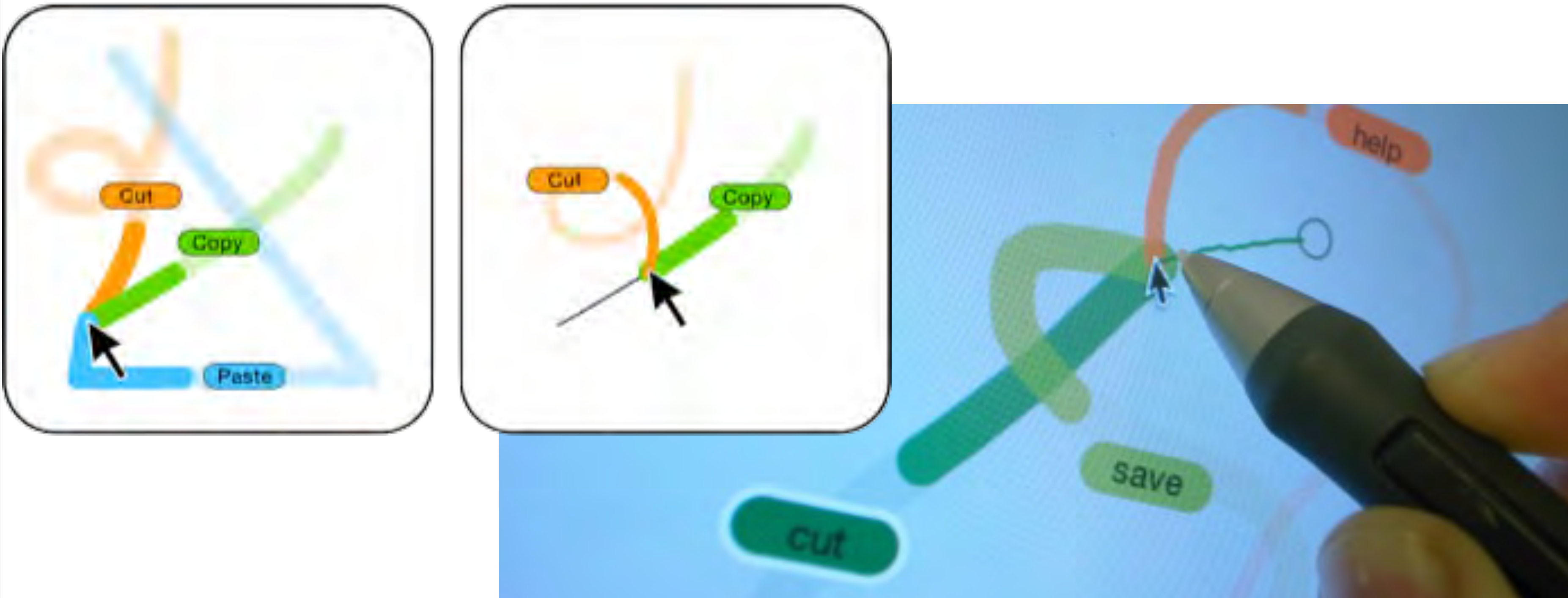
Dynamic partnership:

Progressive algorithms reveal intermediate recognition states



Octopocus: Learning complex gestures

Experts *just do it*



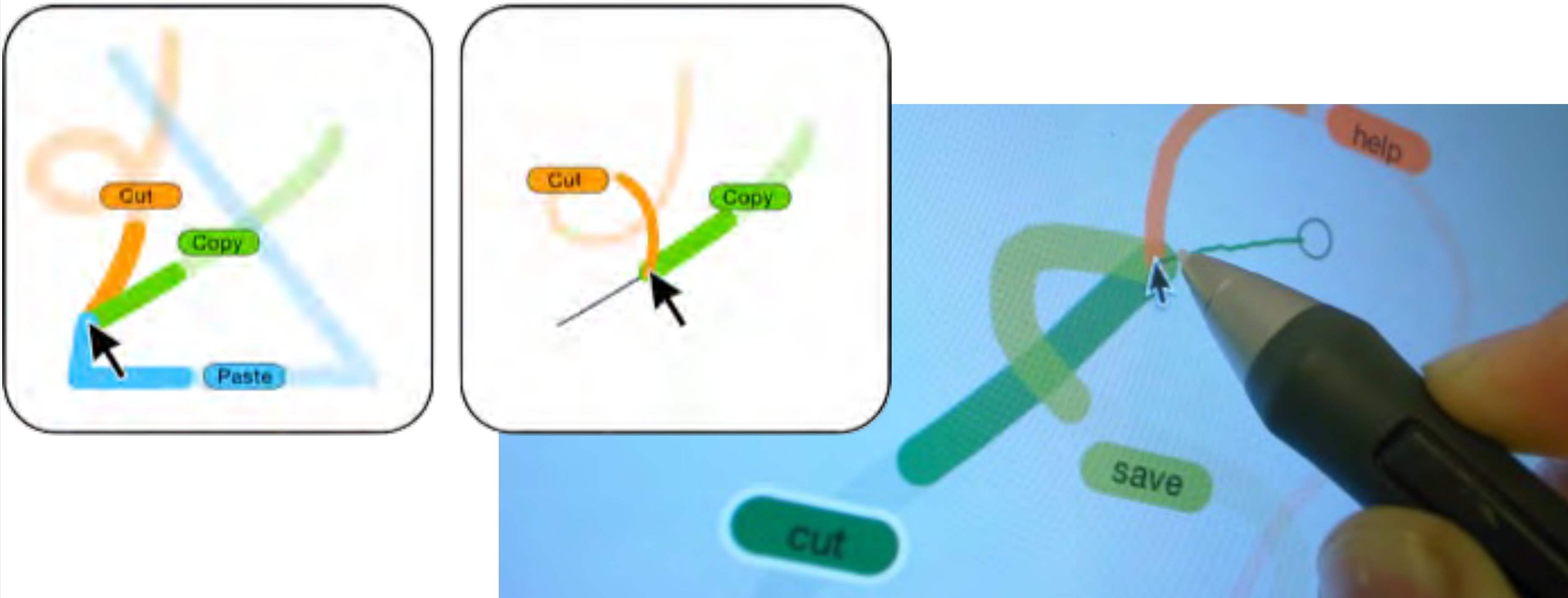
Octopocus: Learning complex gestures

Experts *just do it*

Novices *hesitate* … which activates:

feedforward shows current available gestures

feedback shows what the recognizer sees



OctoPocus

A Dynamic Guide for Learning Gesture-Based Command Sets

Olivier Bau & Wendy E. Mackay
In Situ, INRIA Saclay - LRI

UIST 2008

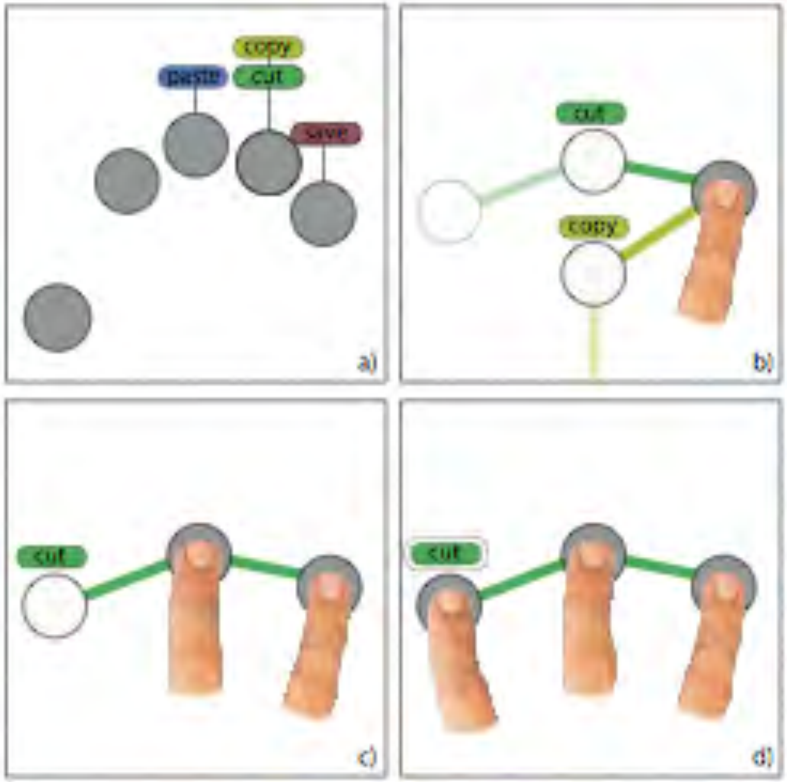
Physical tools are easy to appropriate — software tools are not

Arpege: Learning chords on a multi-touch surface

Beyond one- and two-finger gestures :

novice to expert transition

feedforward and feedback



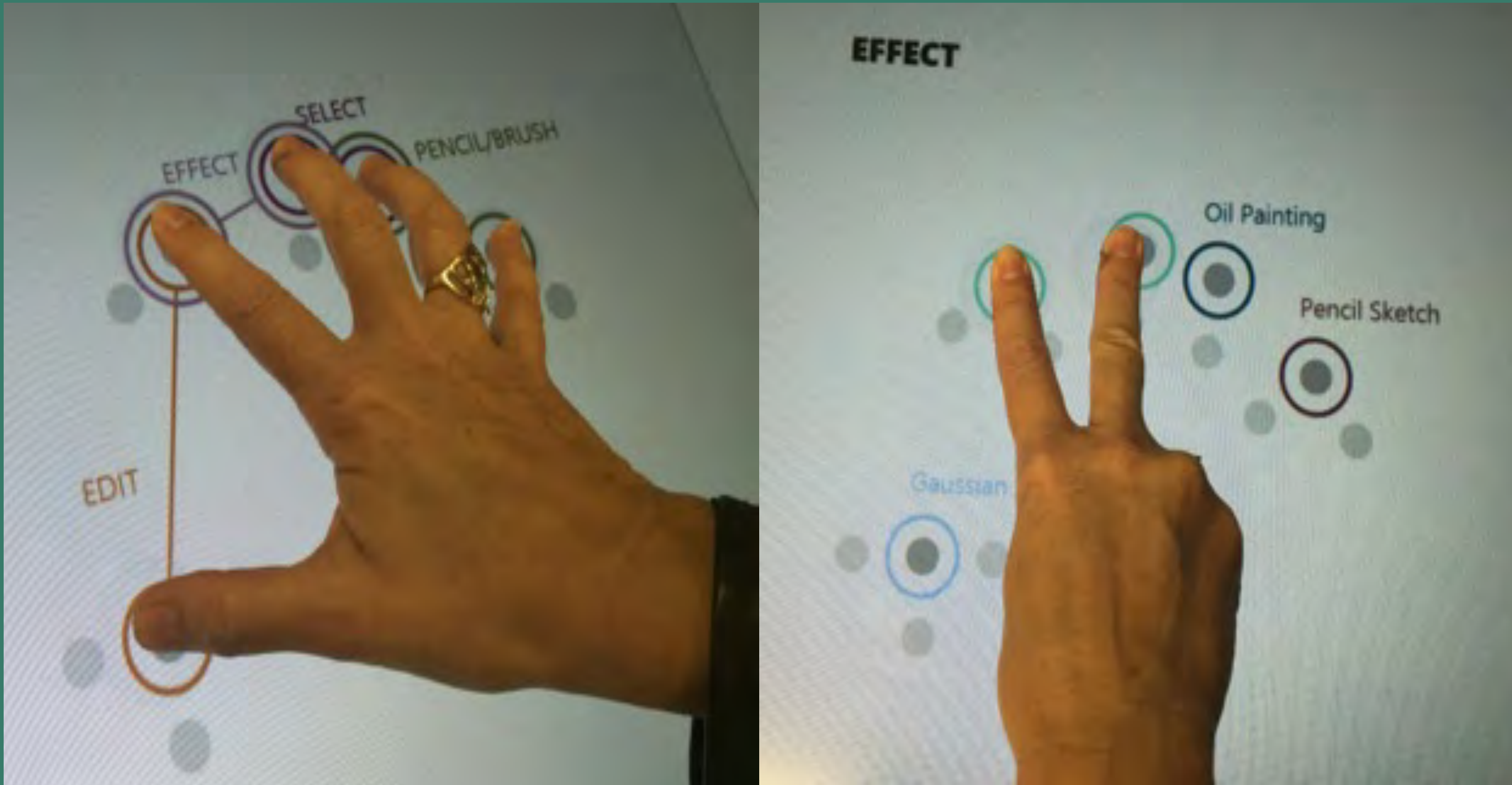
Arpege: Design and learning of multifinger chord gestures

Submitted for review
CHI 2010

Dynachord: Combining chords and gestures

Chord sequences for a larger chord vocabulary

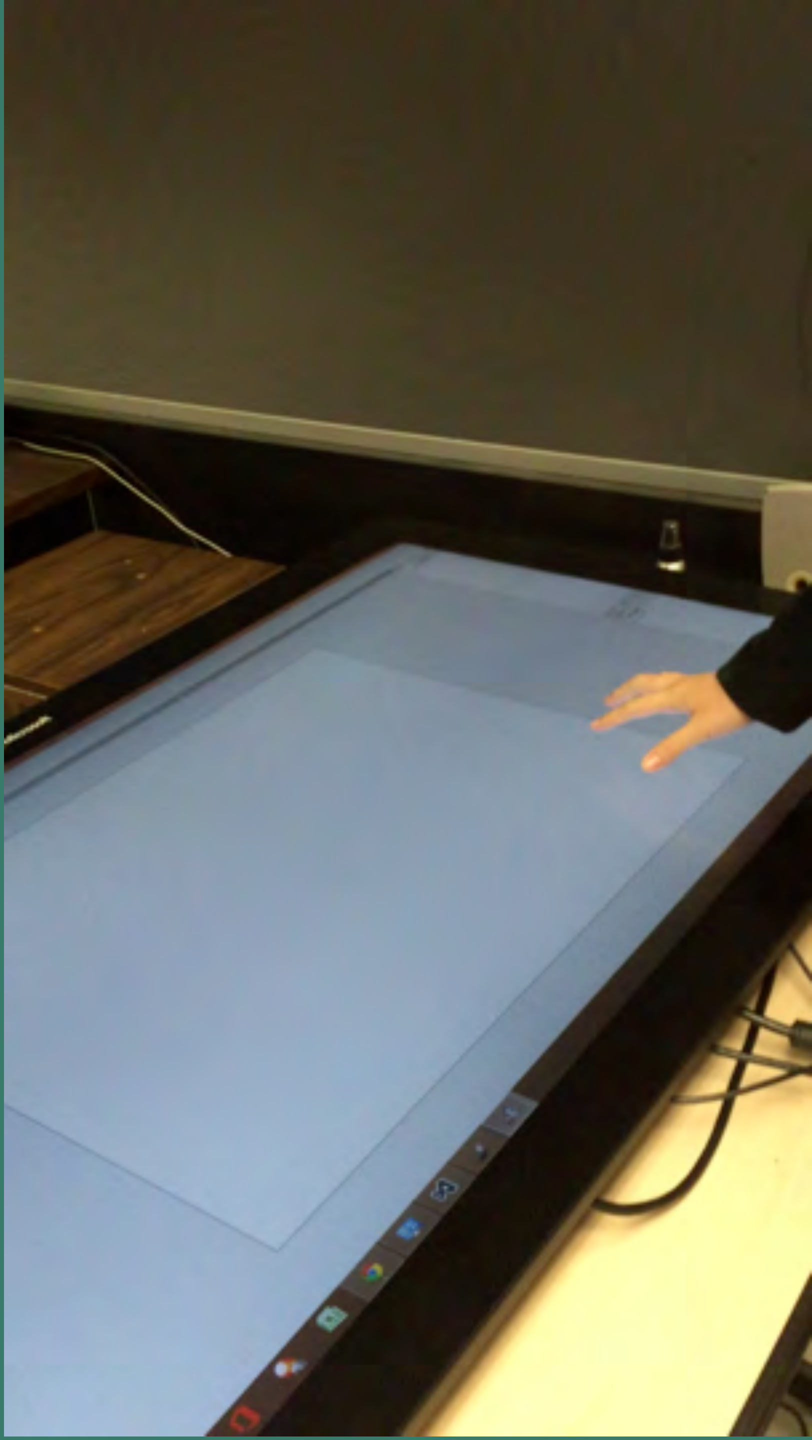
Dynamic adjustment of parameters



Dynachord

Enter a chord with one hand to choose a color

Continuously adjust the color with the other hand



How can we help users choose and control their own tools ?

Appropriation

Interaction designers usually assume that users will focus on their system and use it as intended

Users often use systems in different ways

- They may have a different mental model of the system
- They may turn ‘mistakes’ into opportunities
- ‘Bugs’ become ‘features’

Anything that involves communication among people is usually adapted for new purposes

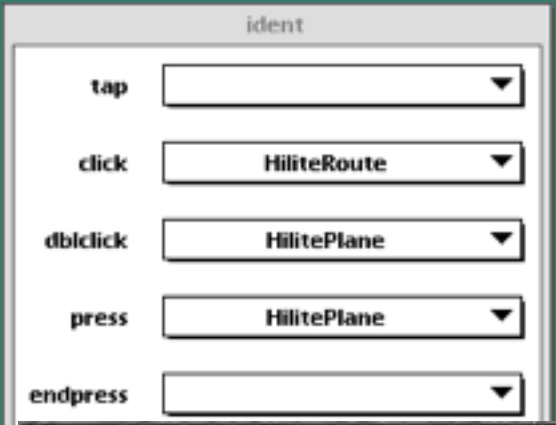
How can we help users appropriate technology ?

Creating a partnership in which
the user defines the **semantics** of the interaction
with the computer

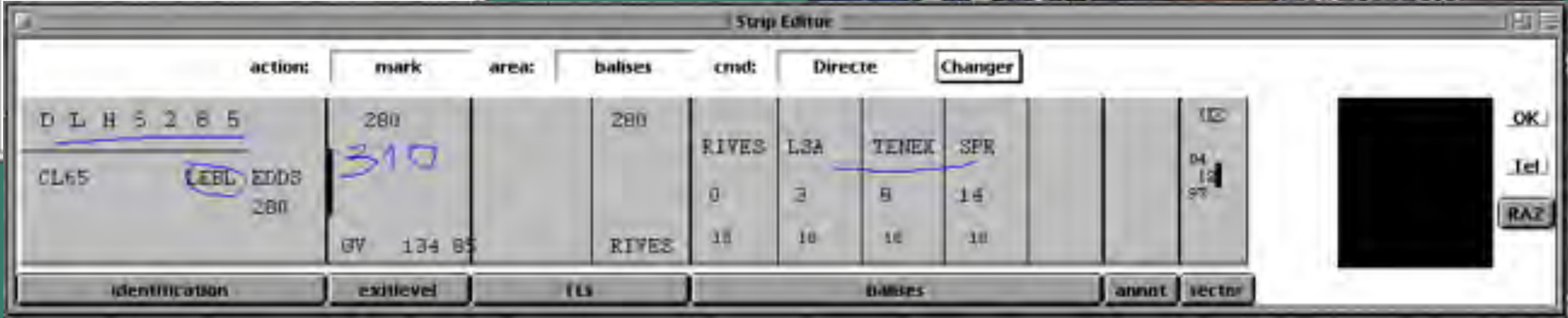
Interaction Browser :	Linking marks to actions
Knotty Gestures :	Interacting while writing
Musink :	Creating a user-defined language
Façades :	User-reconfigurable interfaces

Interaction browser: User-defined commands

- Air traffic controllers annote flight strips
  - Marks can be linked to RADAR and other computer functions
  - Users define what marks mean


ident

tap	
click	HiliteRoute
dblclick	HilitePlane
press	HilitePlane
endpress	


Strip Editor

action: mark area: balises cmd: Directe

D L H 5 2 6 5
CL65 LEBL EDDS
280

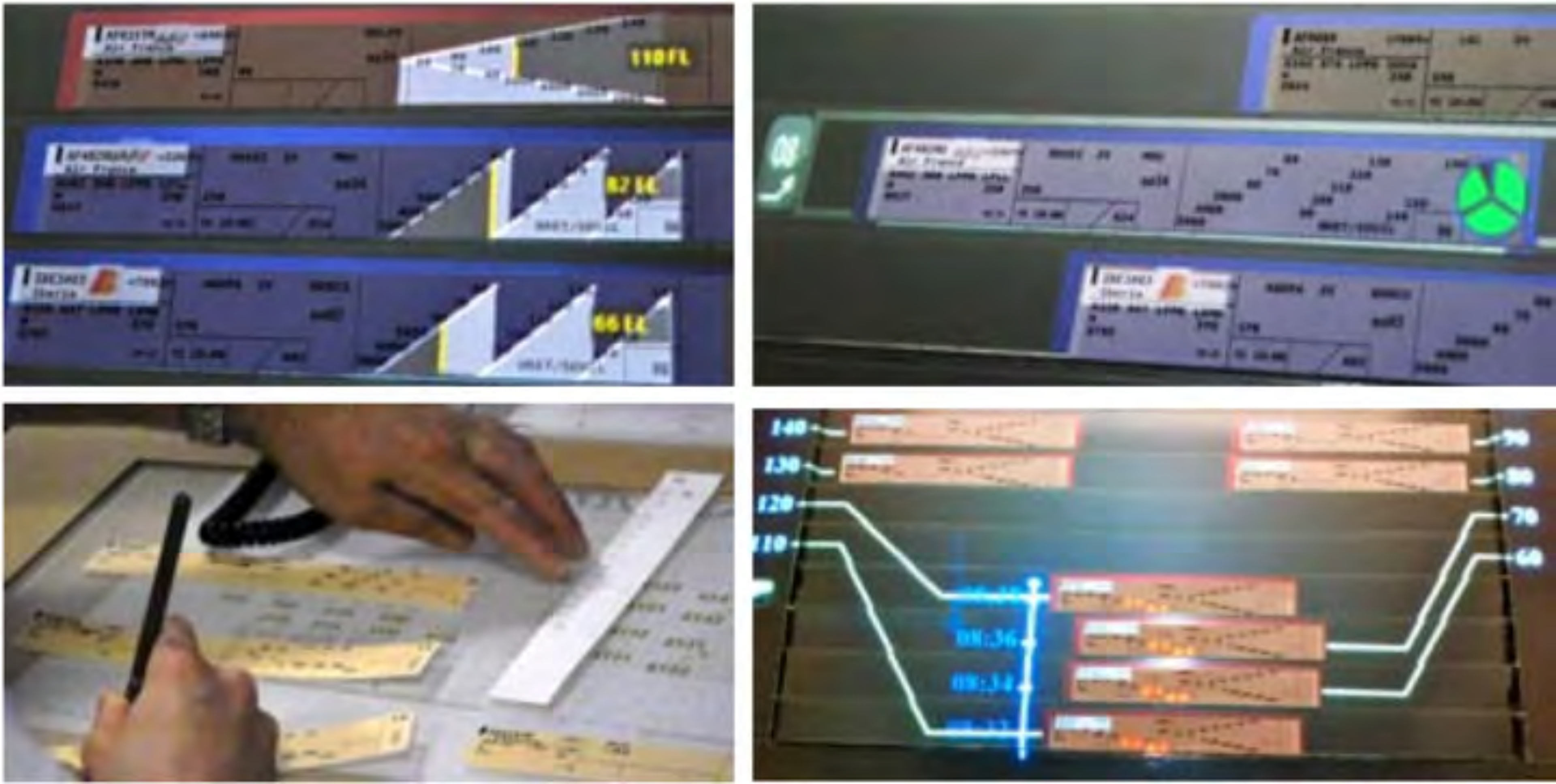
280
310
GV 134 85

280
RIVES

RIVES	L3A	TENEX	SPR
0	3	8	14
18	18	18	18

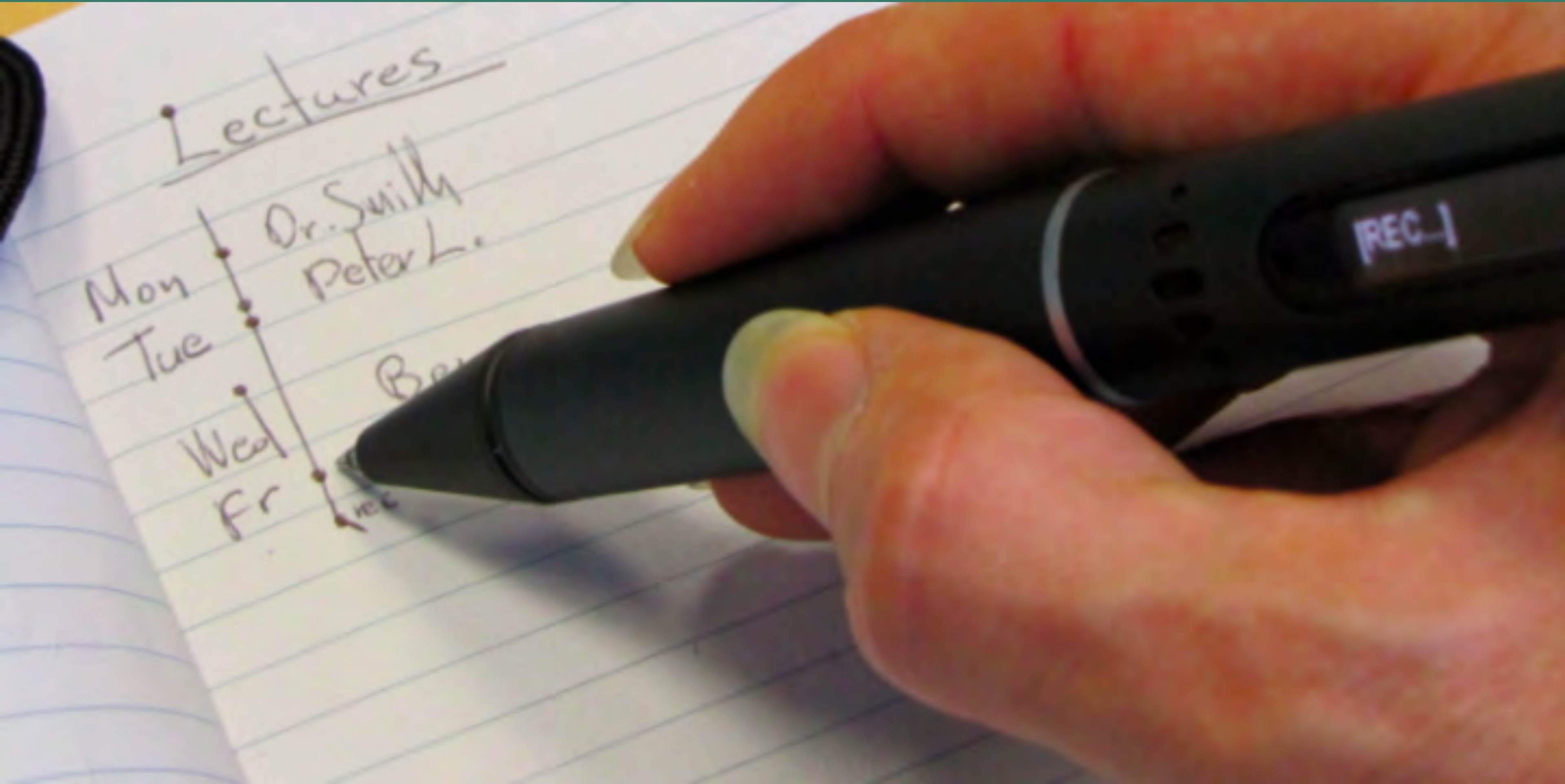
Striptic

Flights in my Hands: Coherence Concerns in Designing a Tangible Space for Air Traffic Controllers, (Letondal et al., CHI'14)



Knotty Gestures

Draw a dot, define a command
Interact while writing
Interact with command later



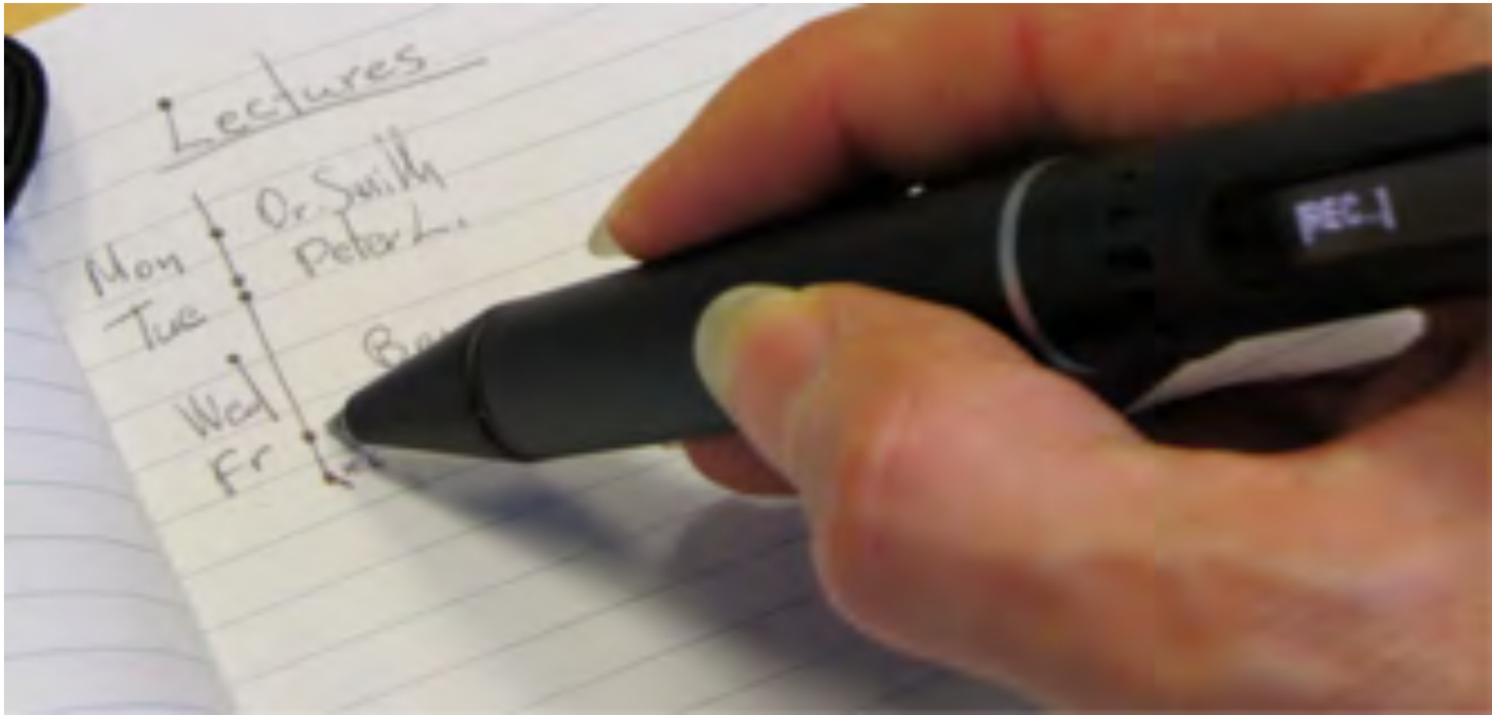
Knotty Gestures

Interactive Paper

Users interact as they write

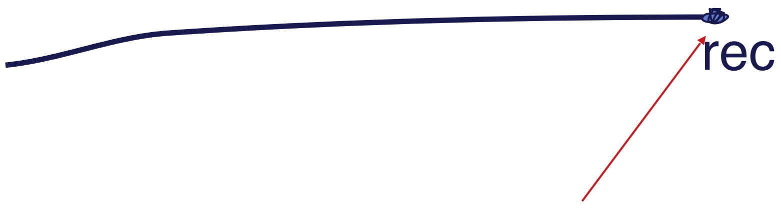
or define their own gestures

and interact with them later



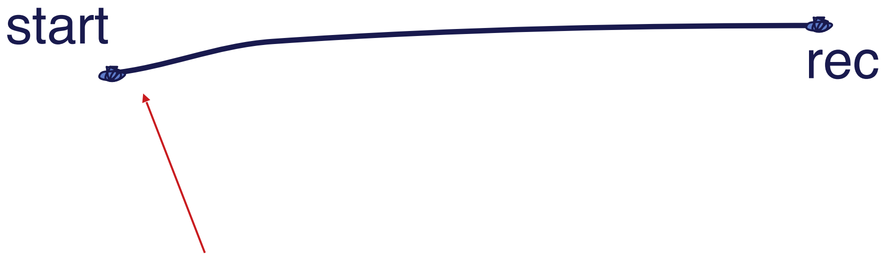
Knotty Gestures: Creating an interactive controller

Draw a line with a 'knotty gesture' at the end



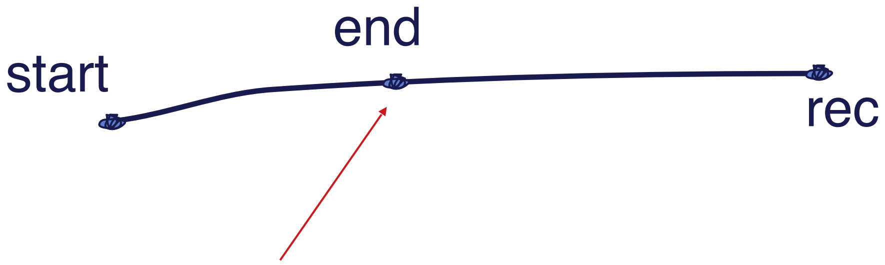
Choose "recording" to define the type of line

Knotty Gestures: Creating an interactive controller



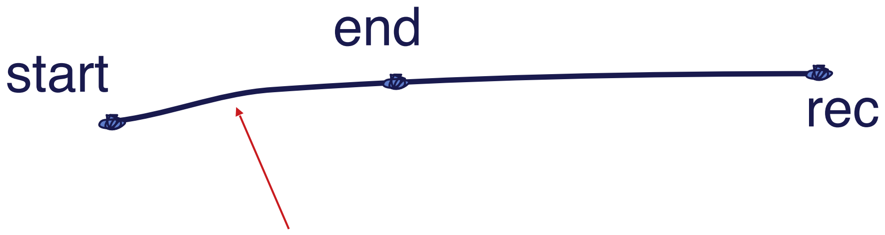
Define where the recording will start

Knotty Gestures: Creating an interactive controller



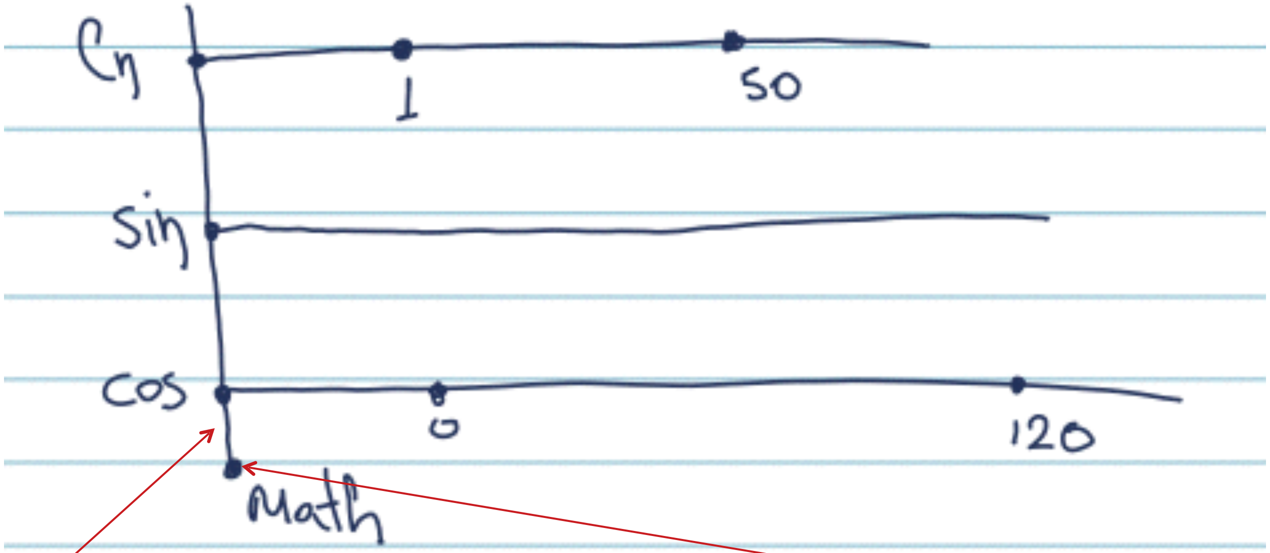
Define an end point for the recording

Knotty Gestures: Creating an interactive controller



Slide the pen along the line to move forward or backward on the recording

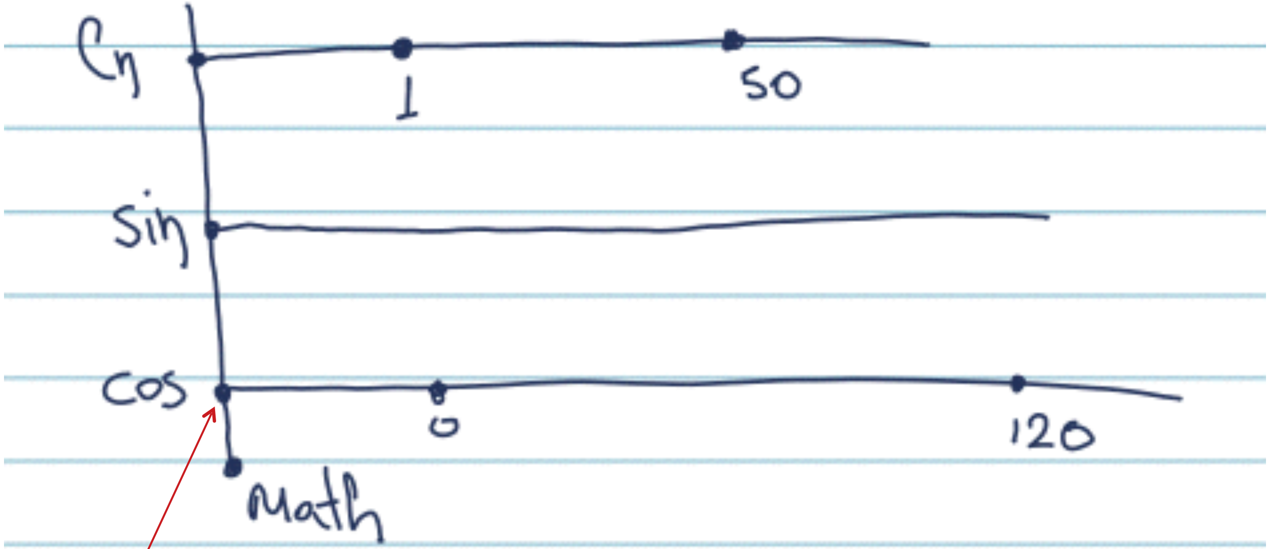
Drawing a Math Calculator



This line acts as a base for attaching mathematical value sliders

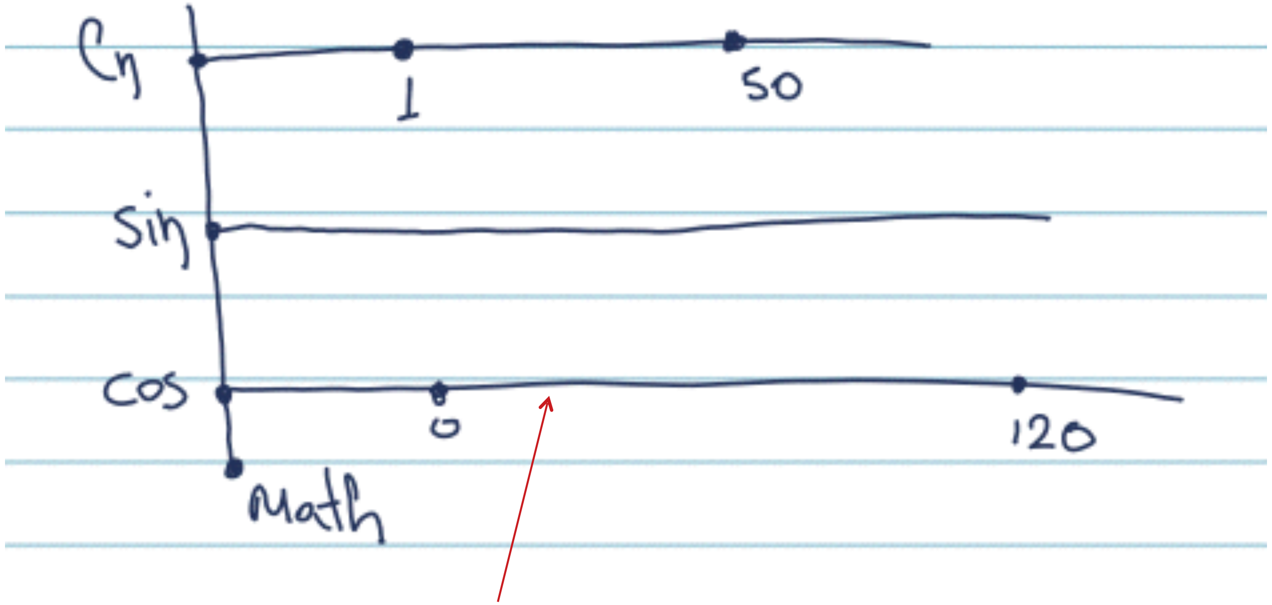
The knotty gesture at the end defines the type

Drawing a Math Calculator



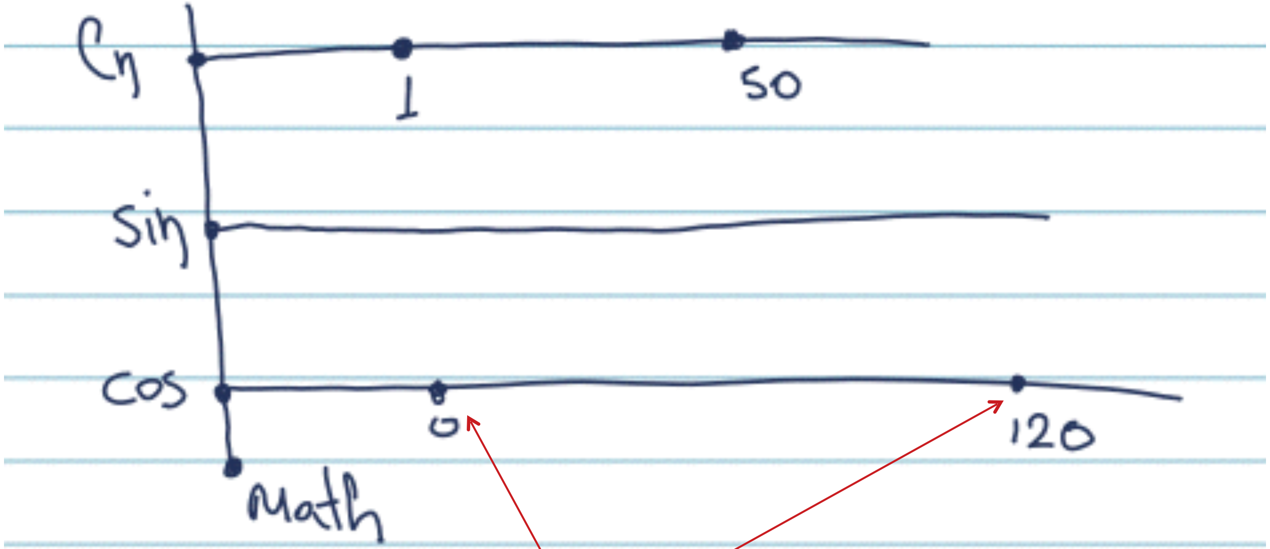
Any knot drawn on line lets the user select a mathematical function

Drawing a Math Calculator

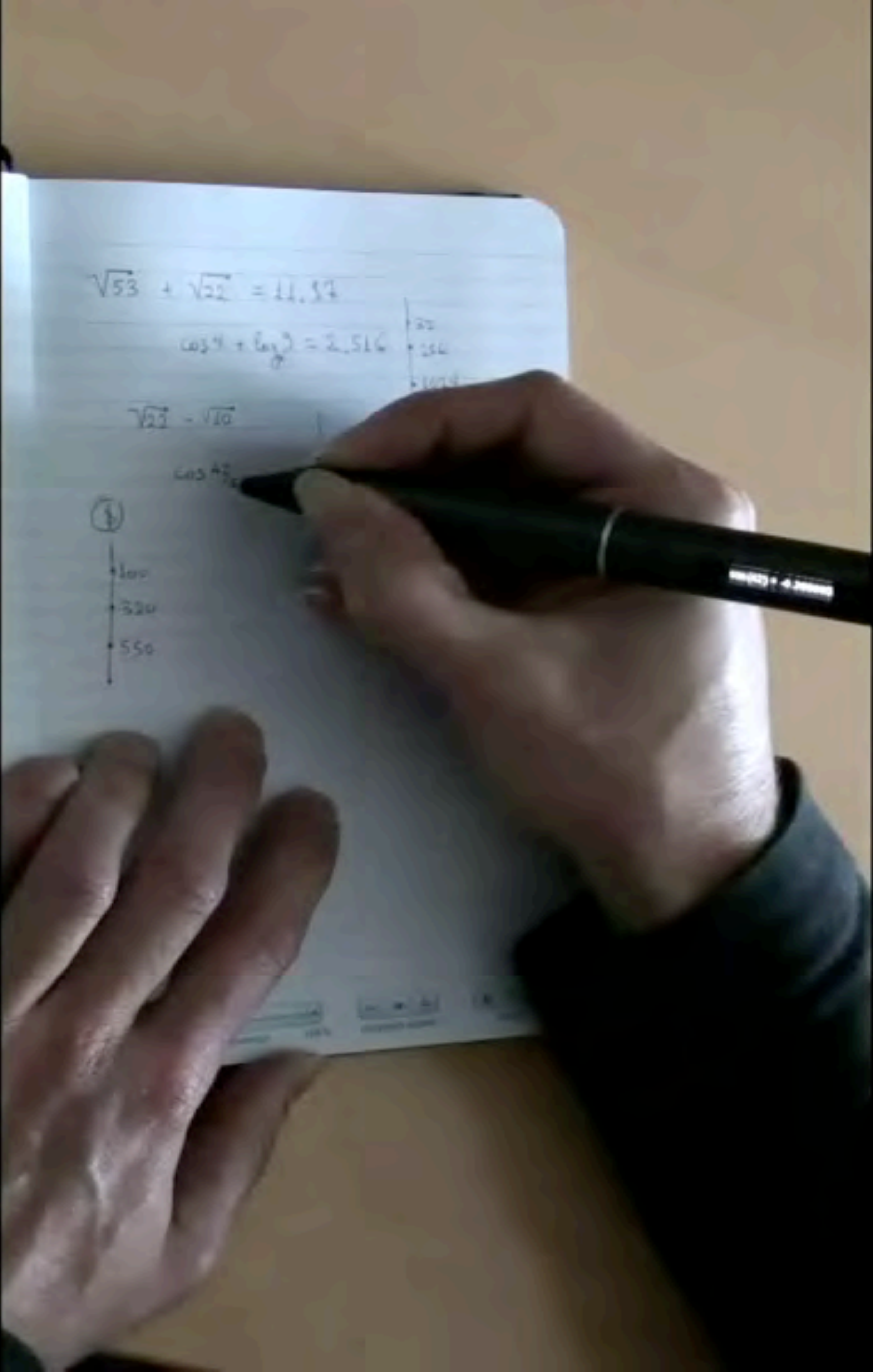


The extensions act as value controllers
Sliding the pen over the line moves through range of function values, shown on the pen display

Drawing a Math Calculator



Knots may define ranges or act as traces of past interactions with specific values



$\sqrt{53} + \sqrt{22} = 11.97$

$\cos 4 + \log_8 3 = 2.516$

$\sqrt{22} - \sqrt{10}$

$\cos 45$

But recognition is not the only problem …

Recognition must be *good enough*

- but users override and reinterpret
- no single ‘correct’ interpretation
- recognized and non-recognized gestures co-exist

Real question:

- Can Mus*ink* support the creative process?
- What are the design implications for Mus*ink* v2?

Semi-Structured Delayed interpretation

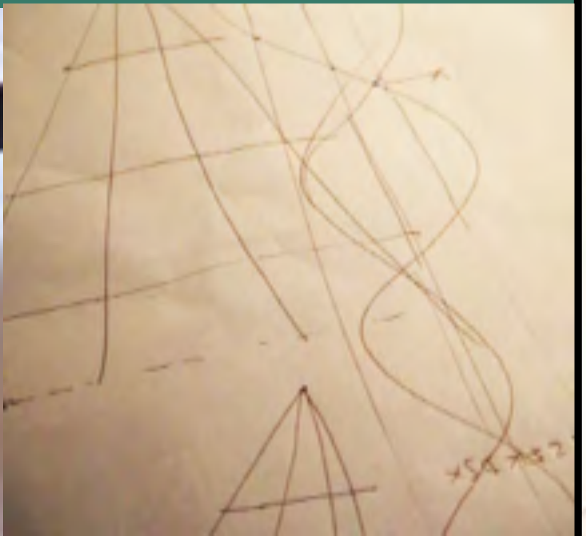
Key insights:

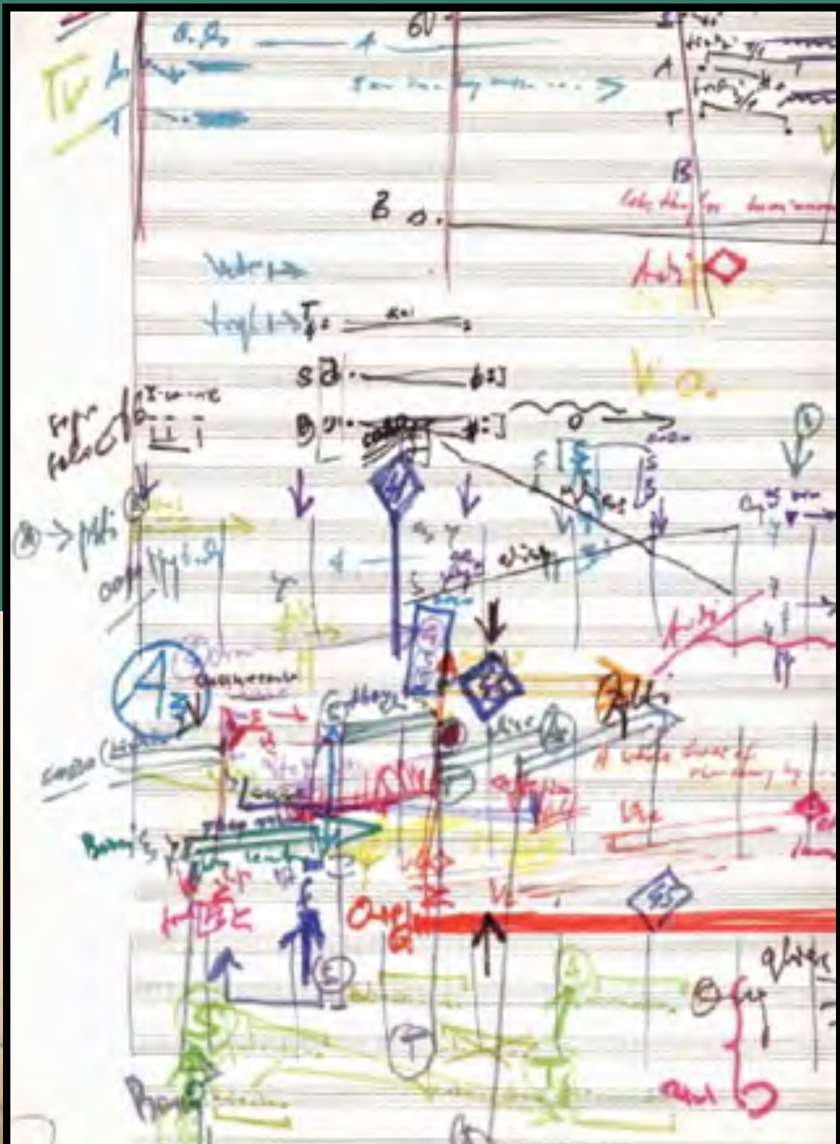
- Spatial structure on paper
    - improves recognition
    - under user's control
- Recognition need not be immediate
    - users decide *when* to intrepret
    - interpretation *changes over time*

Musink

Musicians create their own musical languages on paper

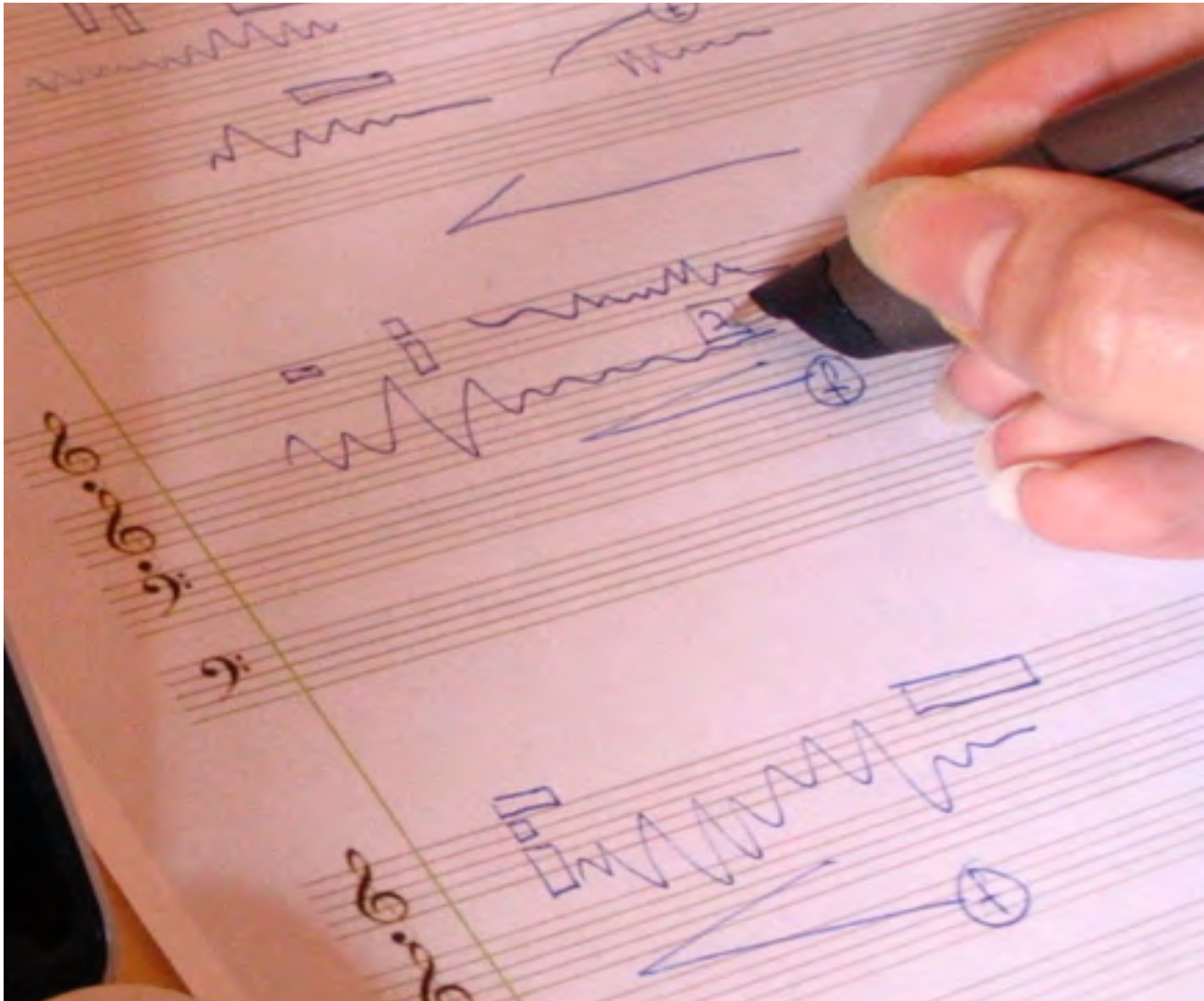
… and go back and forth between paper and computer

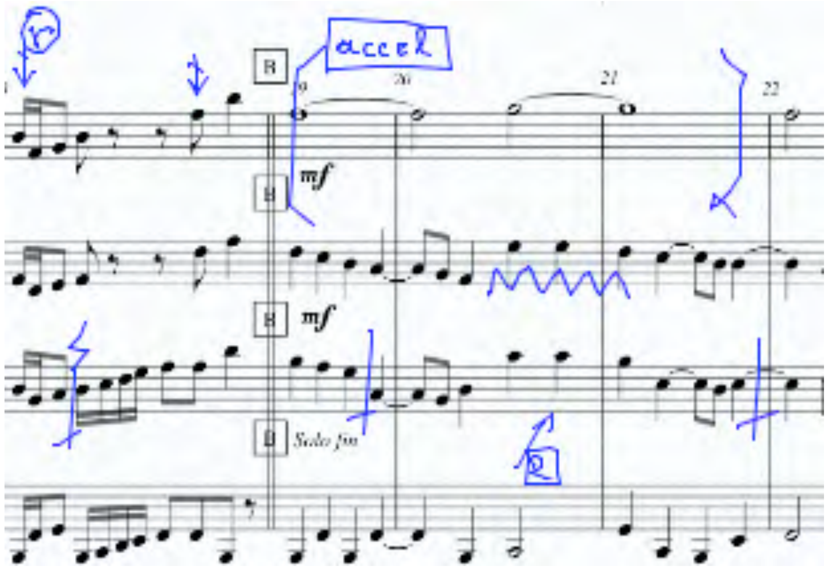


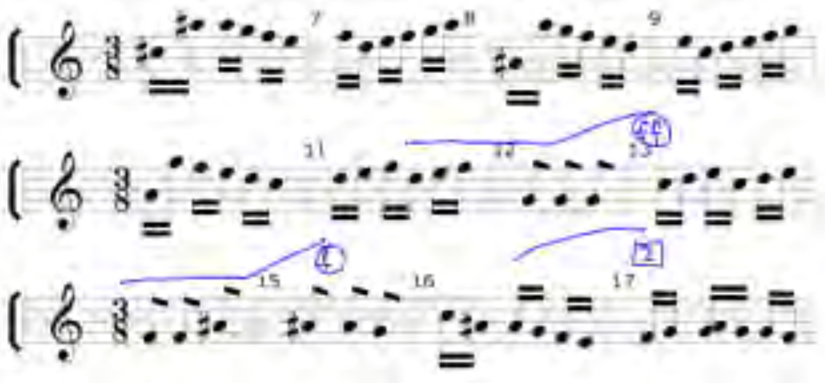


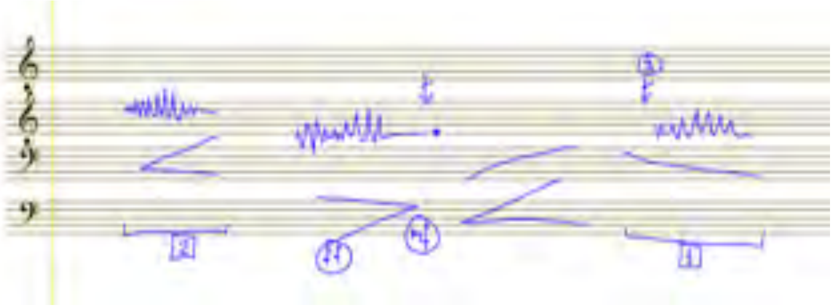
Mus*Ink* Define meaning of gestures over time

User decides if and when to interpret each gesture



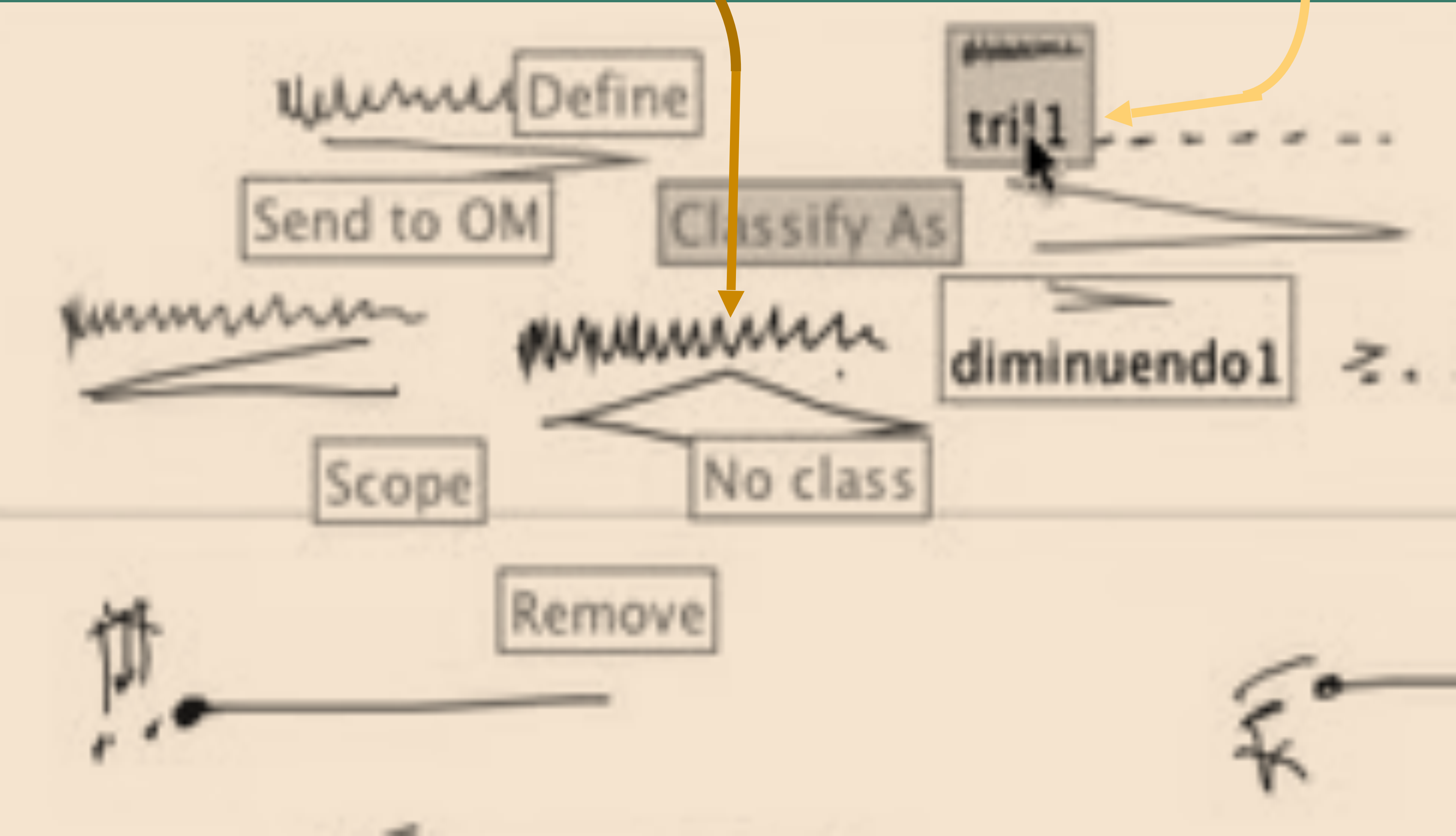




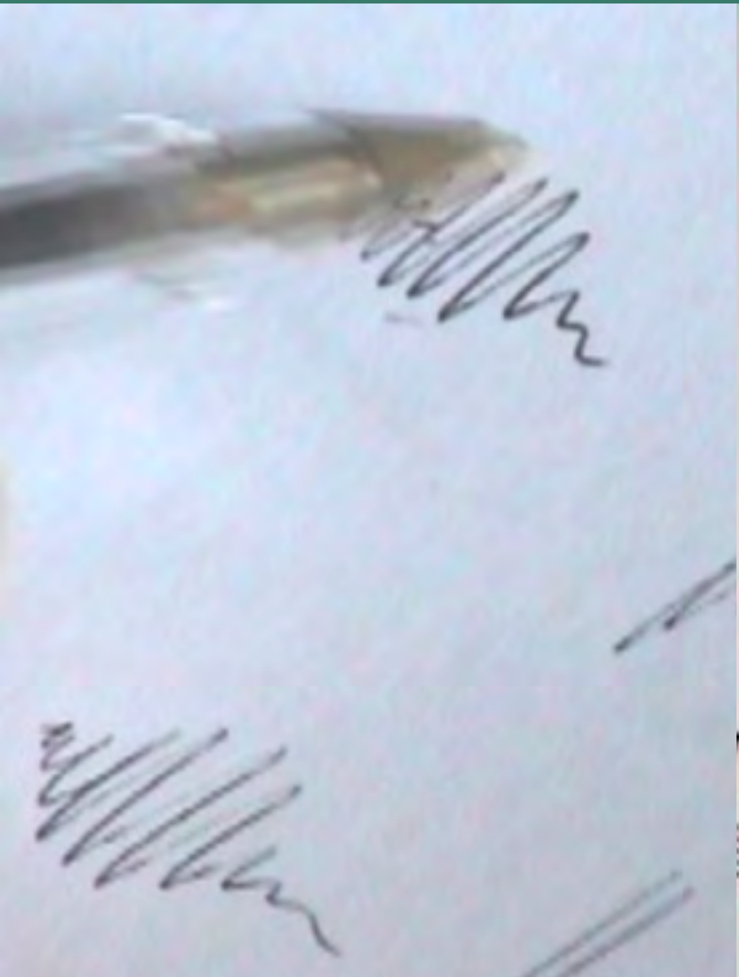


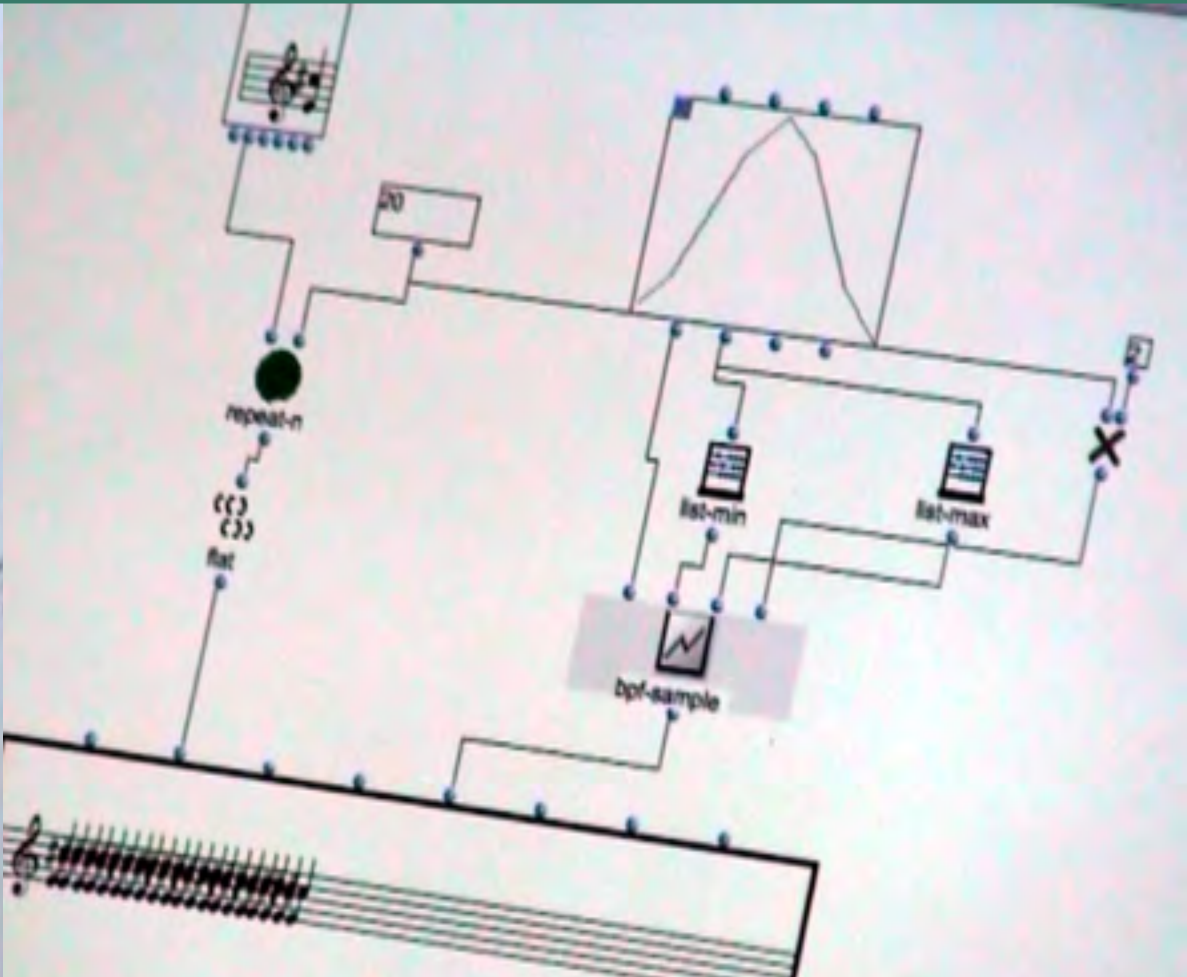
Create interactive annotations

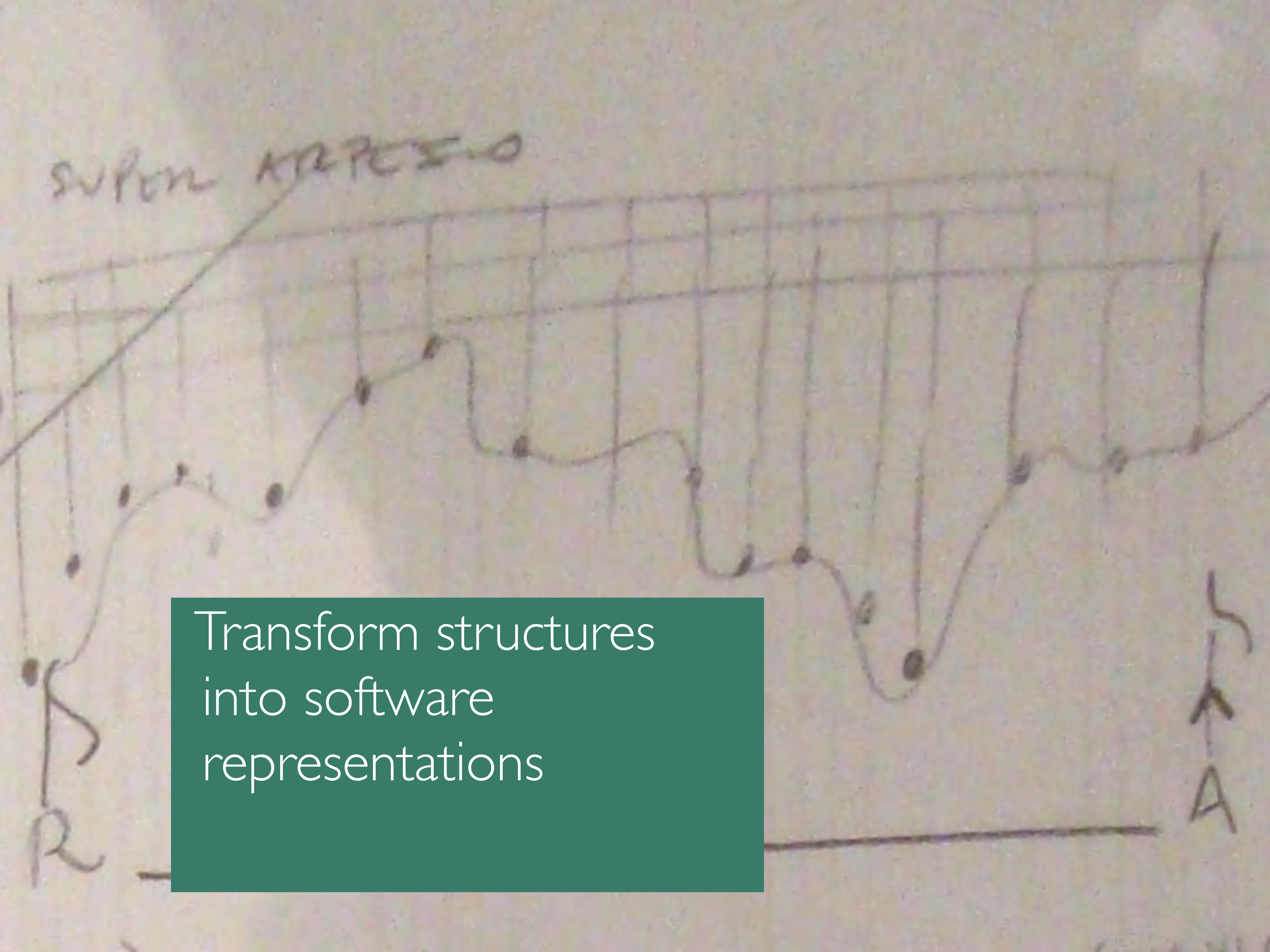
Reclassify a 'squiggle' and turn it into a trill



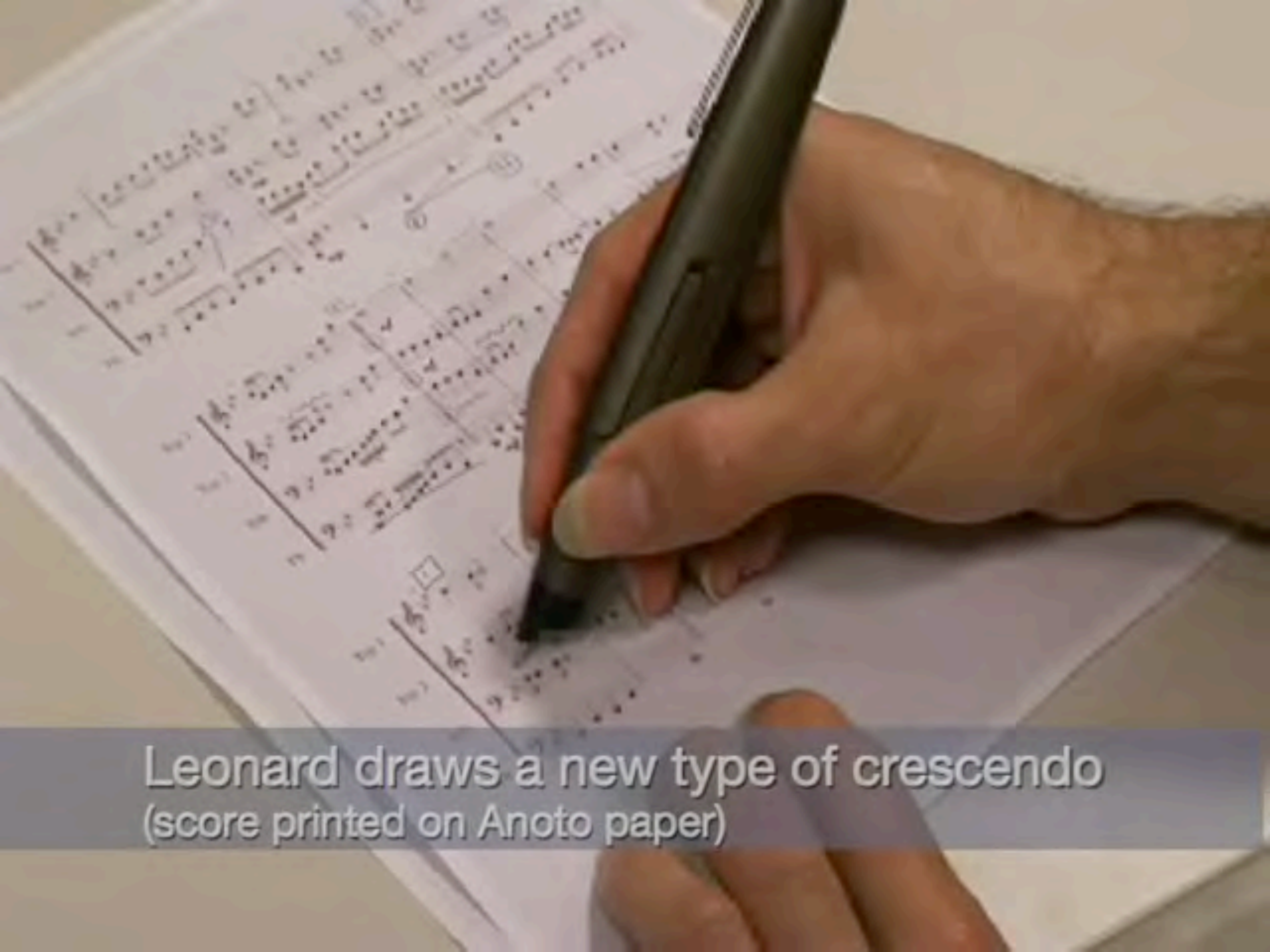
From symbols to wave forms:
Interpret a tremolo gesture as a waveform by *OpenMusic*







Transform structures into software representations

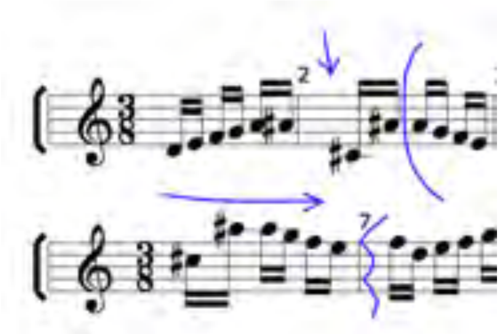


Leonard draws a new type of crescendo
(score printed on Anoto paper)

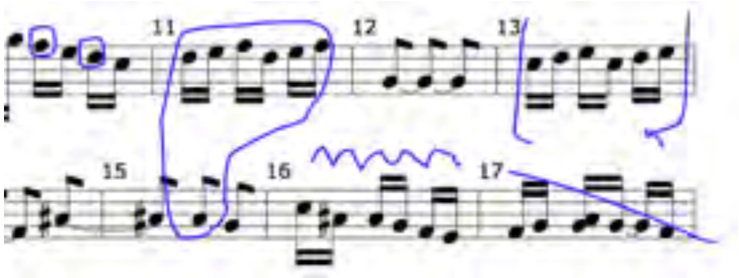
Musink: Semi-structured, delayed interpretation

Users decide when and how each annotation should be interpreted by the computer

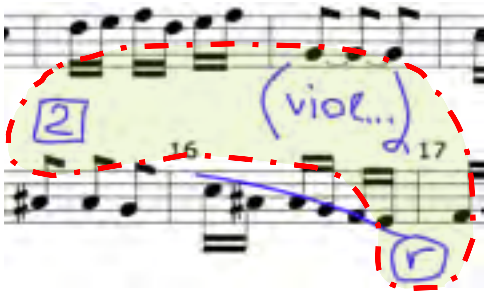
score pointers



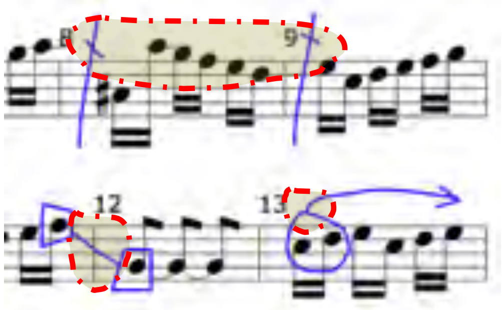
scoping gestures



textual elements



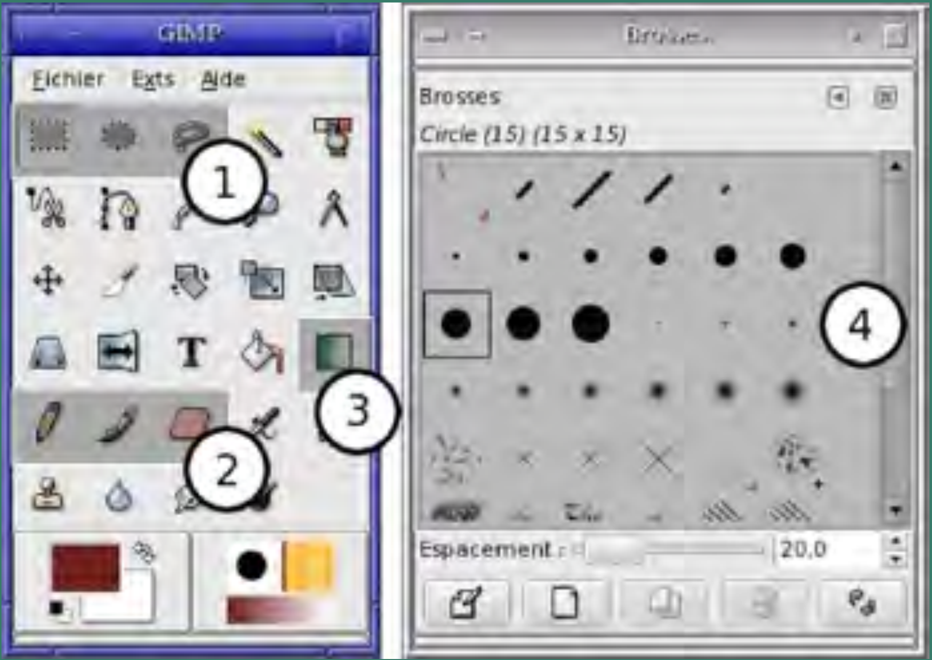
connectors

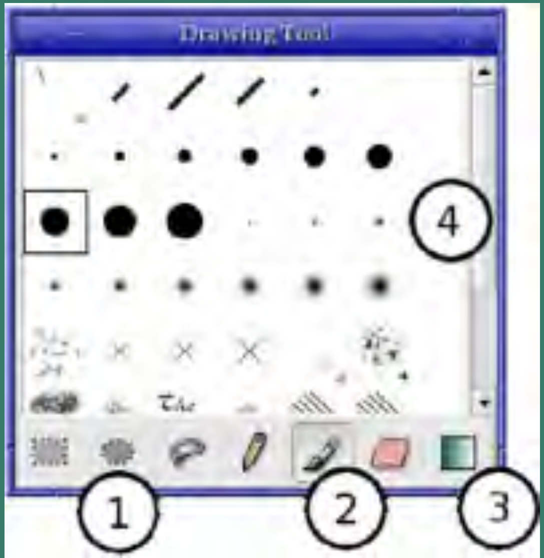


Façades: Reconfiguring interfaces

Users can adopt parts of **any** Linux interface
and reconfigure it for specific needs

Grab three selections from GIMP and choose a brush
and create a new, custom-made palette





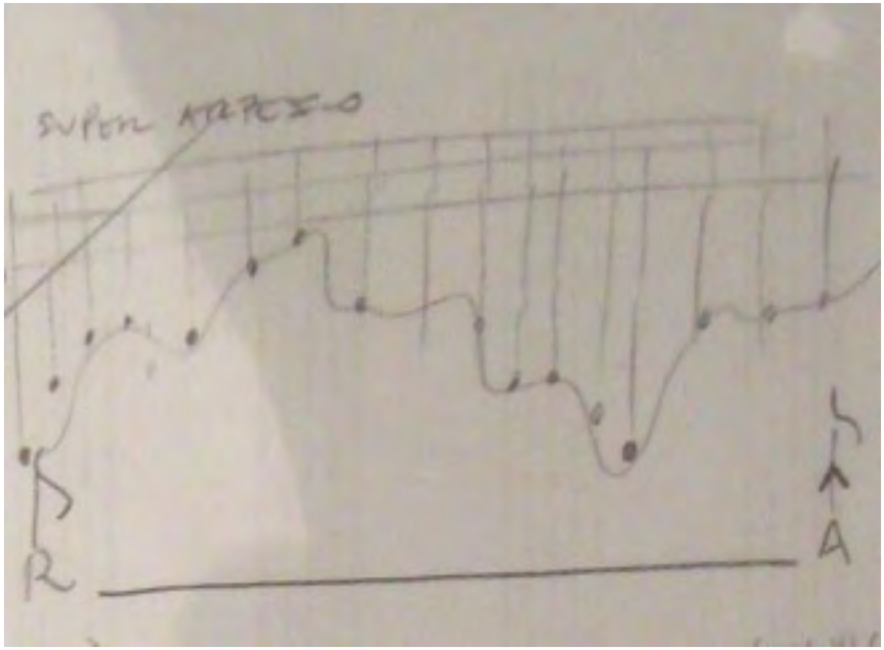
Substrates

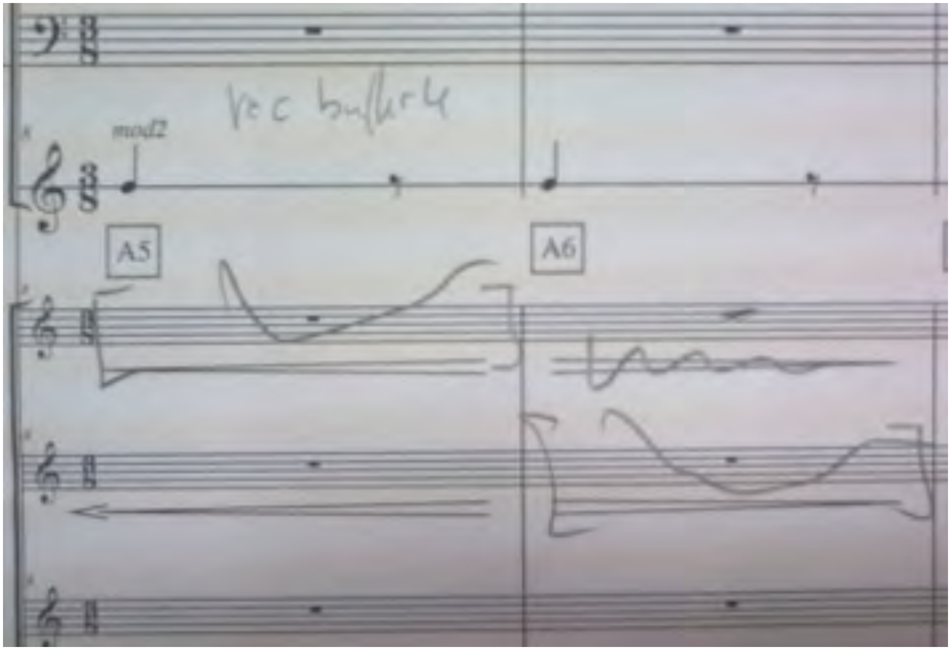
Define the structures and rules

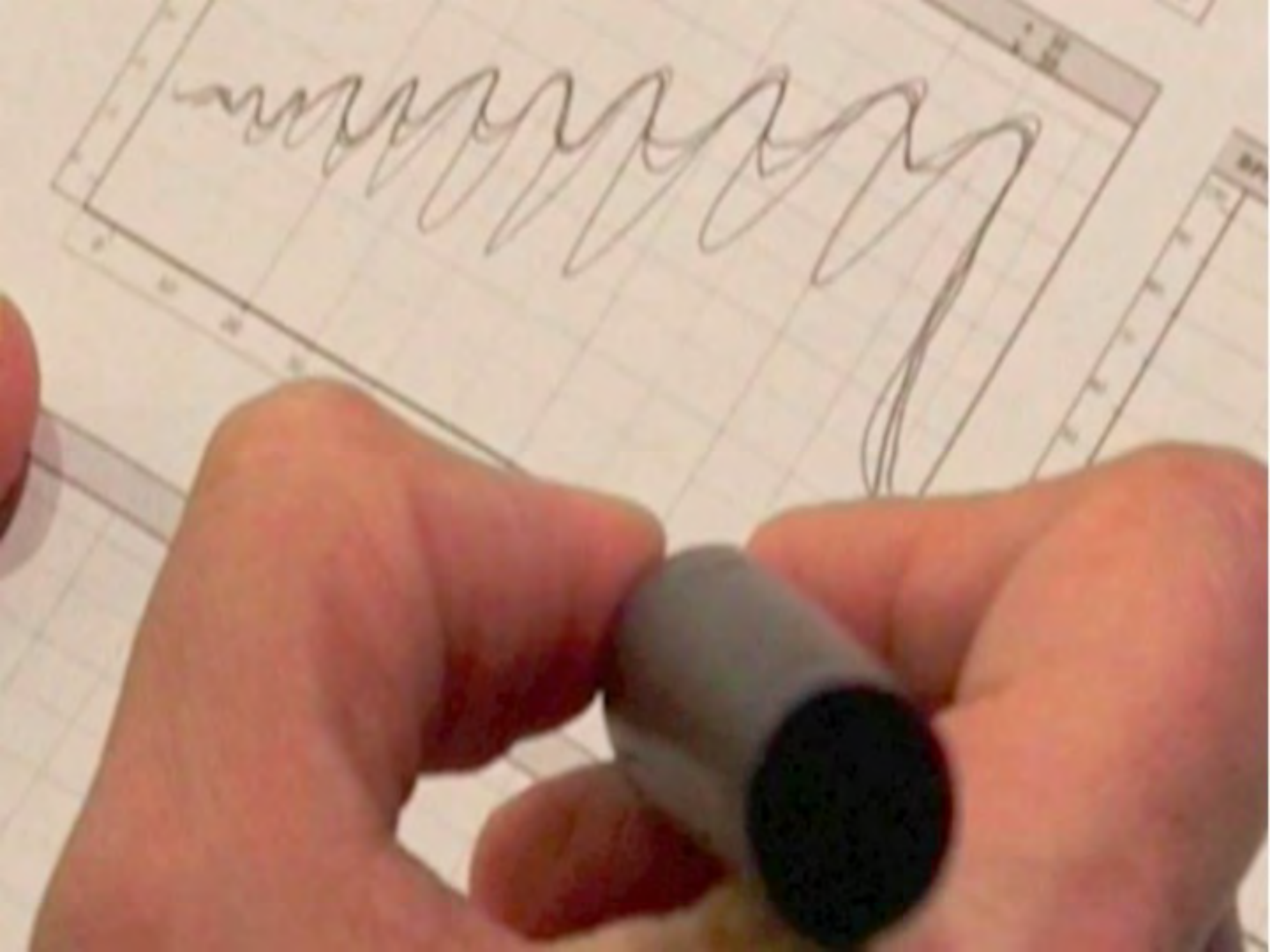
Ways to interpret the data

Different structures

to facilitate
interpretation

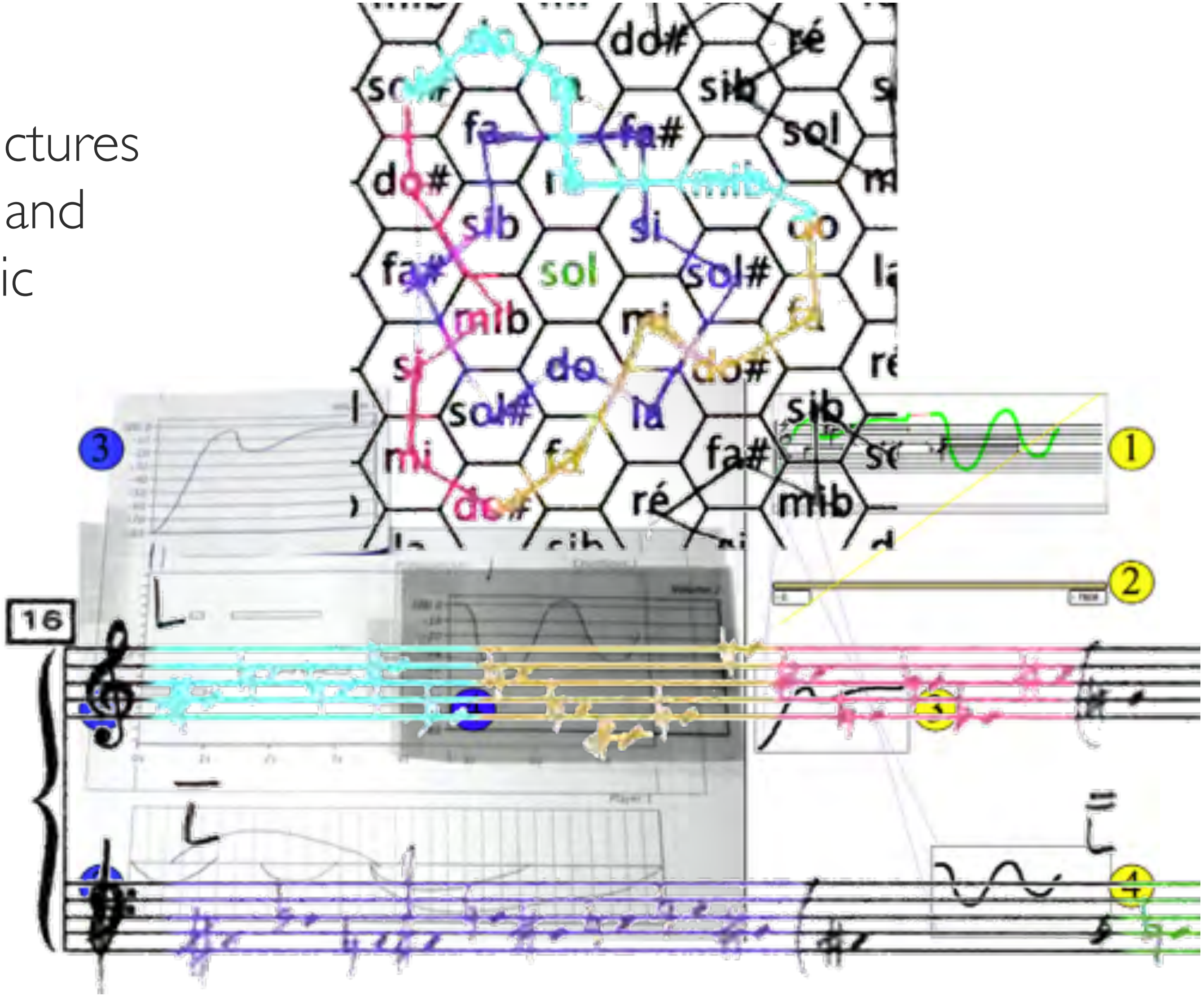


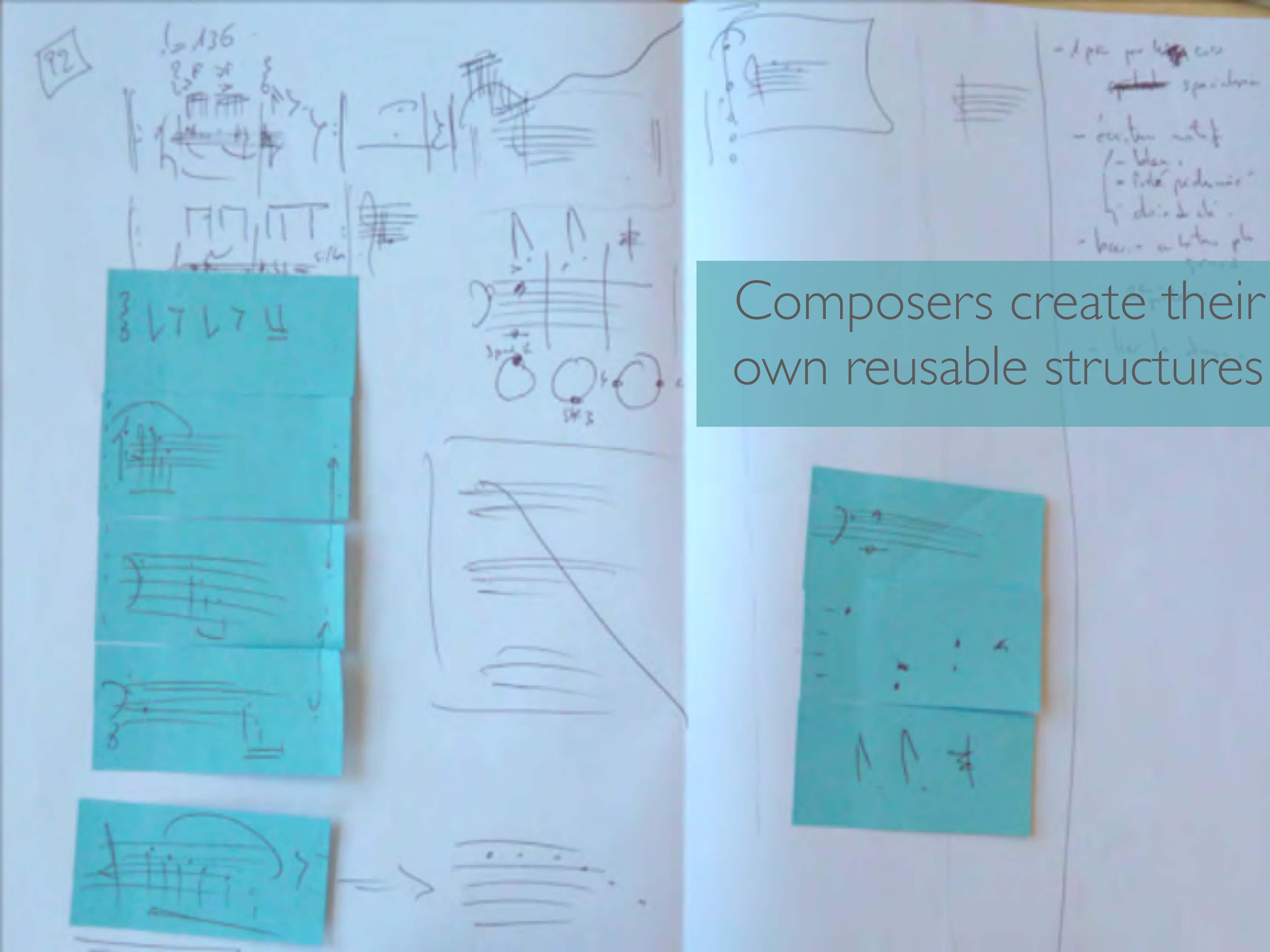




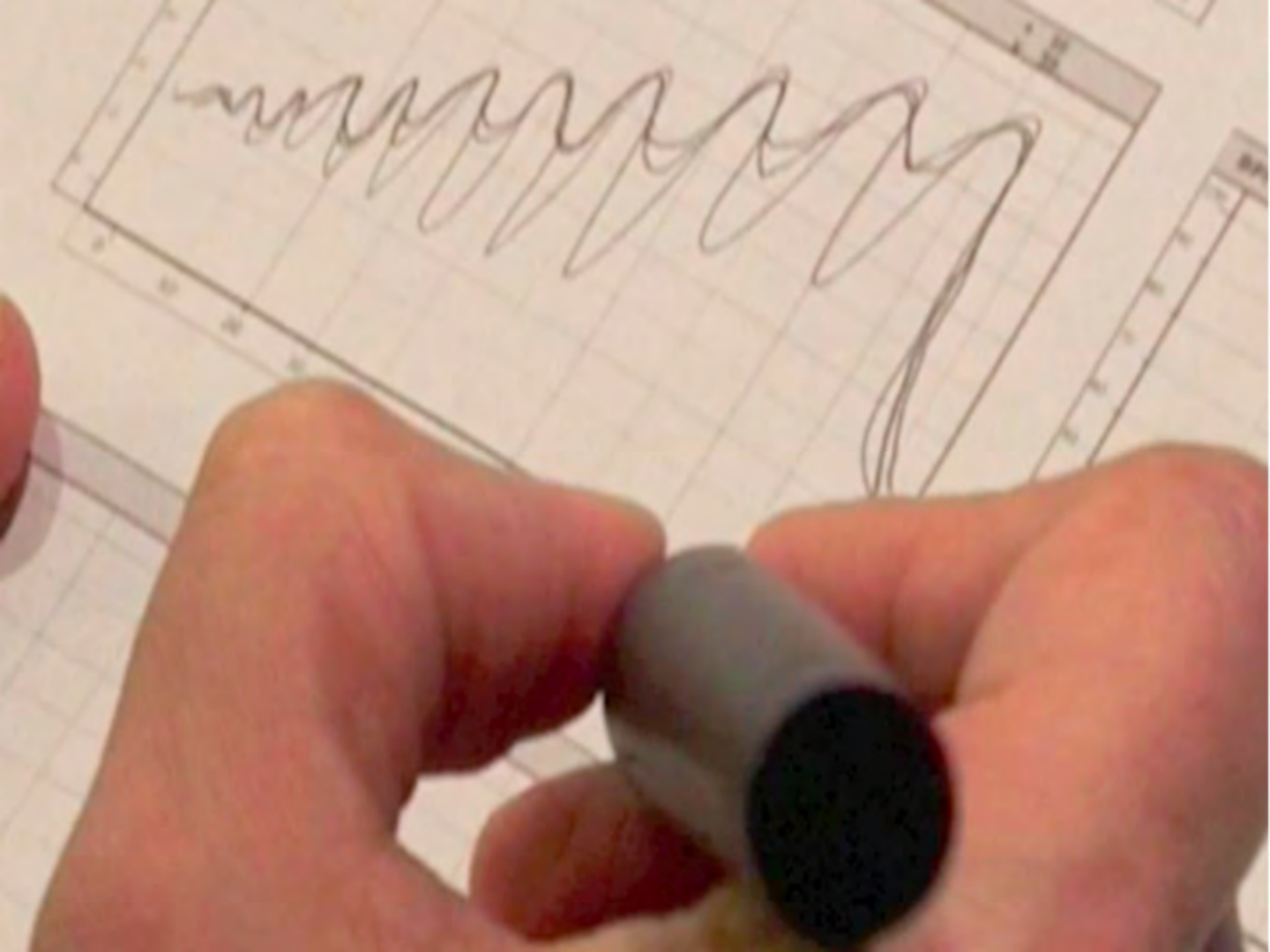
Paper Substrates: create own language & structure

Composers
create new structures
for interpreting and
composing music



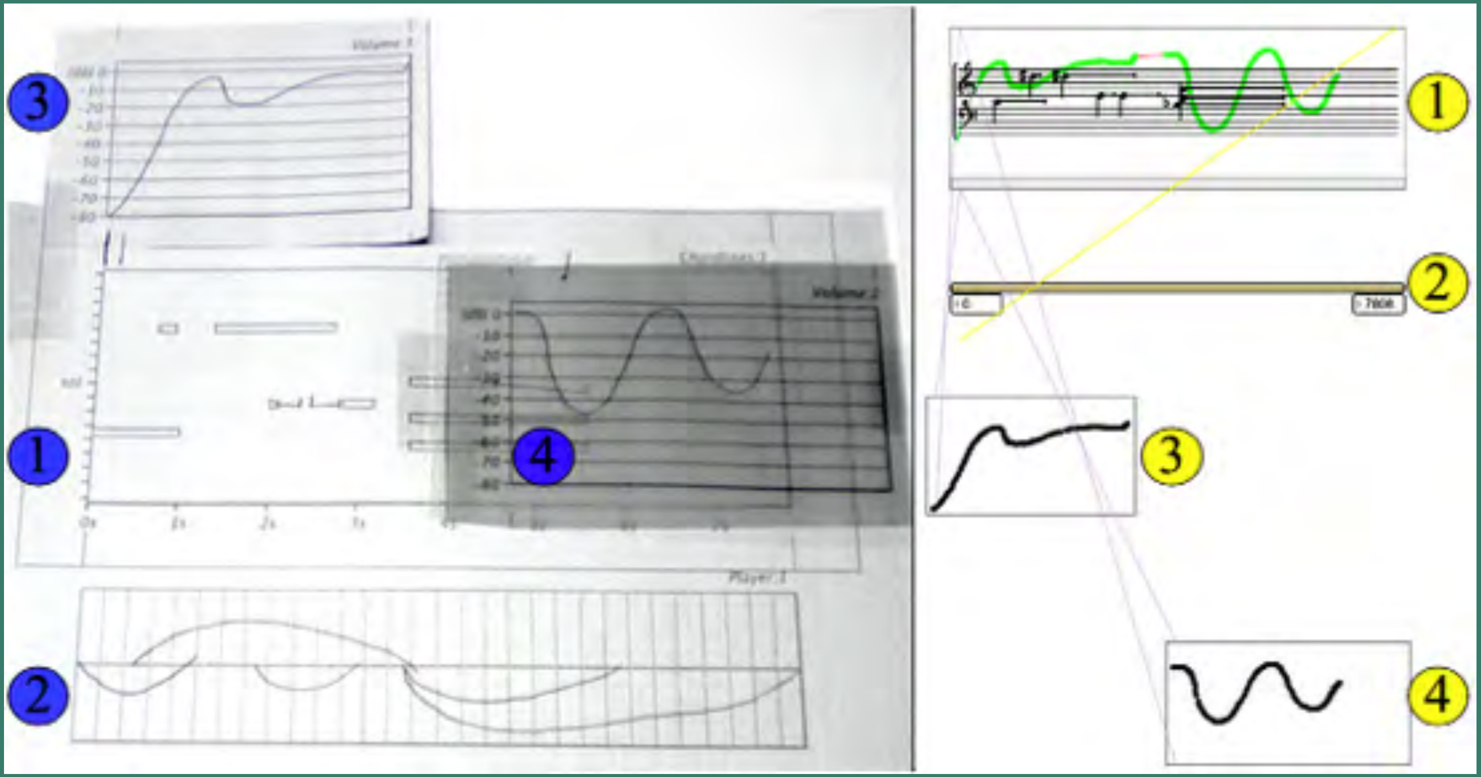


Composers create their own reusable structures

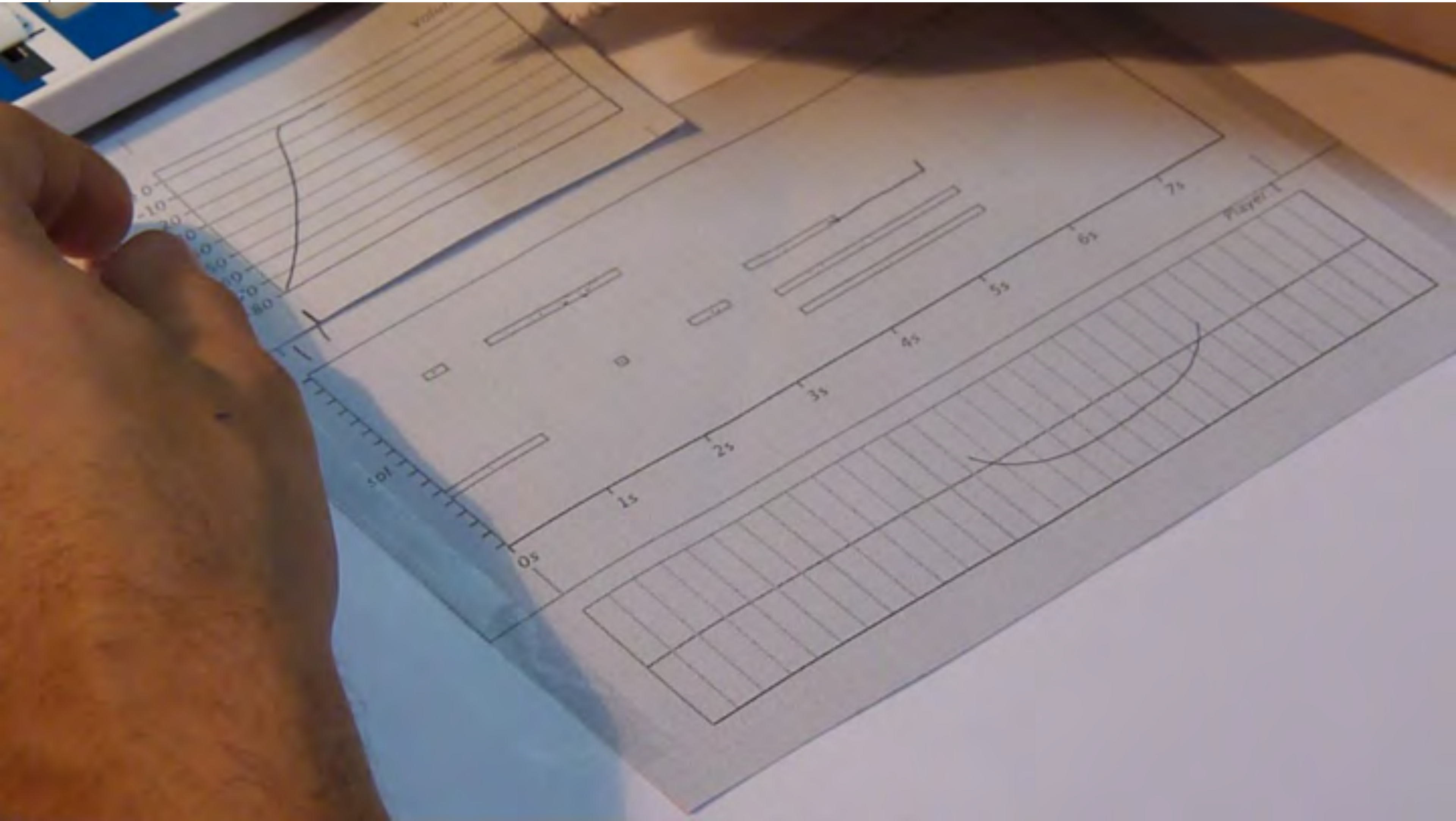


Paper Substrates

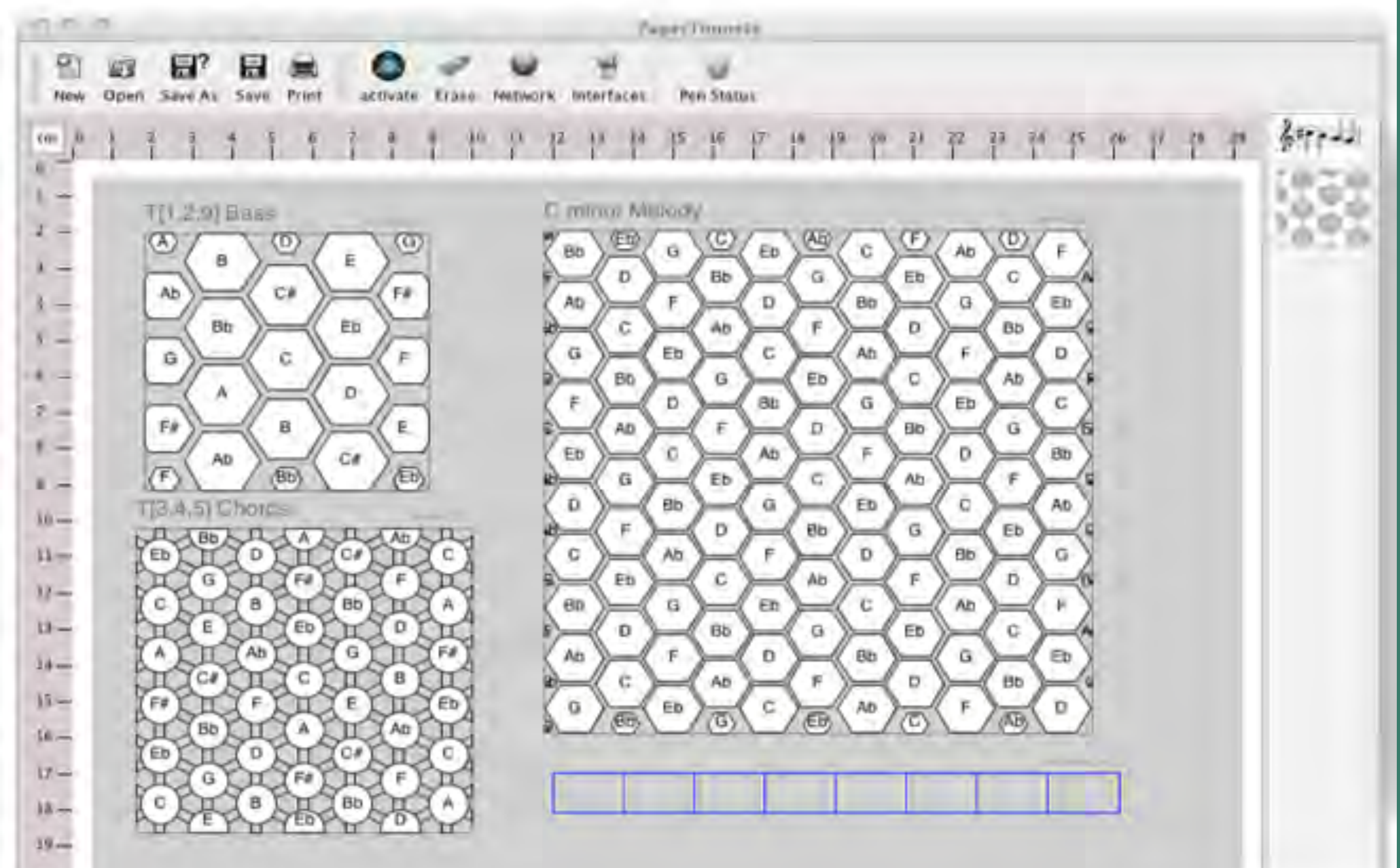
A substrate is both an instrument
for interpreting a personalized language
and an object in its own right



Interactive Paper Substrates for music composers



PaperTonnetz   Draw music based on musical relationships among pitches

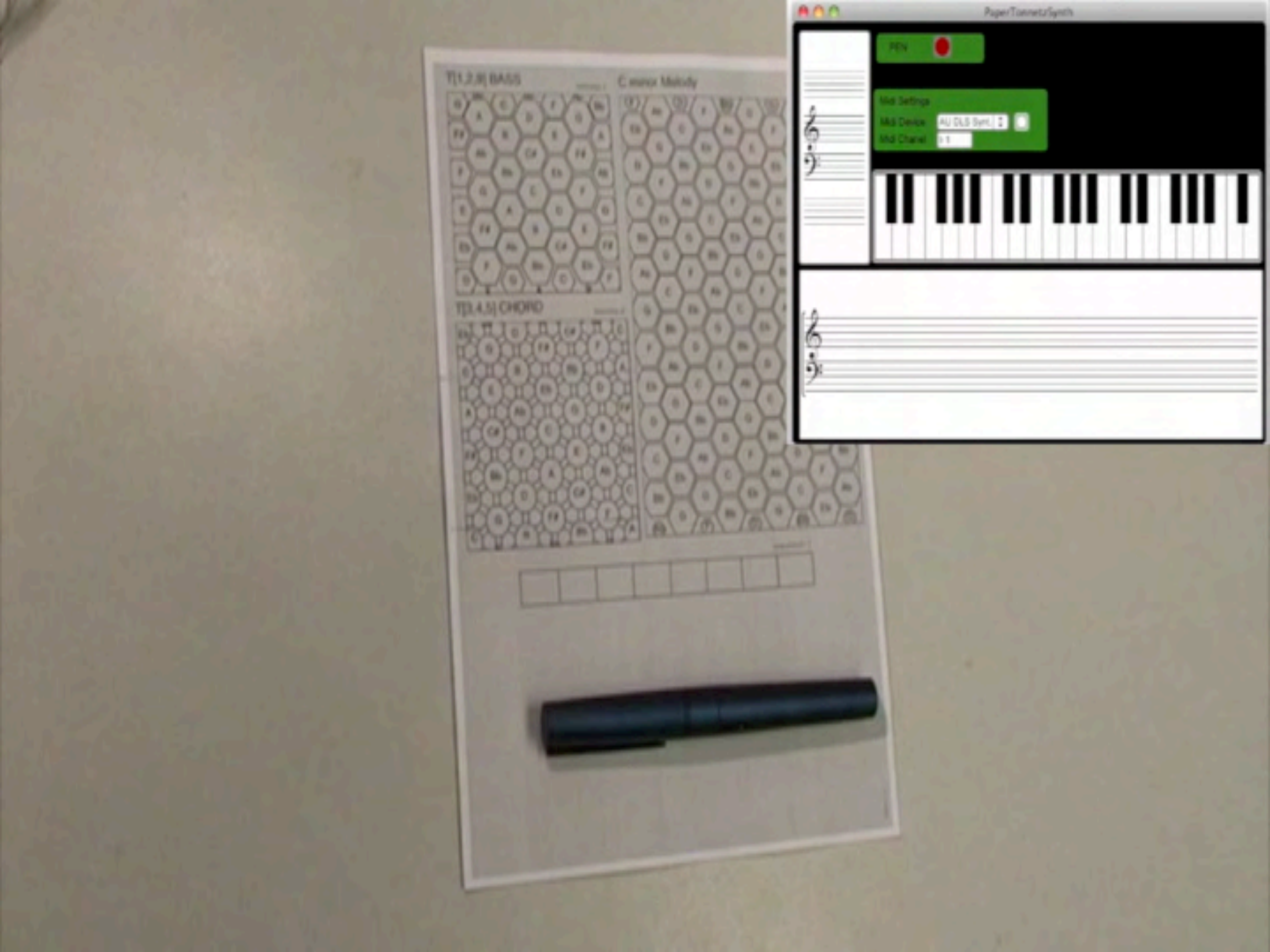


PaperTonnetz

Supporting Music Composition with Interactive Paper

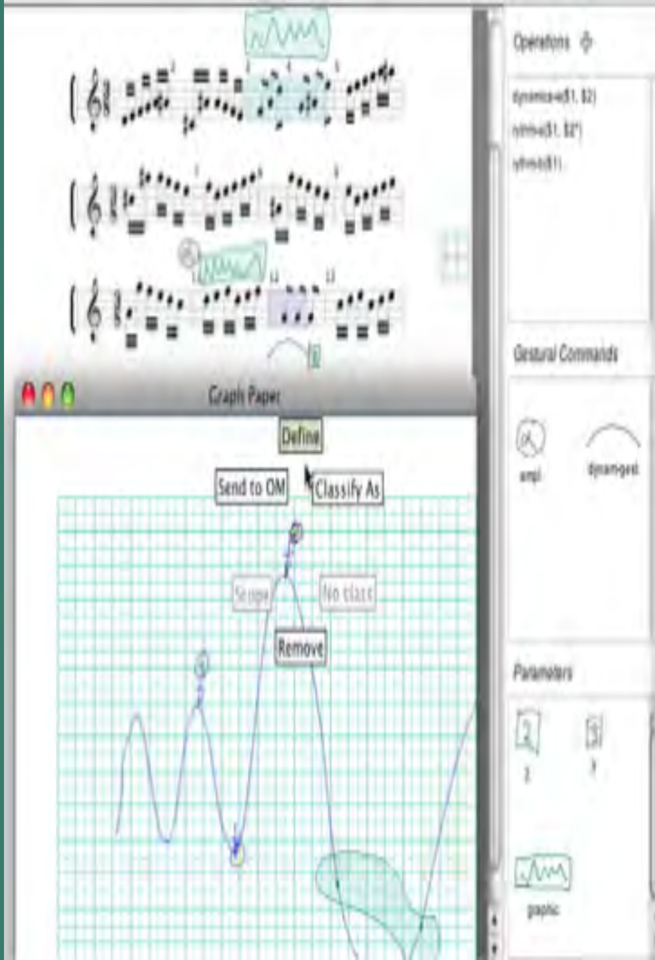
Jérémie Garcia, Louis Bigo, Antoine Spicher and Wendy E. Mackay

INRIA, IRCAM, LACL


T[1,2,9] BASS
C minor Melody
T[3,4,5] CHORD
Paper/Tonnetz/Synth
PEN
Midi Settings
Midi Device
AU DLS Synt
Midi Chanel
1

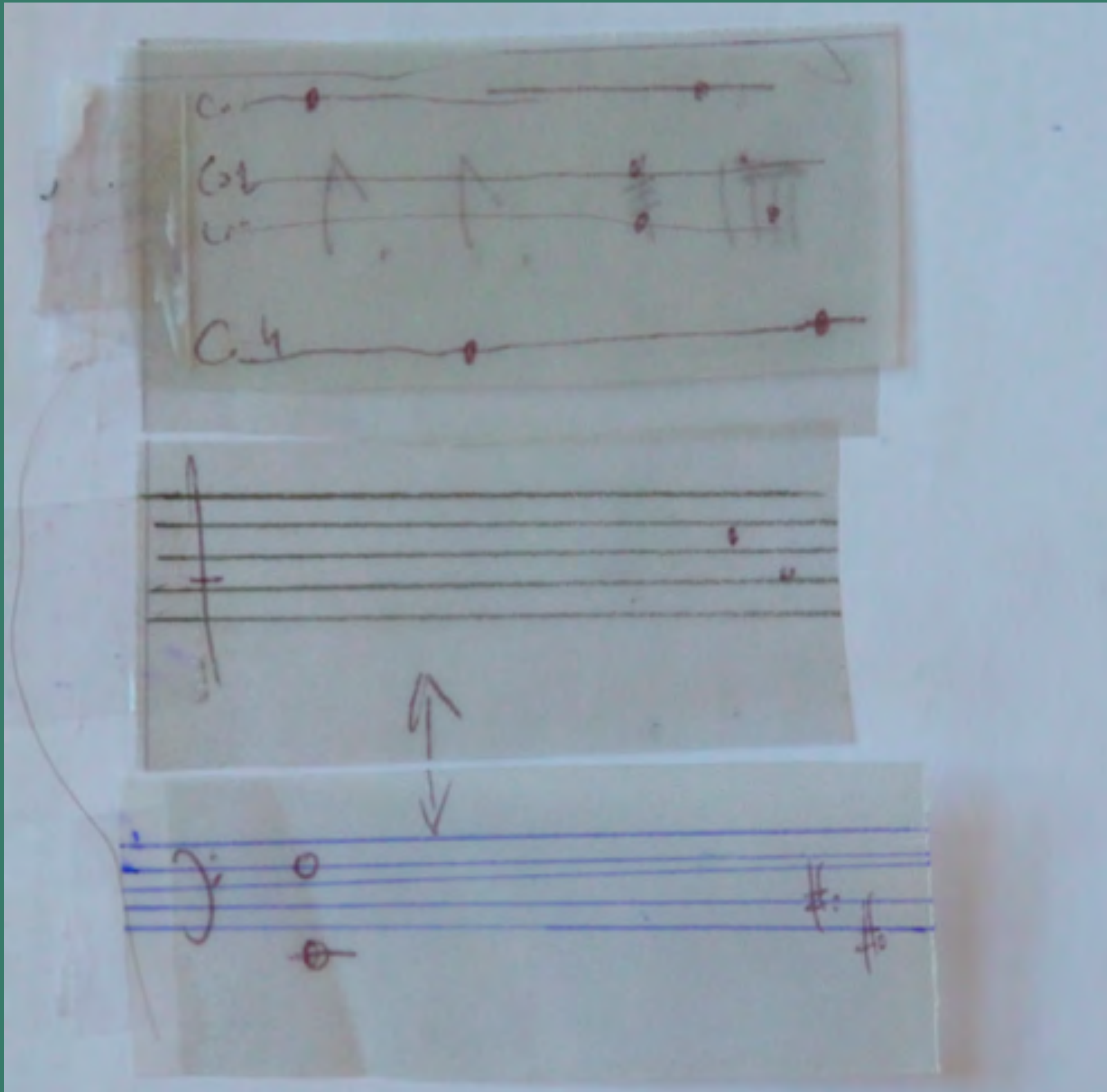
Paper Substrates

Composer create their own
reusable musical structures



establish relationships among them

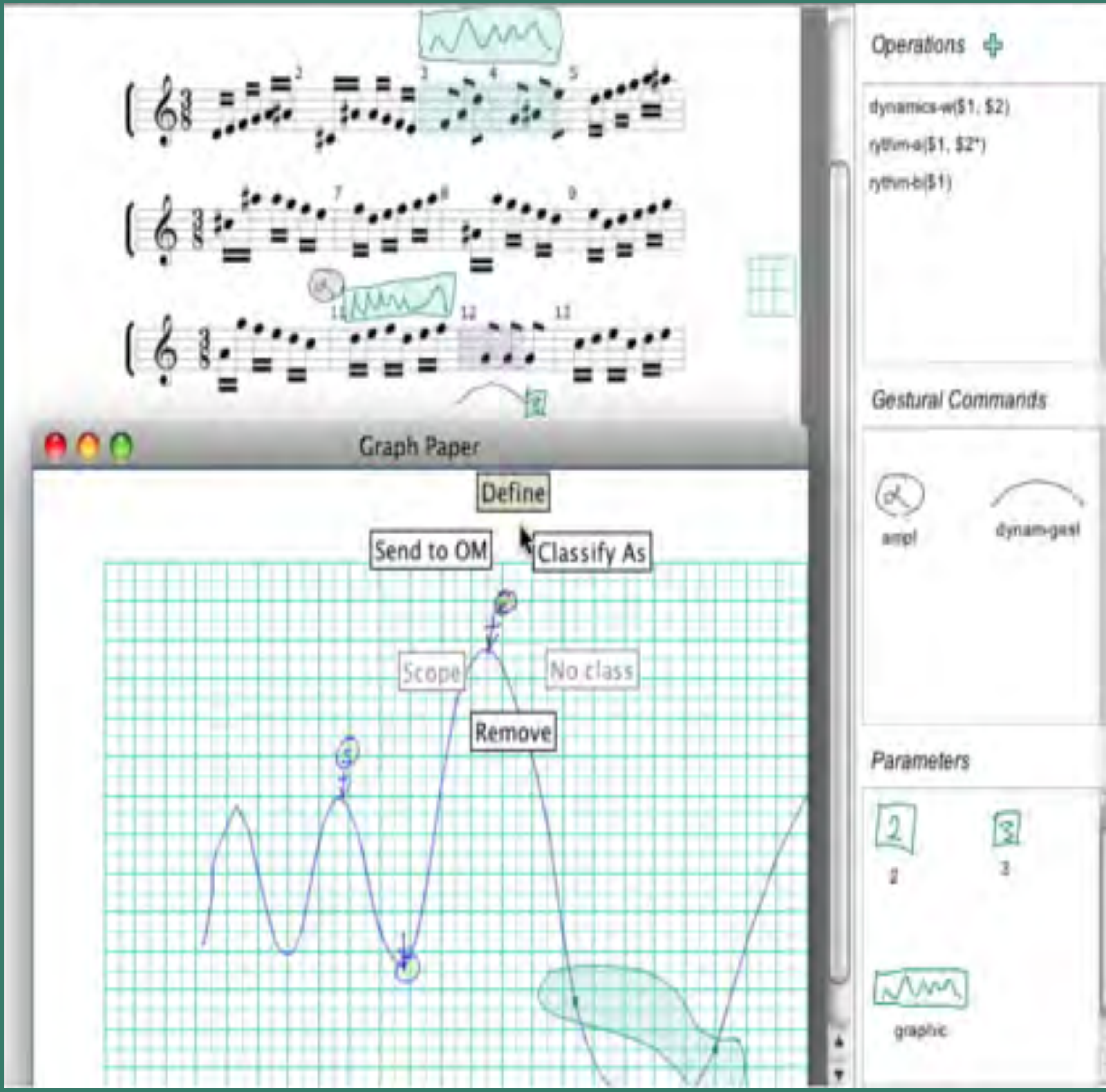
Arrange
and
Link
substrates



Arrange
and
Link
substrates

to

composition
software



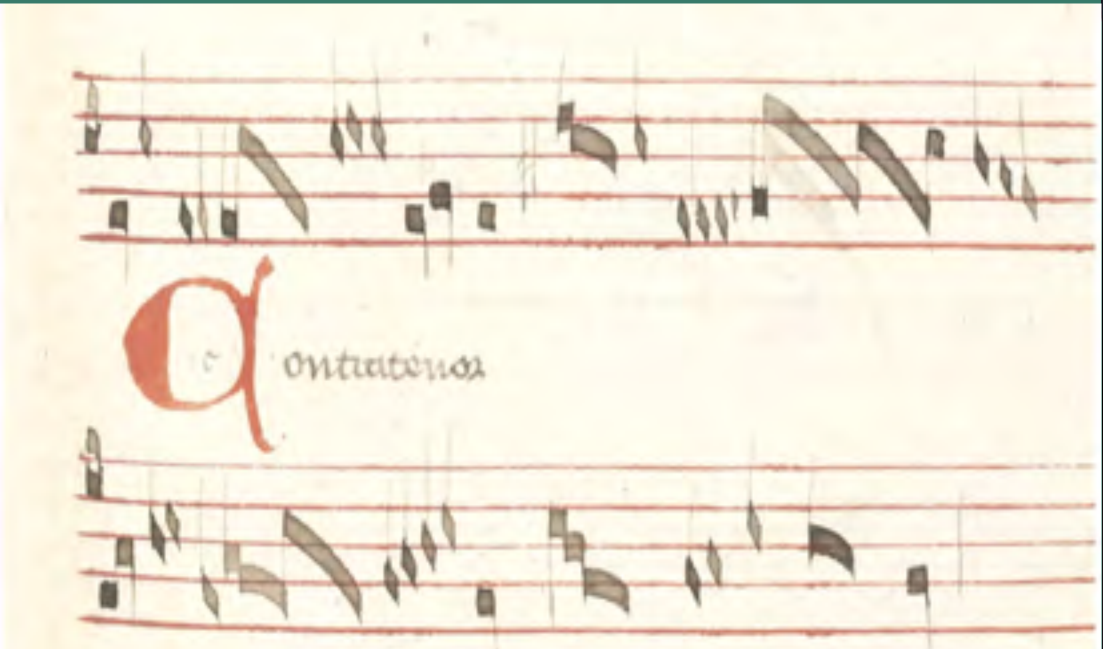
Interactive Paper Substrates
to Support Musical Creation

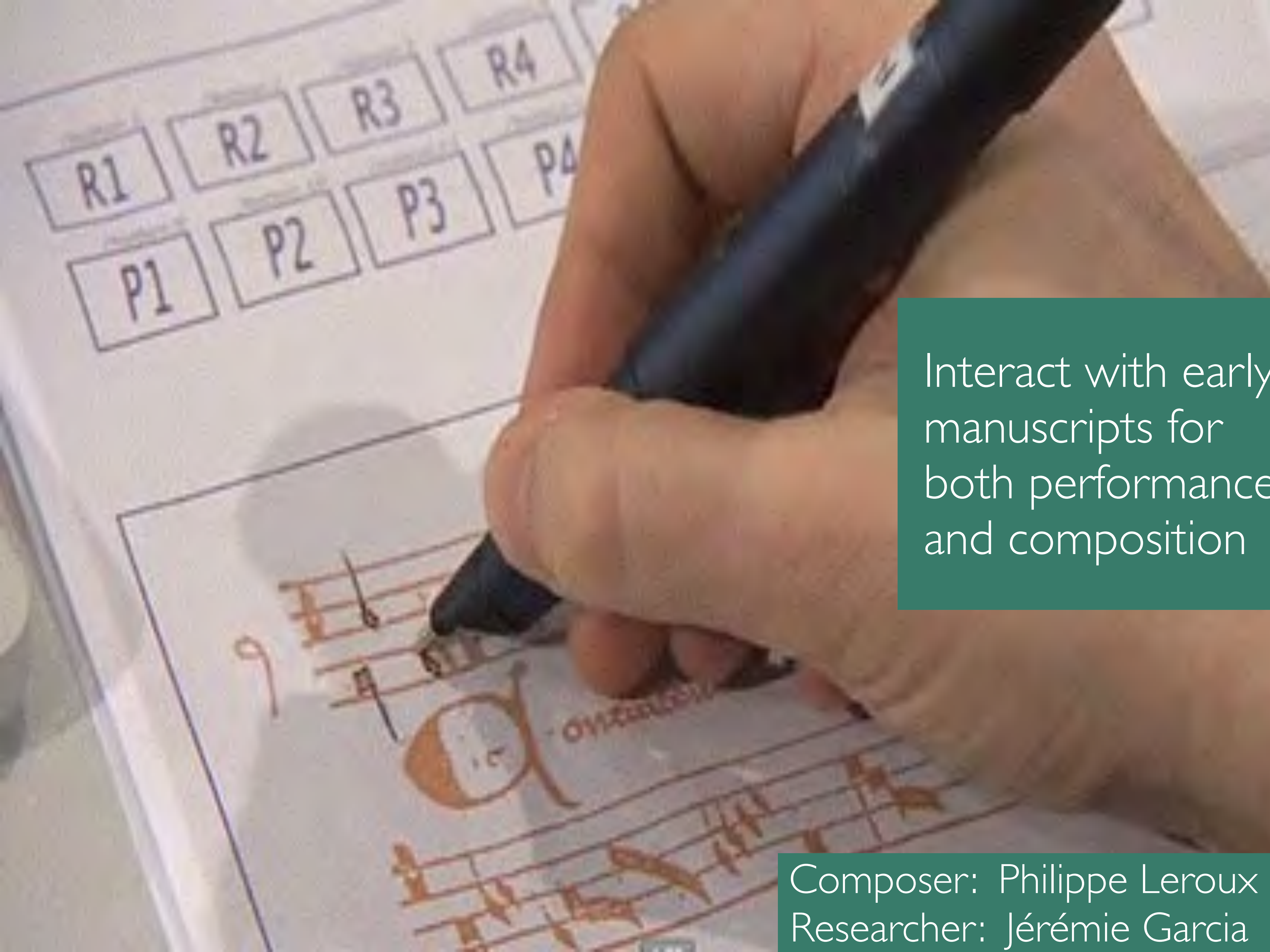
Jérémie Garcia, Theophanis Tsandilas, Carlos Agon & Wendy E. Mackay

INRIA, Université Paris-Sud, CNRS, IRCAM & Stanford University

Quid Sit Musicus
Philippe Leroux

13th century musical scores
Each note indicates expression

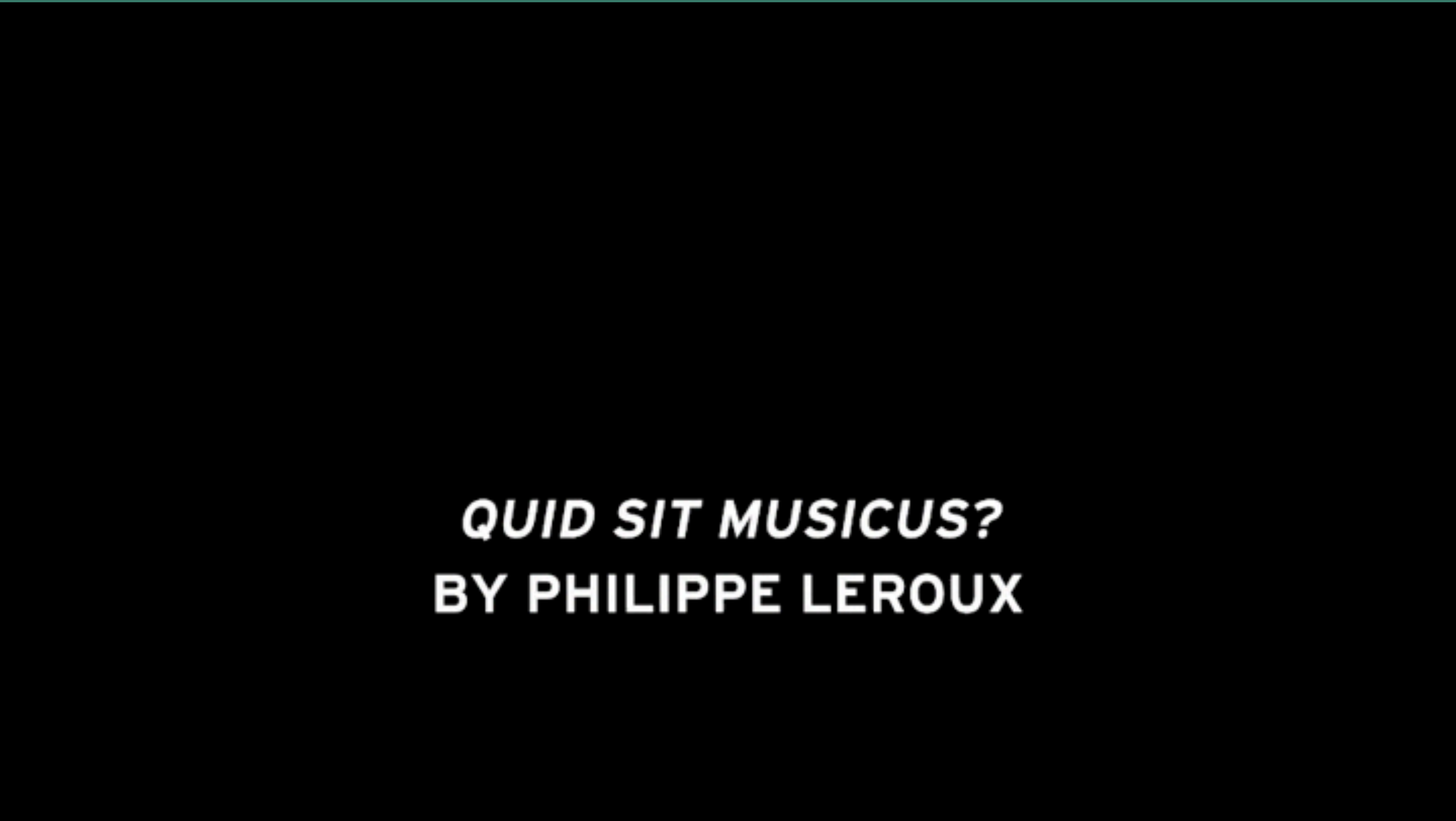




Interact with early manuscripts for both performance and composition

Composer: Philippe Leroux
Researcher: Jérémie Garcia

Quid Sit Musicus
(composer: Philippe Leroux)



QUID SIT MUSICUS?
BY PHILIPPE LEROUX

Reciprocal Co-adaptation

People adapt their behavior to technology
    … they learn it
People adapt the technology for their own purposes
    … they appropriate it

Computers adapt their behavior to people
    … machine learning
Computers adapt human behavior
    … training

What do we mean by 'partnership' ?

Take a taxi

Driver in control

What do we mean by ‘partnership’ ?

Take a taxi
  Driver in control

Drive a motorcycle
  User in control

What do we mean by ‘partnership’ ?

Take a taxi
  Driver in control

Drive a motorcycle
  User in control

Ride a horse
  Shared control

Computer as *tool*

Empower users

Human-Computer Interaction

Computer as *servant*

Delegate tasks

Artificial Intelligence

Computer as *medium*

Communicate

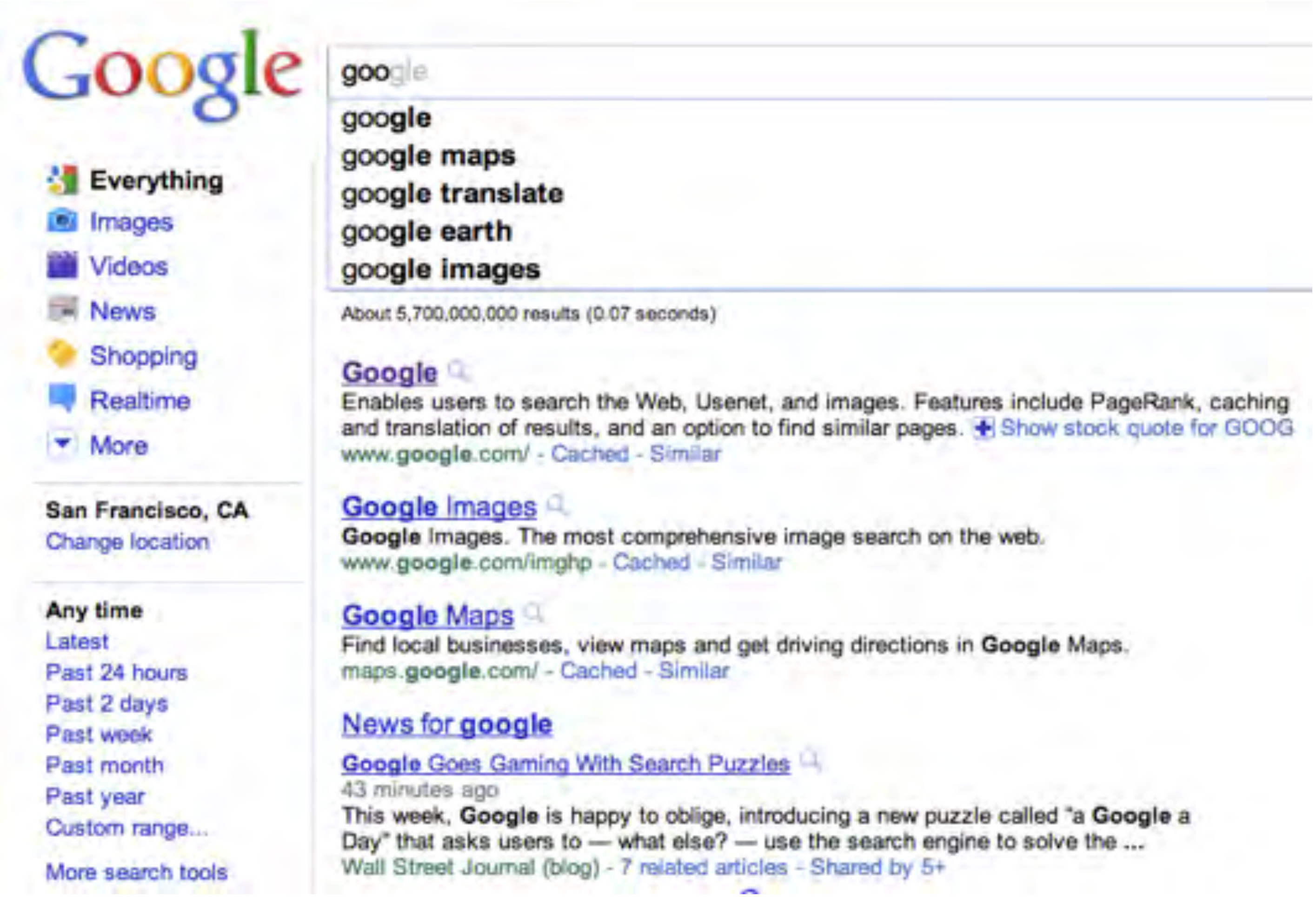
Mediated Communication

Human-Computer Partnerships

Human-in-the-loop

A ‘simple’ human-computer partnership

User types – Google suggests – User chooses



google

google
google maps
google translate
google earth
google images

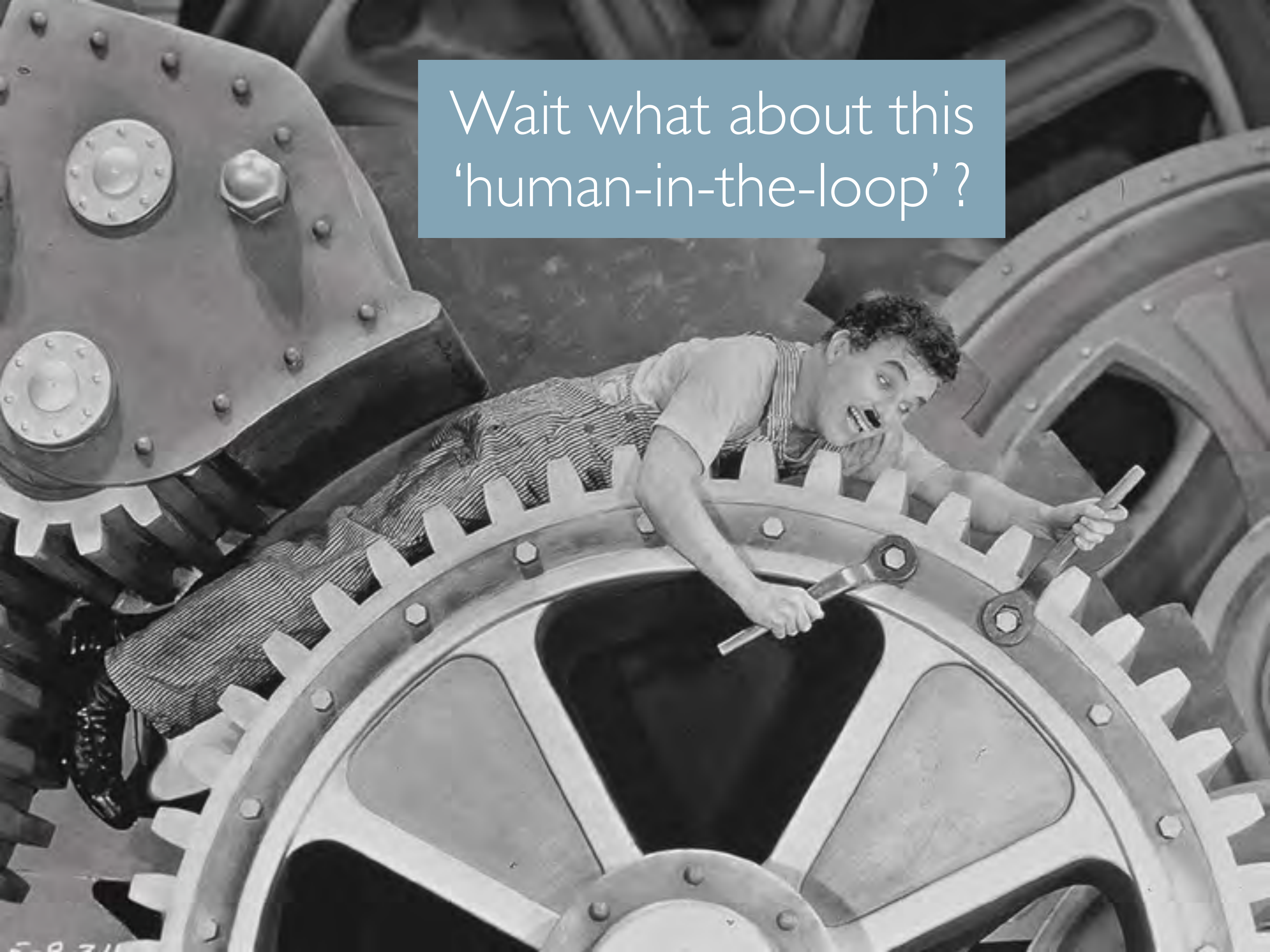
About 5,700,000,000 results (0.07 seconds)

Google
Enables users to search the Web, Usenet, and images. Features include PageRank, caching and translation of results, and an option to find similar pages. + Show stock quote for GOOG
www.google.com/ - Cached - Similar

Google Images
Google Images. The most comprehensive image search on the web.
www.google.com/imghp - Cached - Similar

Google Maps
Find local businesses, view maps and get driving directions in Google Maps.
maps.google.com/ - Cached - Similar

News for google
Google Goes Gaming With Search Puzzles
43 minutes ago
This week, Google is happy to oblige, introducing a new puzzle called "a Google a Day" that asks users to — what else? — use the search engine to solve the ...
Wall Street Journal (blog) - 7 related articles - Shared by 5+



Wait what about this ‘human-in-the-loop’ ?

Human-Computer Partnerships

Human-in-the-loop

Machine learning perspective:

Human helps *improve the algorithm*

Human-Computer Partnerships

Human-in-the-loop

Machine learning perspective:

Human helps *improve the algorithm*

Computer-in-the-loop

How can we empower people ?

Human-Computer Partnerships

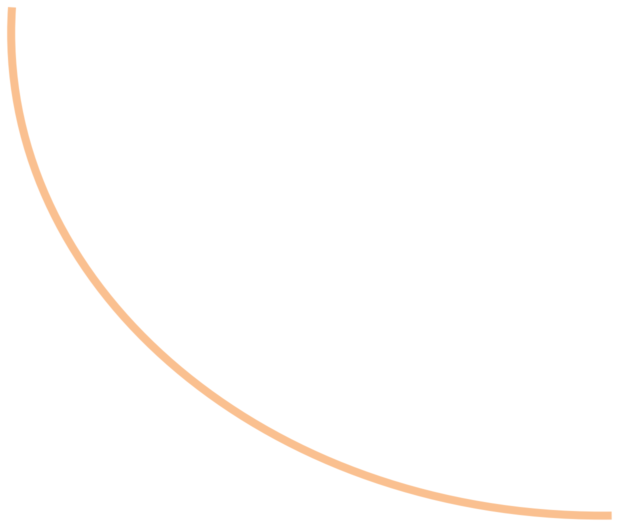
Human-in-the-loop

Human helps *improve the algorithm*

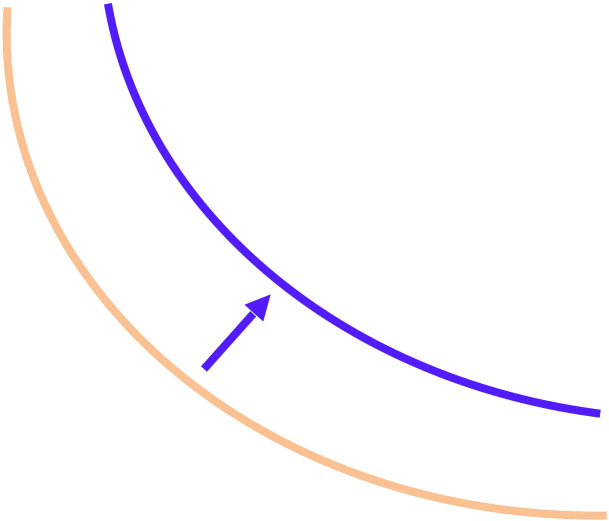
Computer-in-the-loop

Human-computer interaction perspective:
Computers *empower the human user*

We face a major design trade-off



Solution: Shift the curve



Unified principles of interaction

Two complementary perspectives:

System: How to build it ?

Instrumental Interaction
and Substrates

with Michel Beaudouin-Lafon

Unified principles of interaction

Two complementary perspectives:

System: How to build it ?

Instrumental Interaction

and Substrates

Human: How to interact with it ?

Co-adaptive Systems

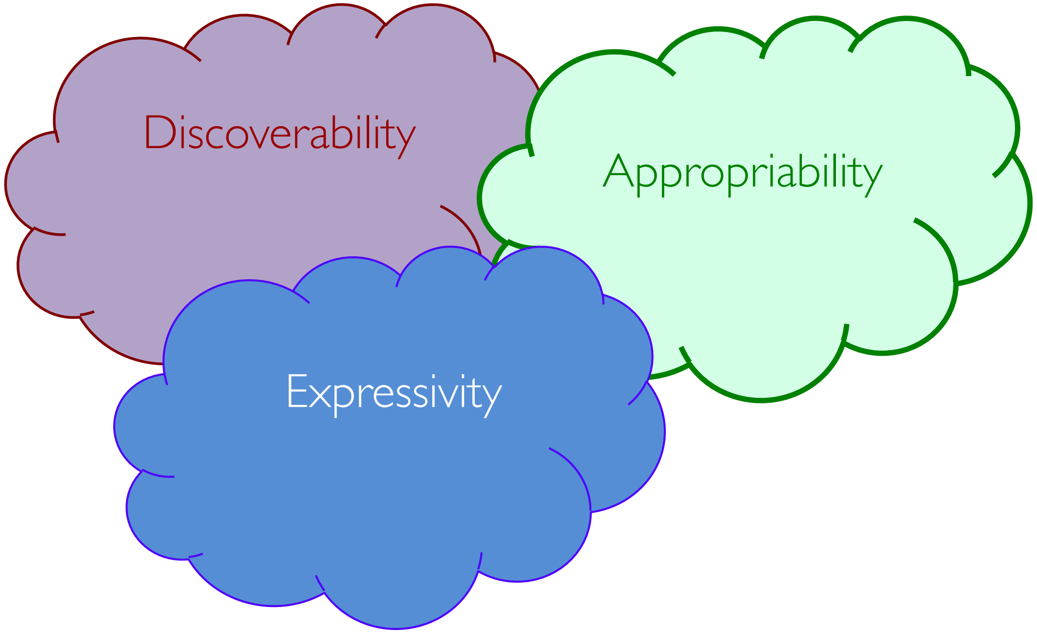
Human-computer partnerships

Human-computer partnerships

People can

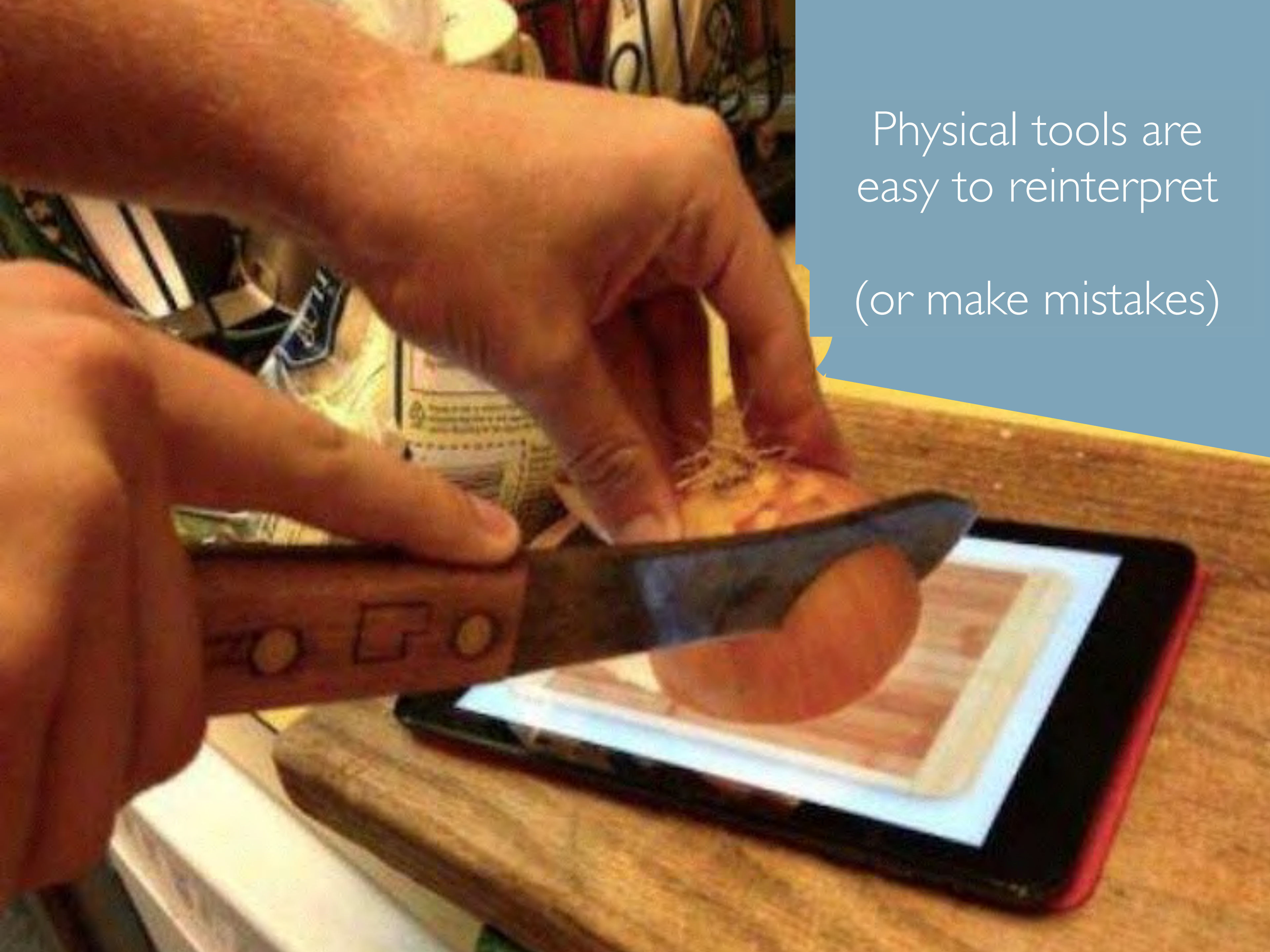
adapt to technology — they learn it

adapt the technology — they appropriate it



What can we learn from physical tools ?

What can we learn from physical tools ?



Physical tools are easy to reinterpret

(or make mistakes)

We can appropriate physical tools

We can appropriate physical tools

We can appropriate physical tools
  … why not software ?

Why do software applications choose for us ?

Imagine needing the manufacturer’s stick to hit a drum !

We define physical tools through use

Musical instruments
become part of
the body

Why can’t we learn to ‘play’ software tools ?
without relearning the interface
with every software upgrade ?

Computer as *tool*

Empower users

Computer as *servant*

Delegate tasks

Computer as *medium*

Communicate

Human-Computer Interaction

Artificial Intelligence

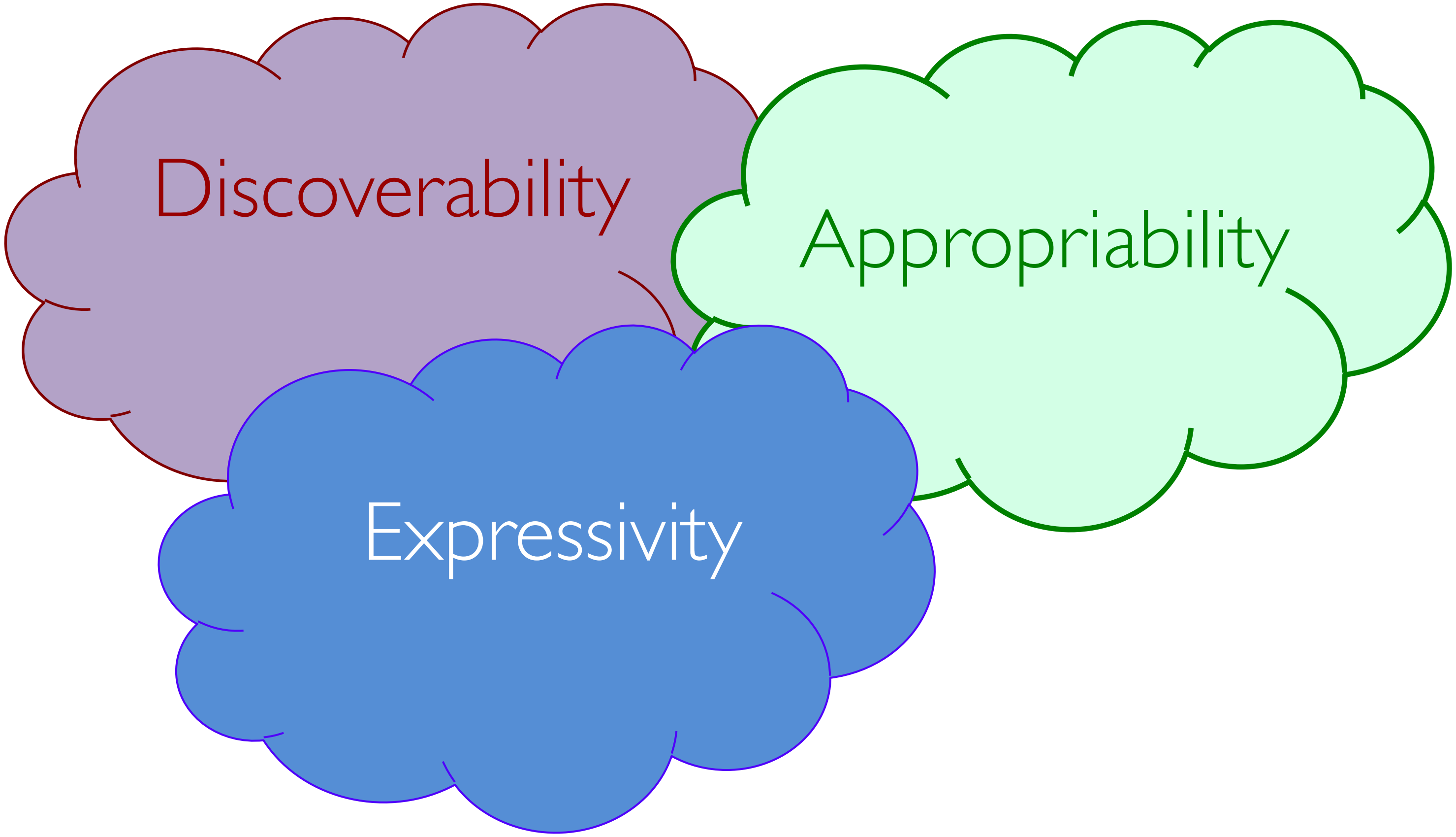
Mediated Communication

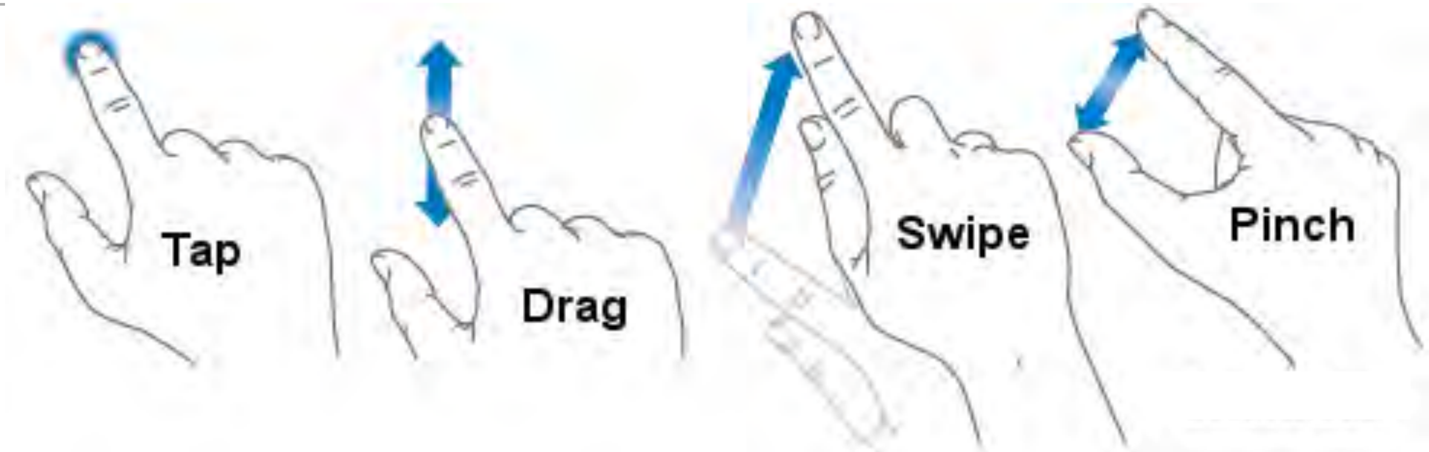
Human-computer partnerships

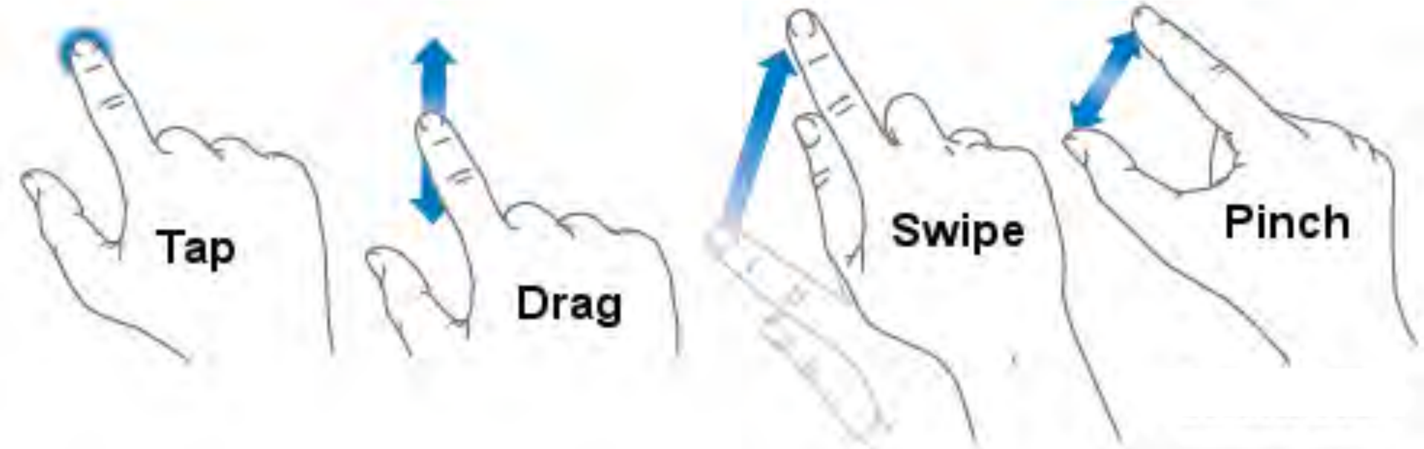
People can	
adapt to technology	they learn it
adapt the technology	they appropriate it
Computers can	
adapt to people	they learn (AI)
adapt people’s behavior	they teach

Human-Computer Partnerships

People seek:




Tap
Drag
Swipe
Pinch


Tap
Drag
Swipe
Pinch

Discoverability

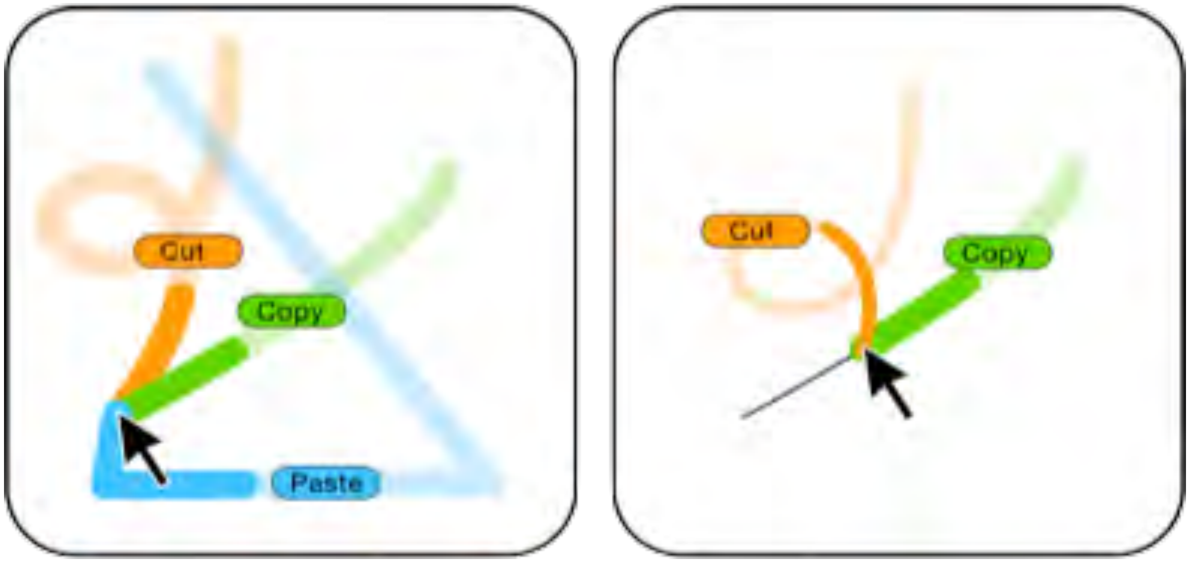
How can I learn
new gestures
and commands ?

Octopocus

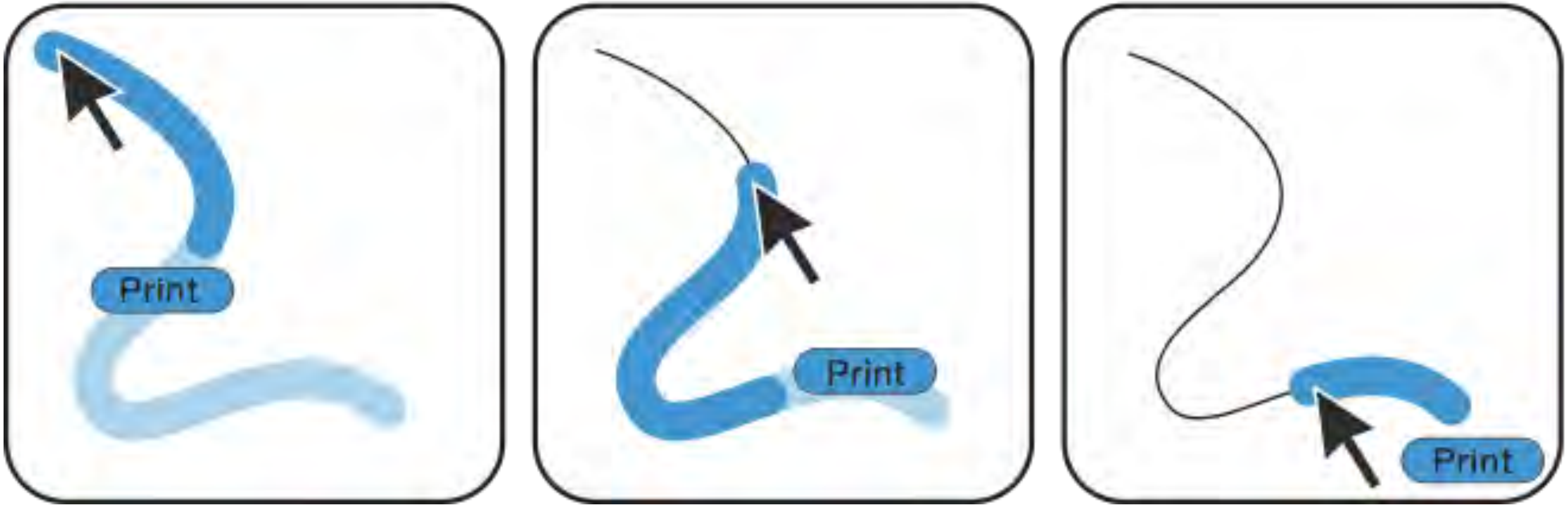
- Learn gestures that issue commands
  - Progressive feedforward
    - *What gestures are available ?*
  - Progressive feedback
    - *What did the system recognize ?*

Novices pause ...
the guide appears

Experts just do it



Dynamic partnership



Inking the '*Help*' command

Appropriability

How can I define
my own gestures ?

Fieldward

CHI '17

Create personal gesture commands

Choose easy-to-remember gestures

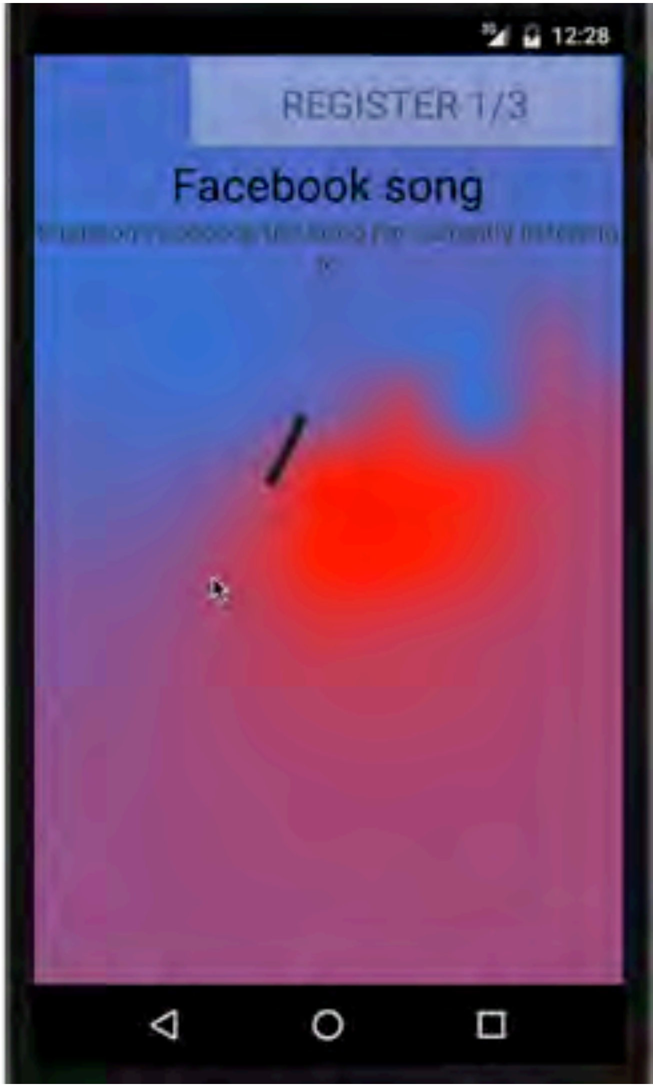
Progressive feedforward reveals whether

- command exists
- it is recognizable

Fieldward: create personal gestures

Fieldward

Shows a color gradient
indicating optimal directions
to make a recognizable gesture



Expressivity

How can I
express myself ?

Expressive Keyboard

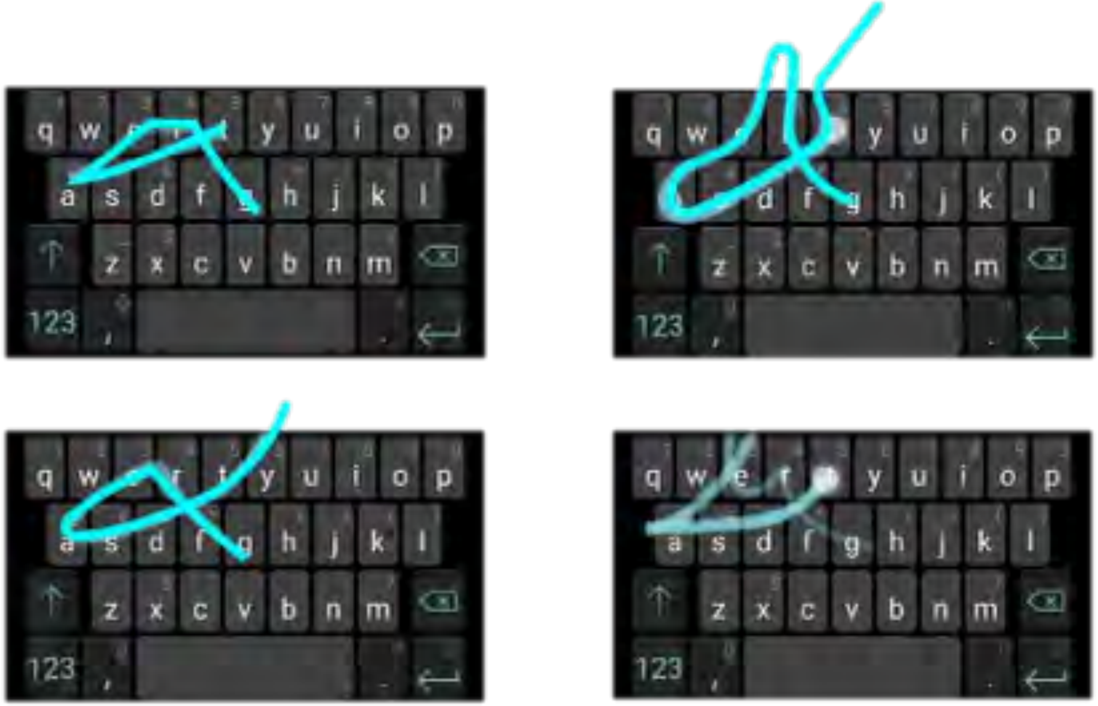
CHI'16

Redefining gesture-typing keyboards

to support user expression

Four ways to type the word "great"

each produces the same result



Machine learning guesses the correct word

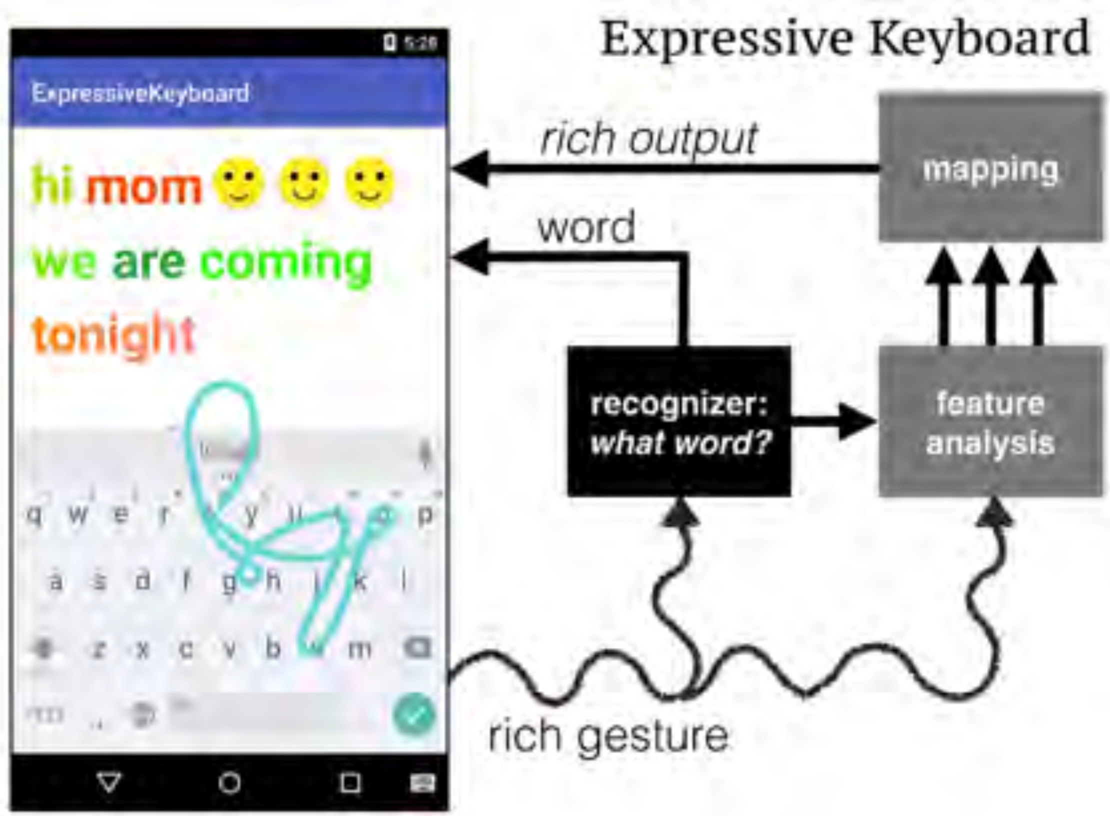
Gesture variations creates expressive text

Users control

text color

font style

and emojis

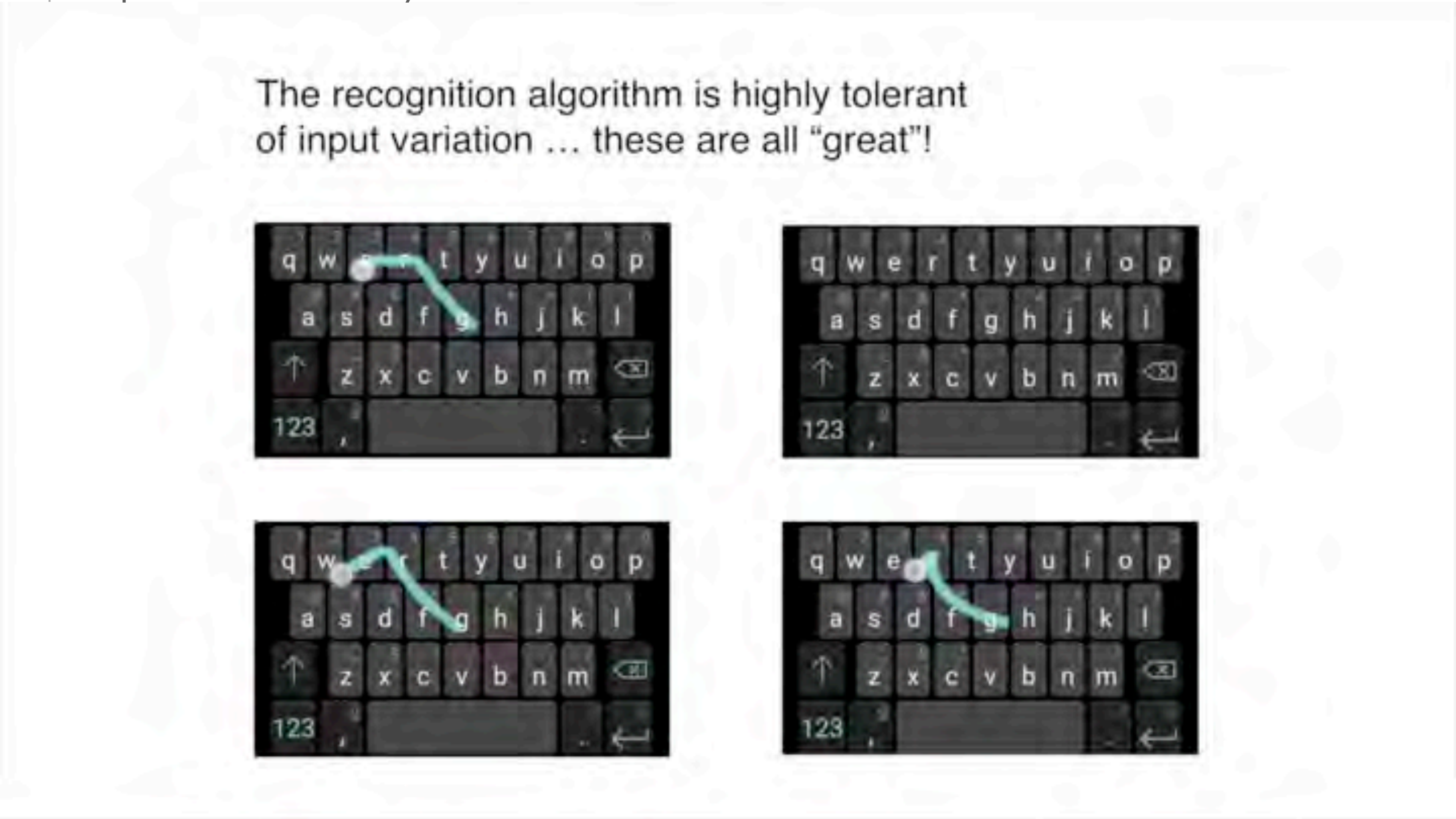


hi mom 🙂 🙂 🙂

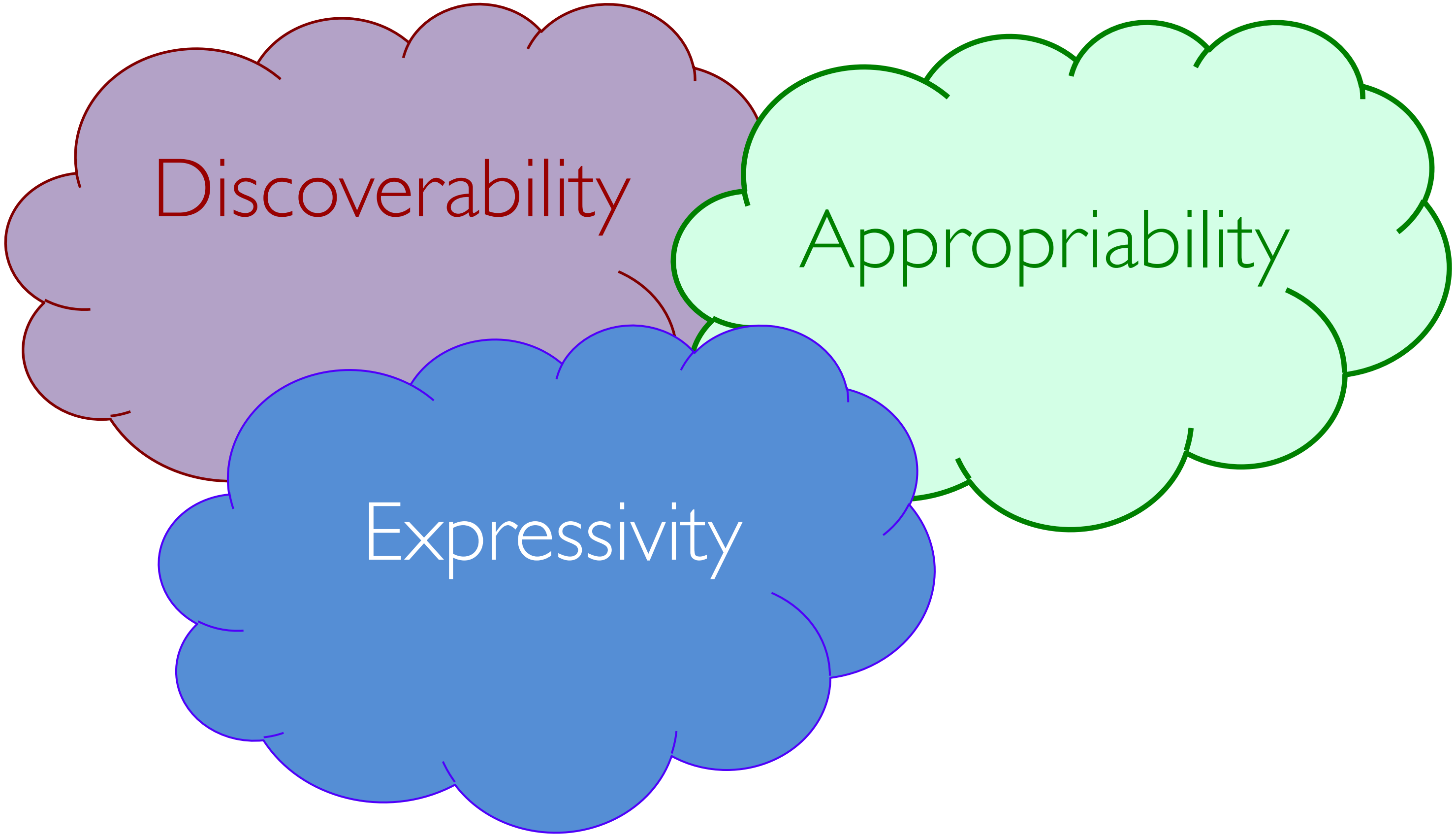
we are coming

tonight

Expressive Keyboard



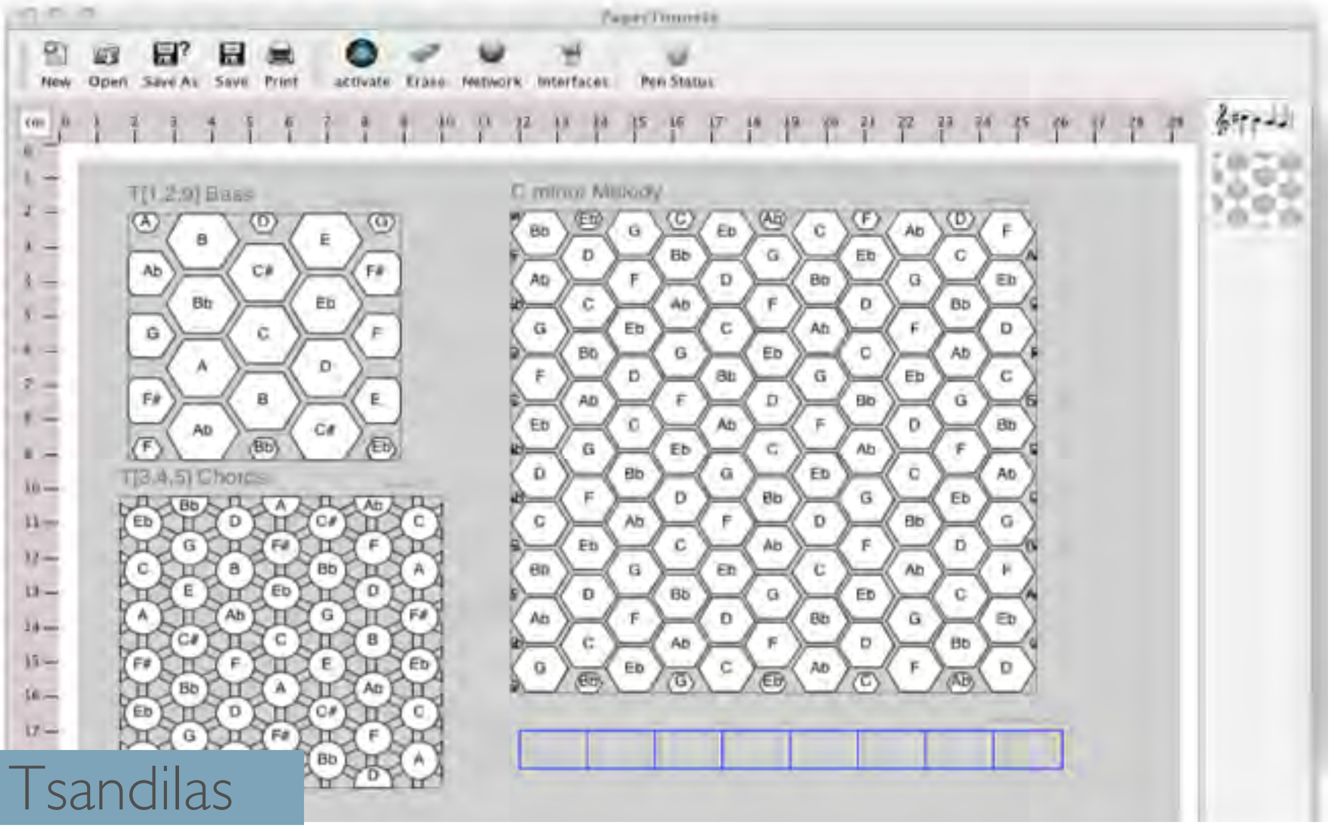
The recognition algorithm is highly tolerant of input variation … these are all “great”!


Discoverability
Appropriability
Expressivity

Discoverability

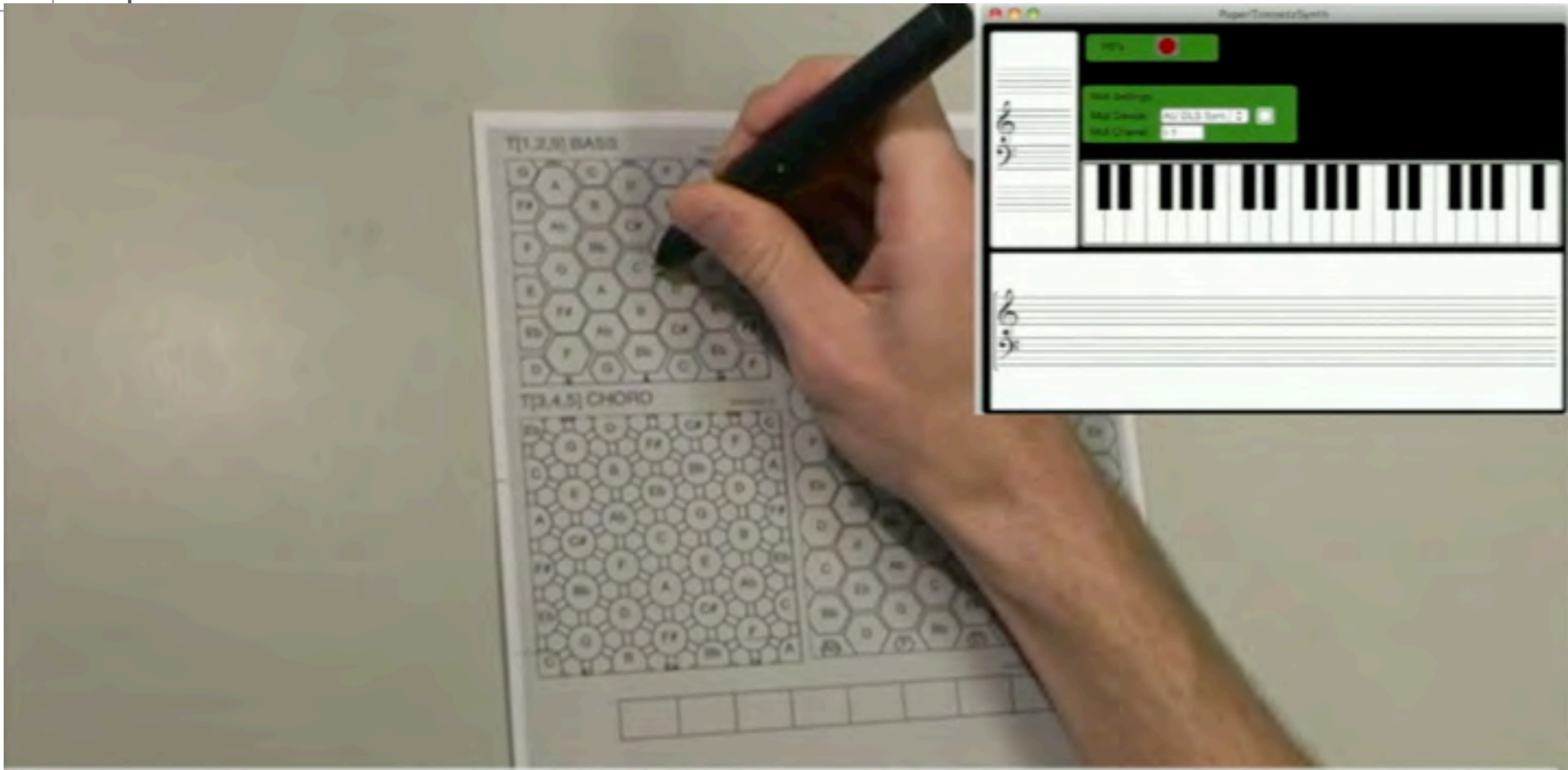
How can I learn
musical relationships ?

Paper Tonnetz



Jérémie Garcia, Fanis Tsandilas

Paper Tonnetz



The composer starts with the bass part

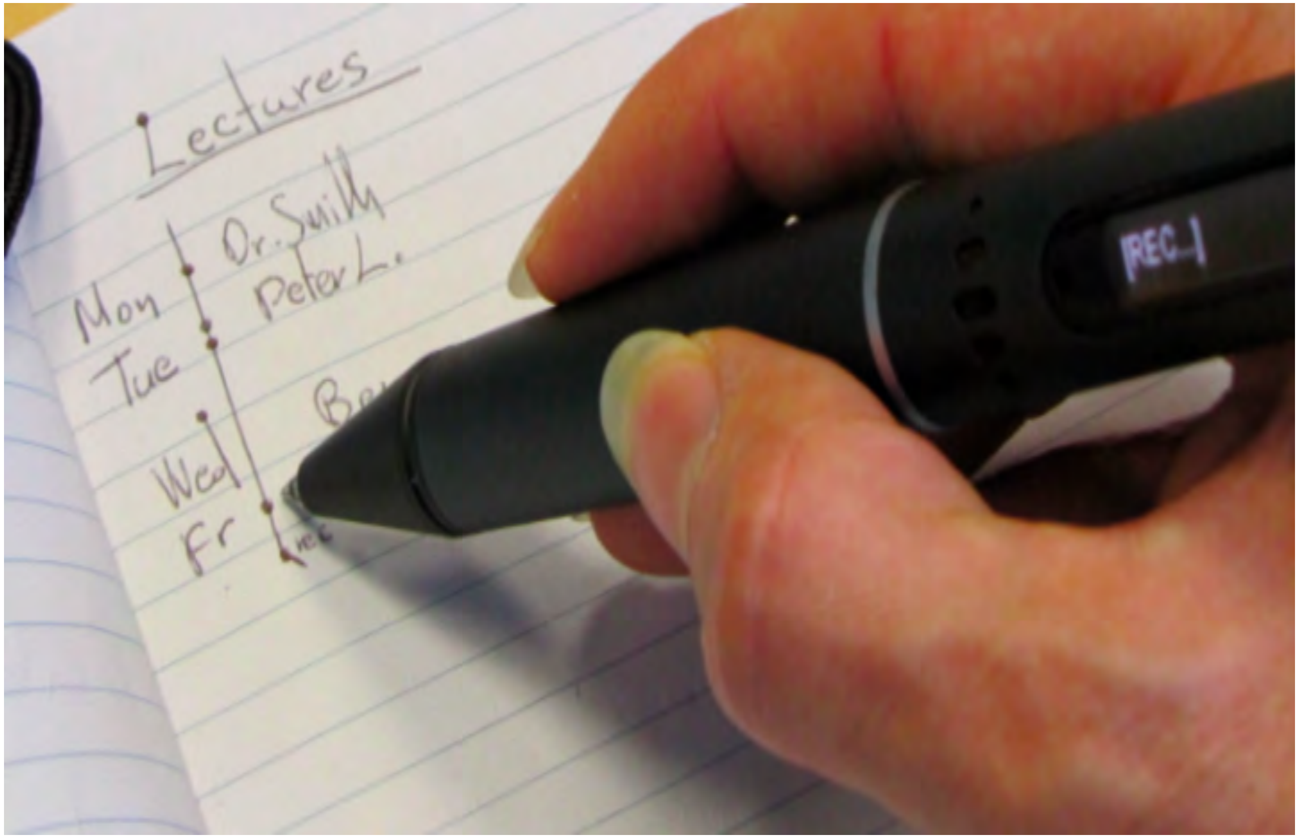
Appropriability

How can I define
my own commands ?

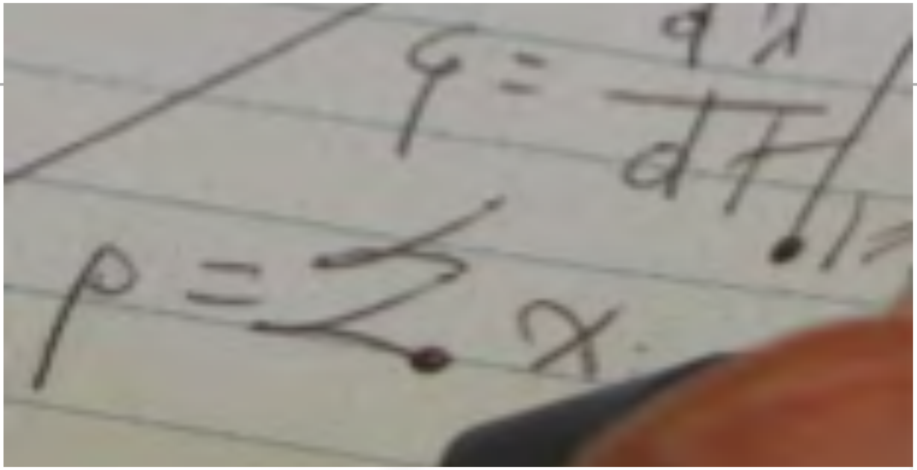
Knotty Gestures:

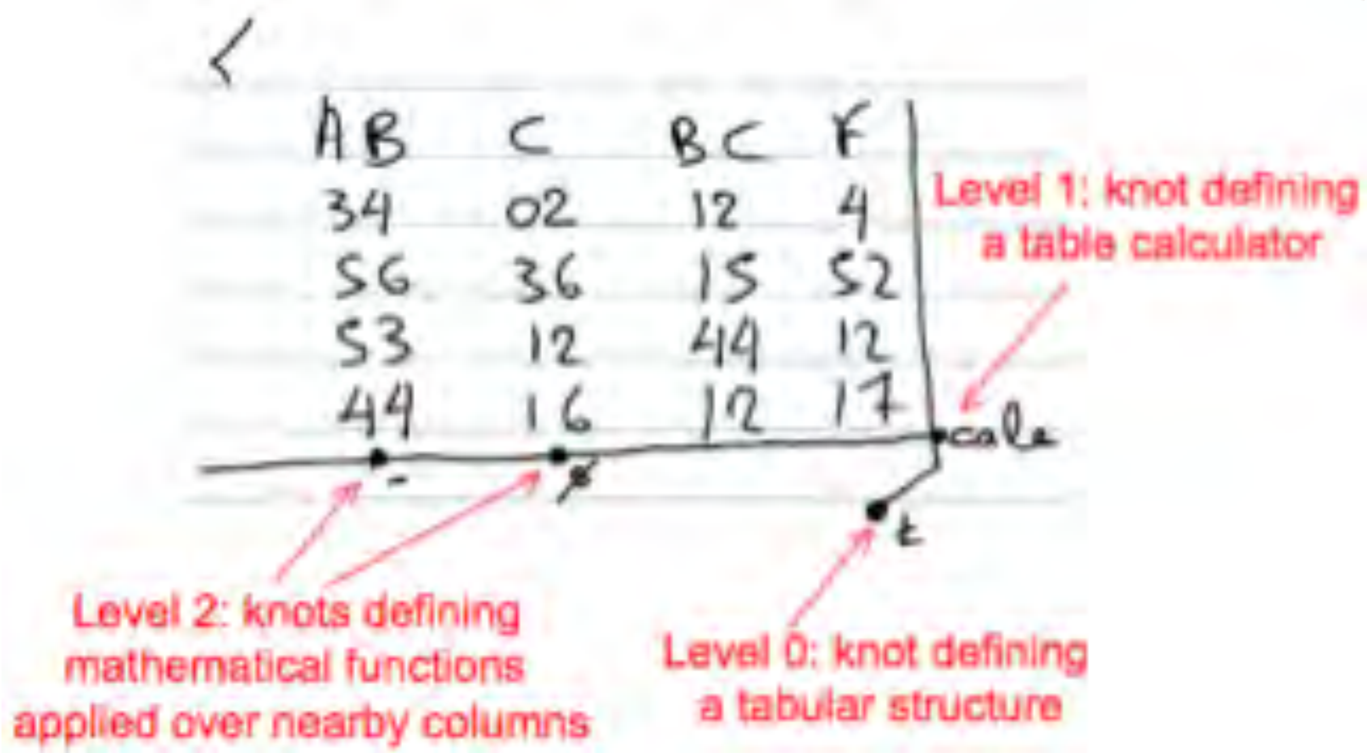
AVI'10

Draw a knot to define a command
Interact now or later



Knots can define mathematical or other relationships





AB	C	BC	F
34	02	12	4
56	36	15	52
53	12	44	12
44	16	12	17

Level 1: knot defining a table calculator

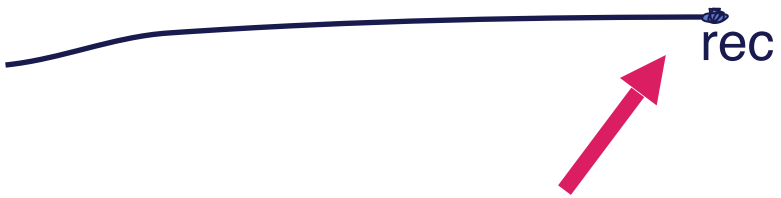
Level 2: knots defining mathematical functions applied over nearby columns

Level 0: knot defining a tabular structure

Knotty Gestures

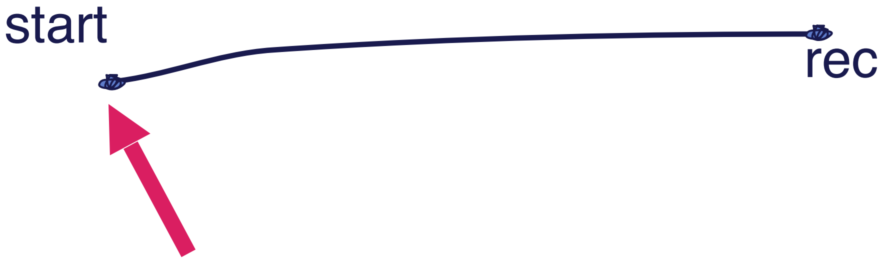
Draw a line with a knot

Choose "recording" to define the type of line



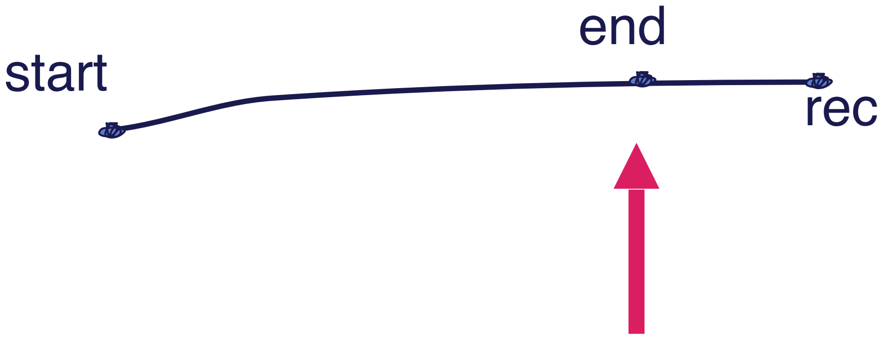
Knotty Gestures

Add another knot to define
the start of the recording



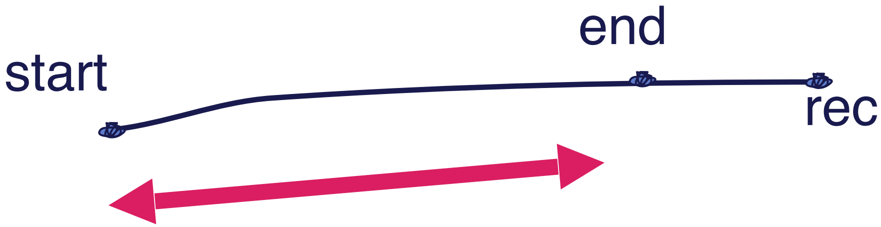
Knotty Gestures

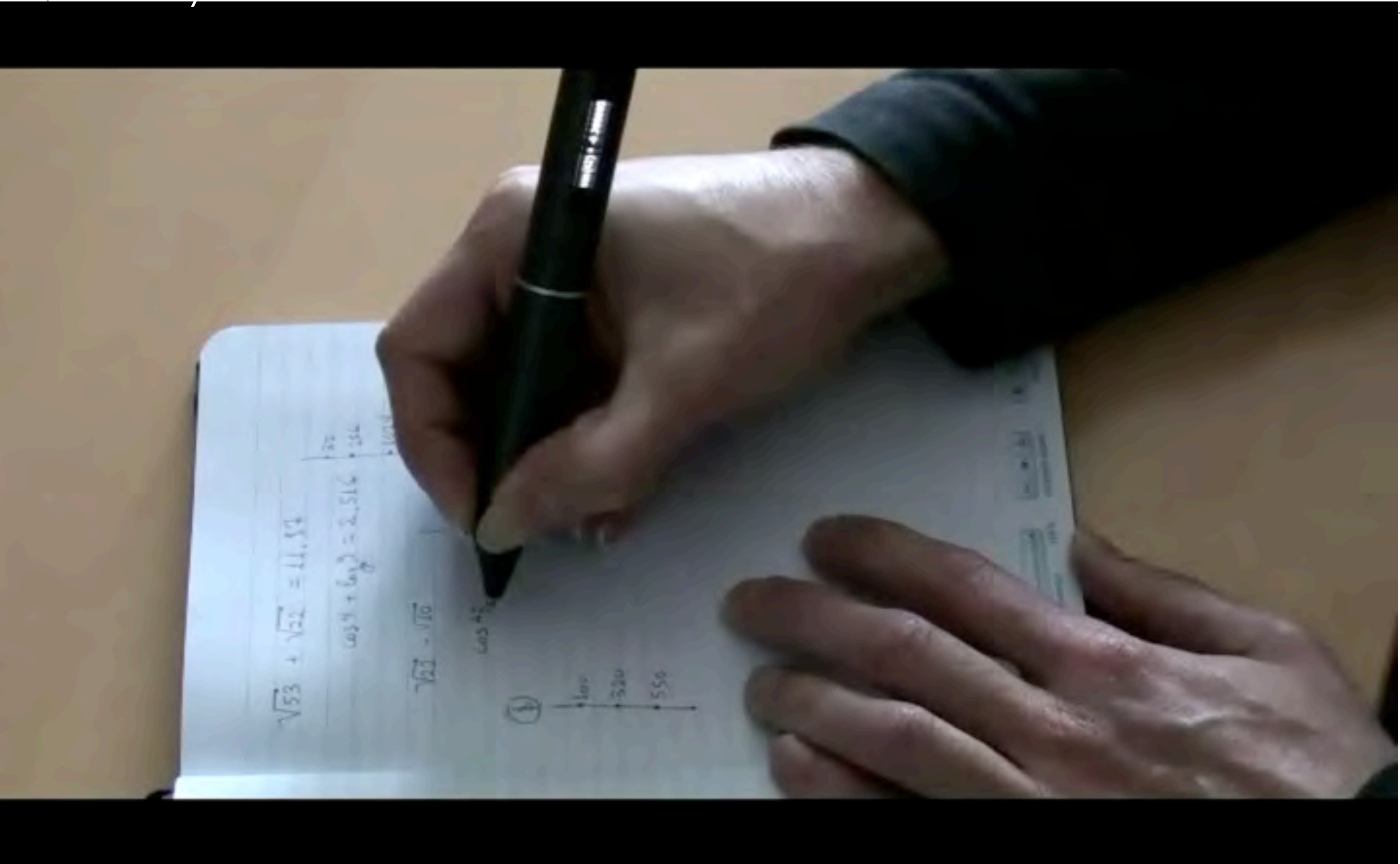
Add a third knot to define the end of the recording



Knotty Gestures

Slide the pen back and forth
to play the recording



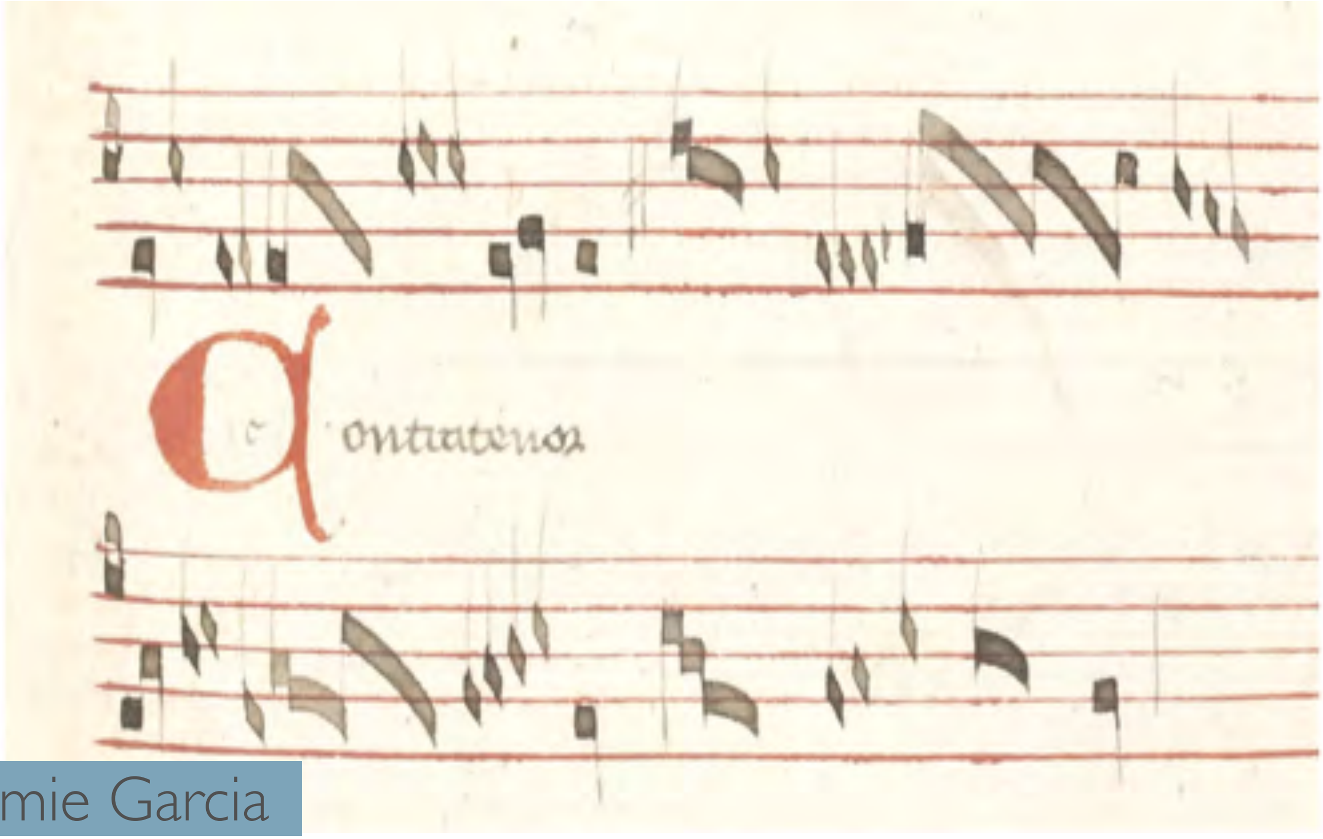

√53 + √52 = 11.57
500
530
550

Expressivity

How can I
express myself ?

13th century musical scores

Each note indicates expression



Philippe Leroux, Jérémie Garcia

Paper Composer

Associate gesture characteristics
with sounds and features

compose
& perform

Jérémie Garcia

QUID SIT MUSICUS?

BY PHILIPPE LEROUX