

# 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

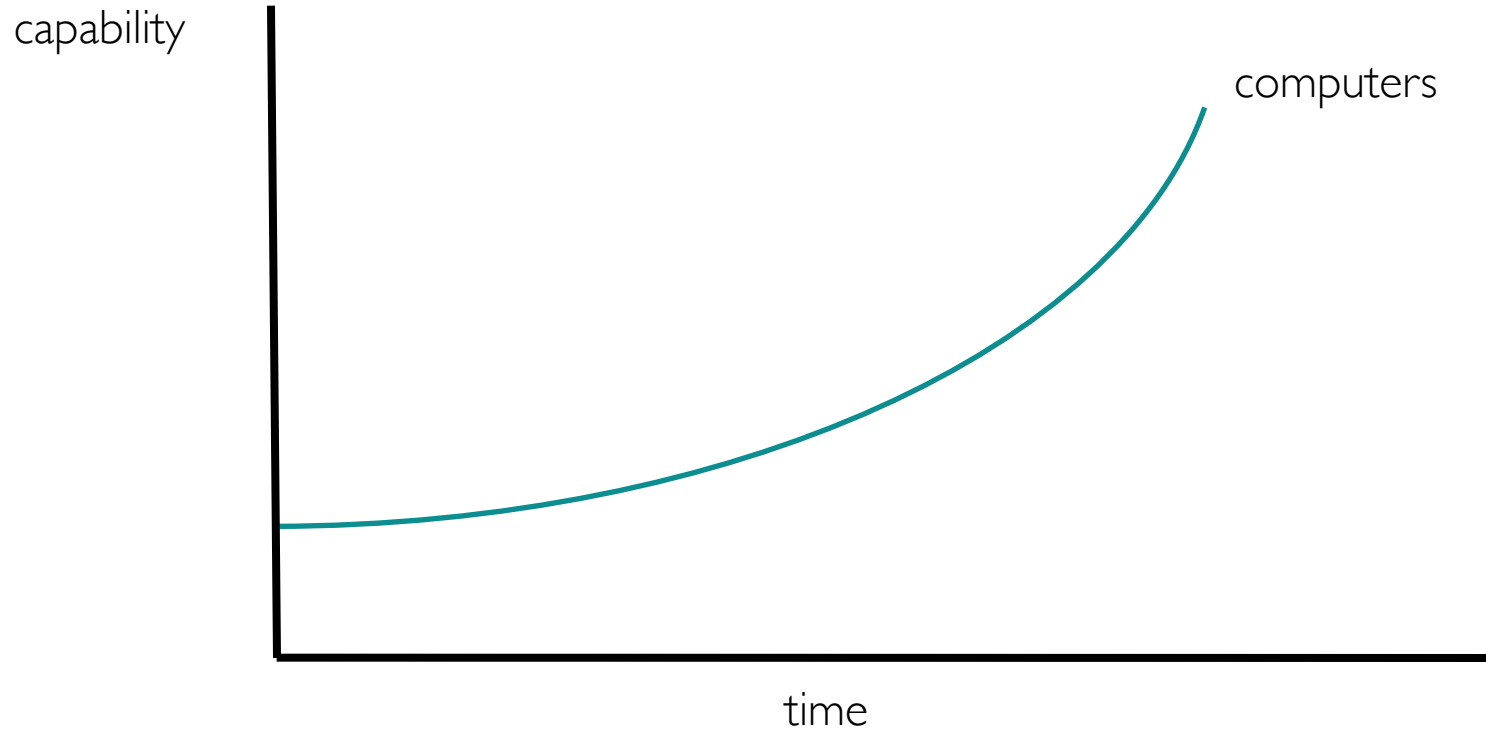
Suggest ideas for revising the techniques  
or creating a new co-adaptive instrument

Human-Computer  
Partnerships  
*or*  
Co-Adaptive Instruments

Computer hardware  
has changed dramatically over the past  
40 years ...

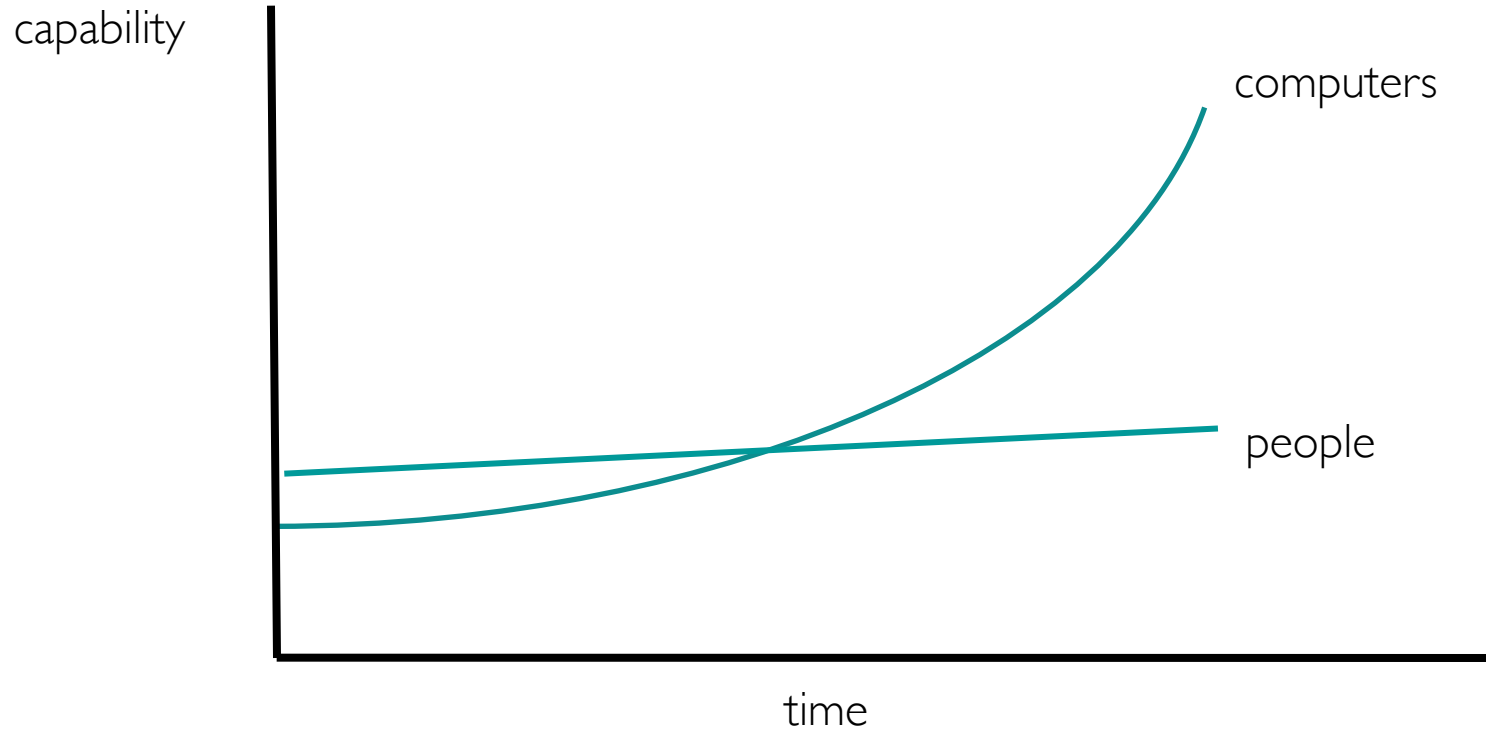


# Computer capabilities are exploding

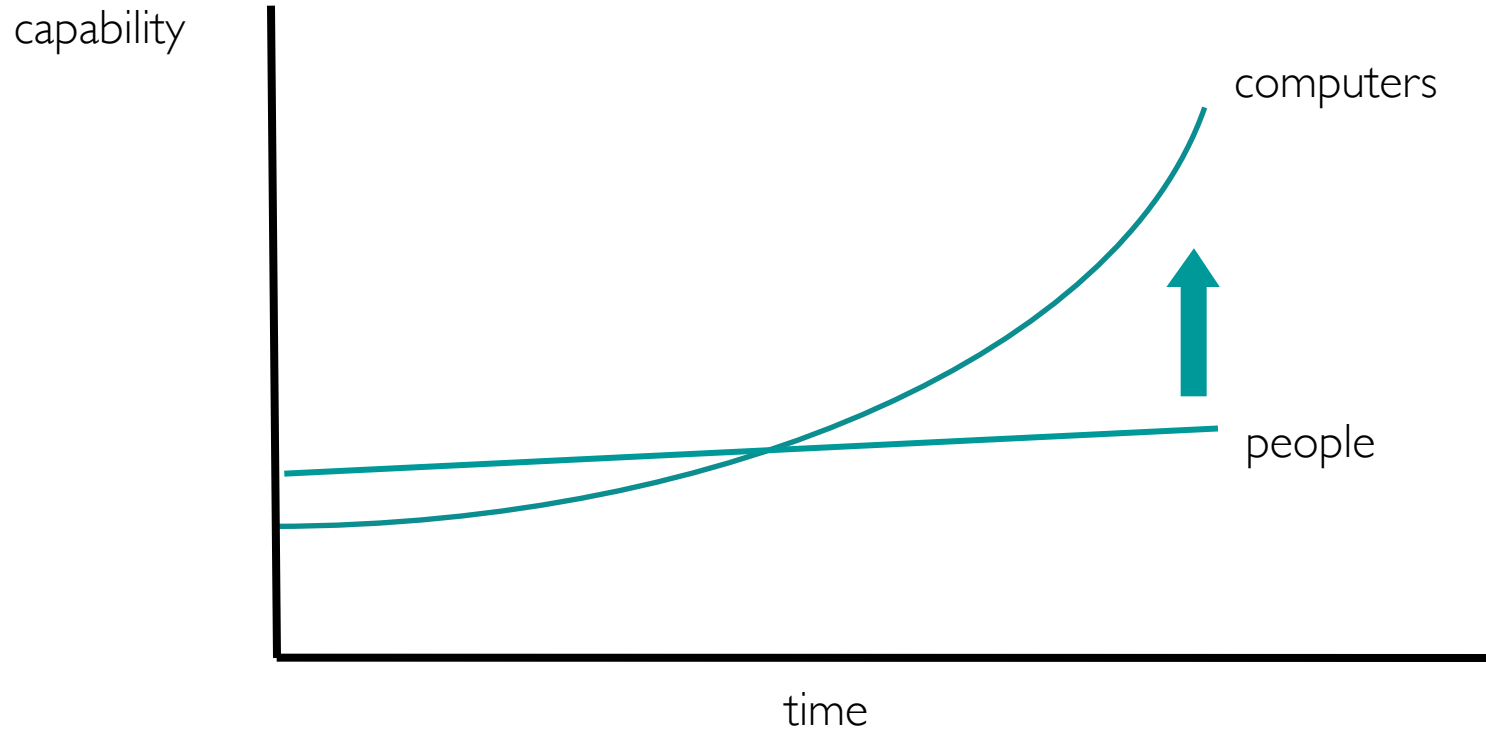


but human capabilities are not...

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 inefficient techniques and are  
constantly forced to change their behavior

# Desktops, the web and apps ...

Require constant relearning:

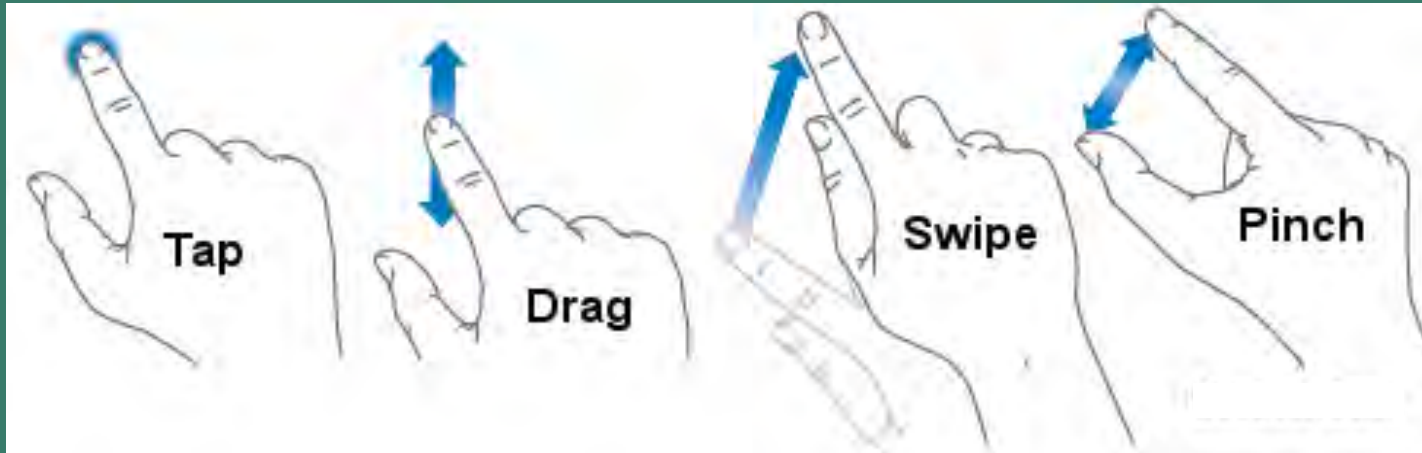
- each new version introduces arbitrary changes
- each system requires slightly different interaction

Require high visual attention

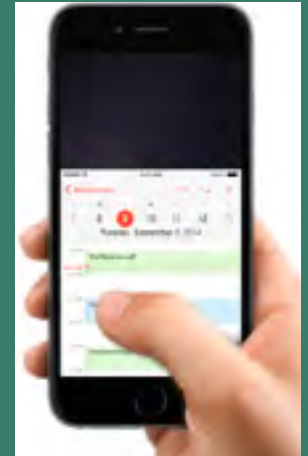
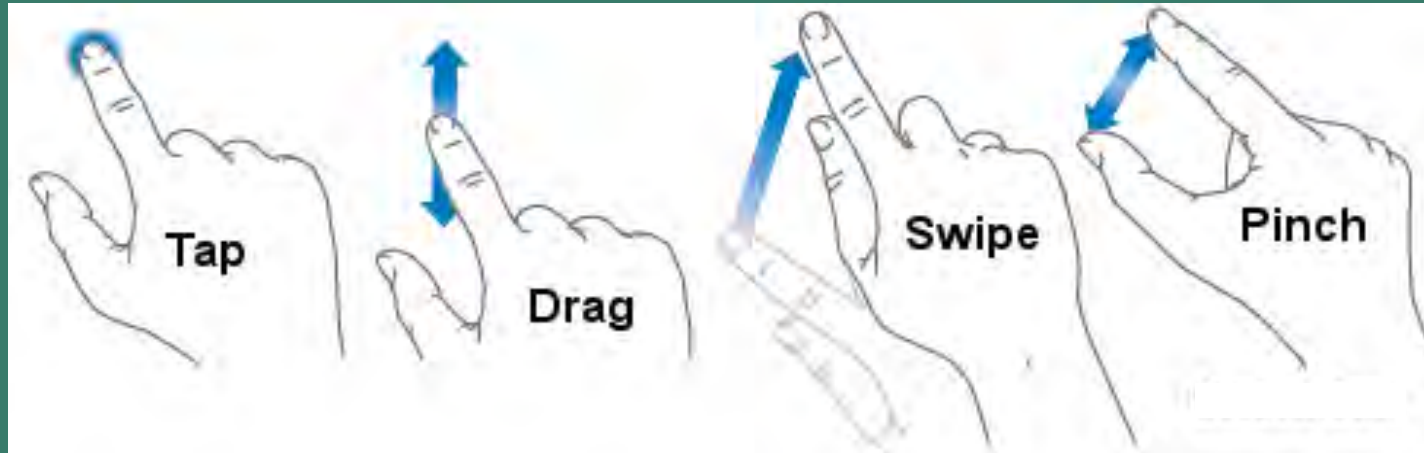
Do not scale

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

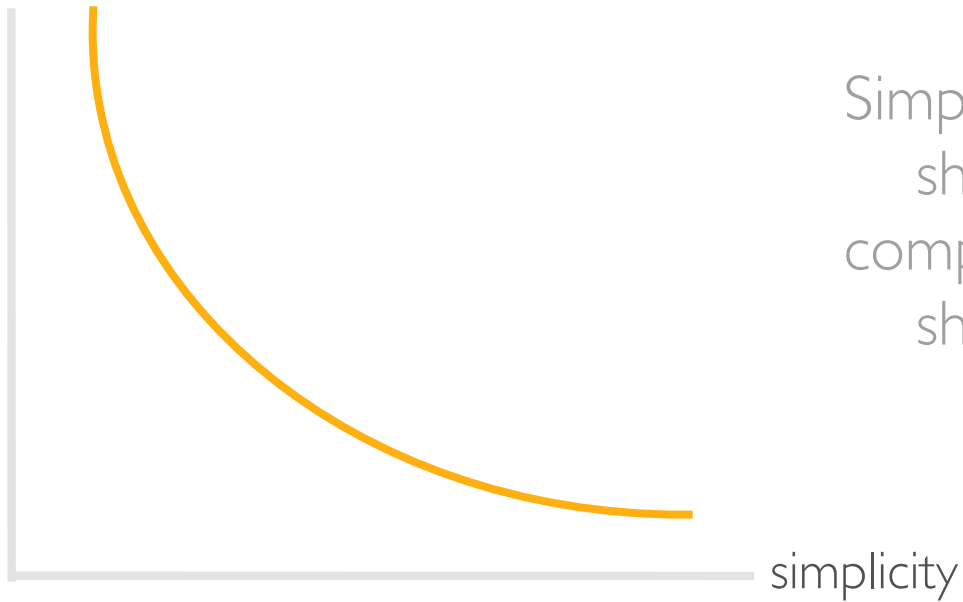
Early assumptions about graphical user interfaces  
no longer hold

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

power



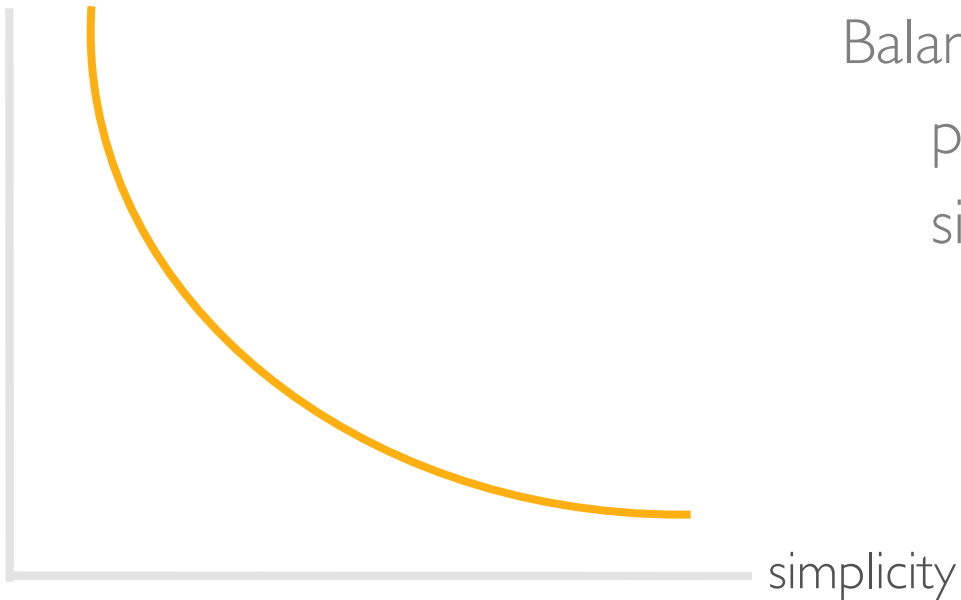
Goal:

Simple things  
should be simple ...  
complex things  
should be possible



# Design Trade-offs

power



Fundamental challenge:

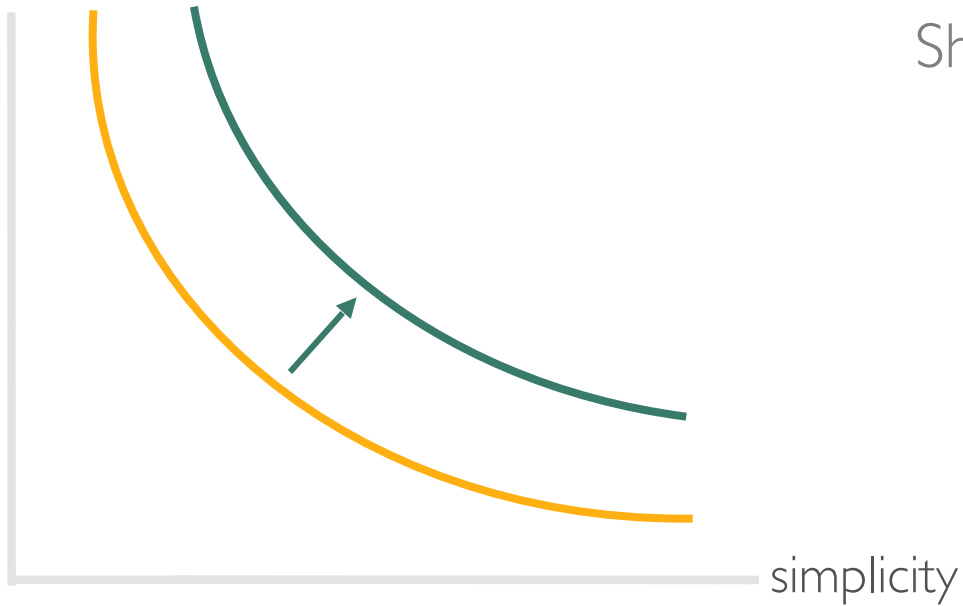
Balance trade-off between:

power of expression

simplicity of execution

# Design Trade-offs

power



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





A prehistoric scene featuring a hominid with dark, shaggy fur. The hominid is standing on a sandy ground, holding a large, light-colored bone tool (possibly a hammer or club) high in its right hand. In the foreground, there is a large animal skull and several other bones scattered on the sand. The background shows a bright sky with scattered white clouds. A green rectangular box with white text is overlaid on the right side of the image.

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

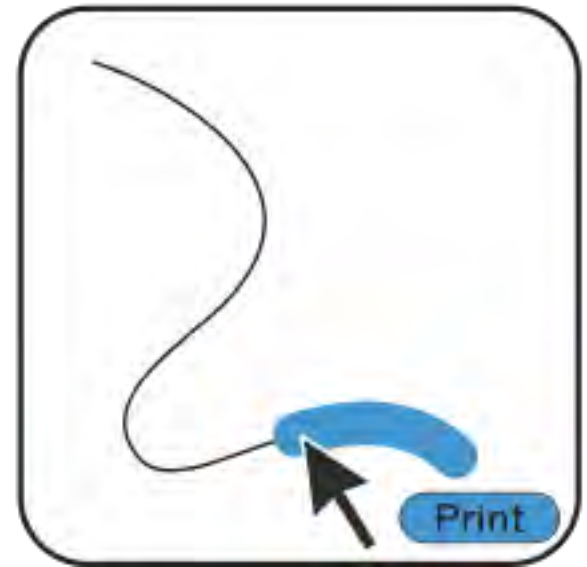
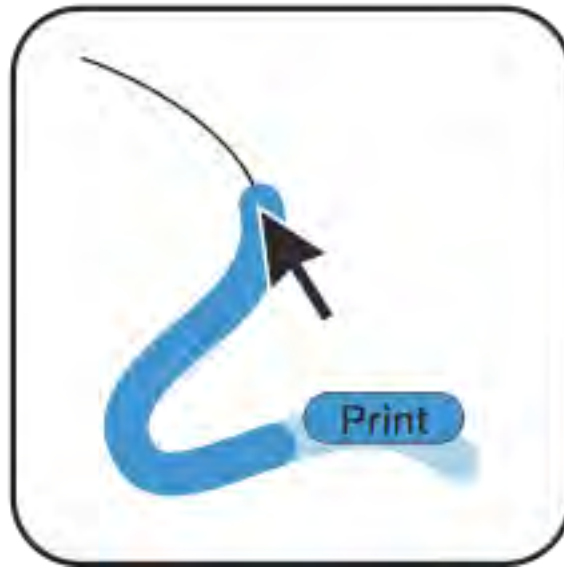
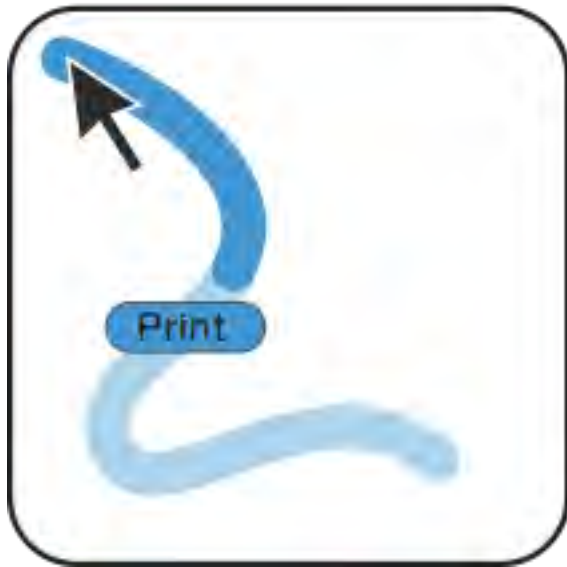
People should choose and  
control their own tools

Software tools should be  
easy to appropriate

# Octopocus: Learning complex gestures

Dynamic partnership:

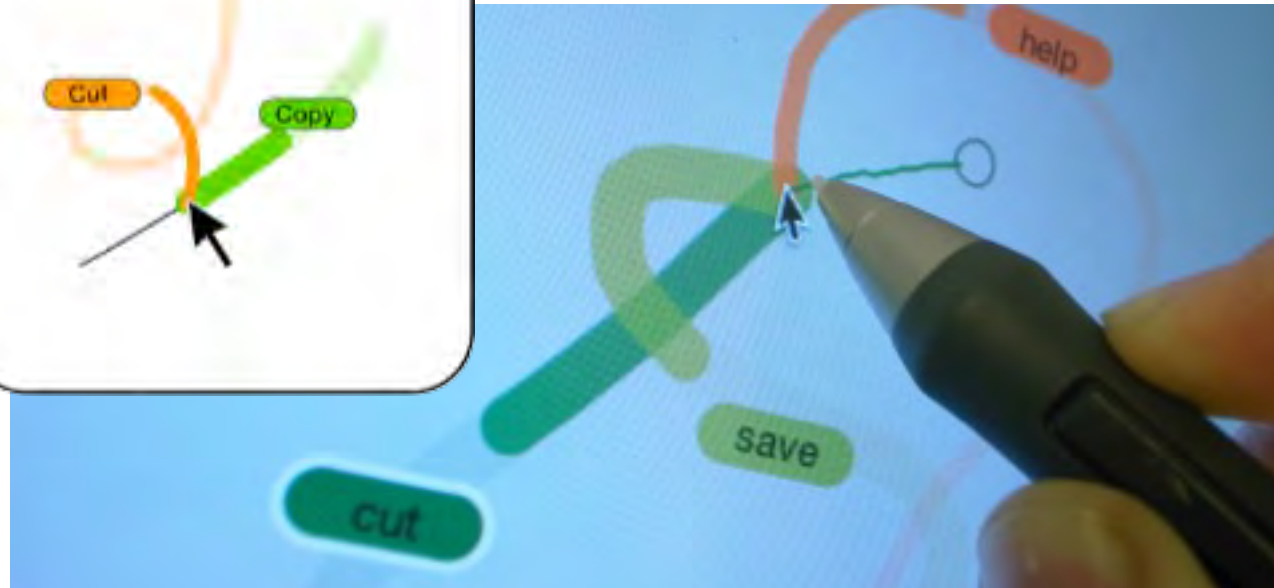
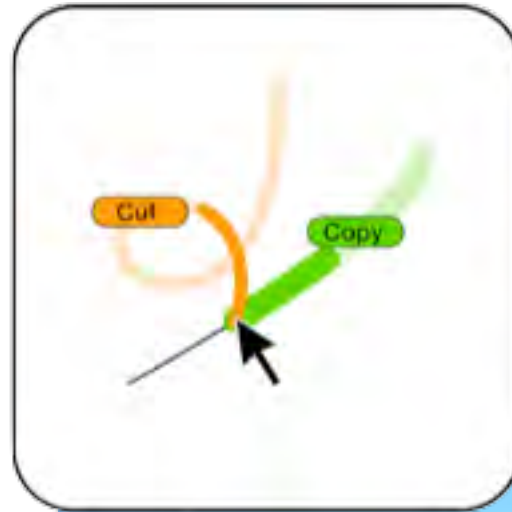
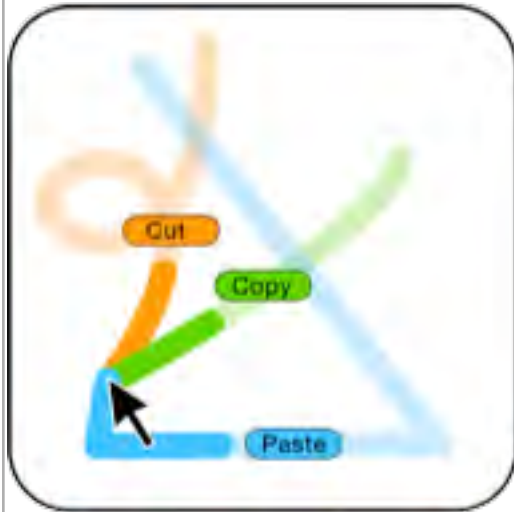
Progressive algorithms reveal intermediate recognition states



Octopocus:

Learning complex gestures

Experts *just do it*



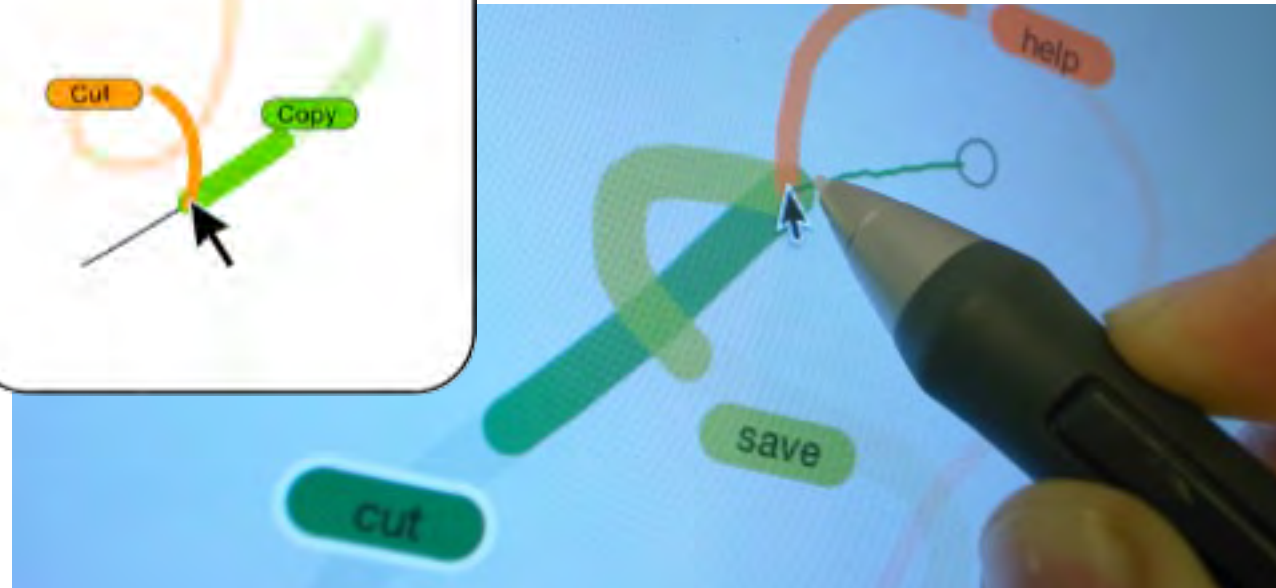
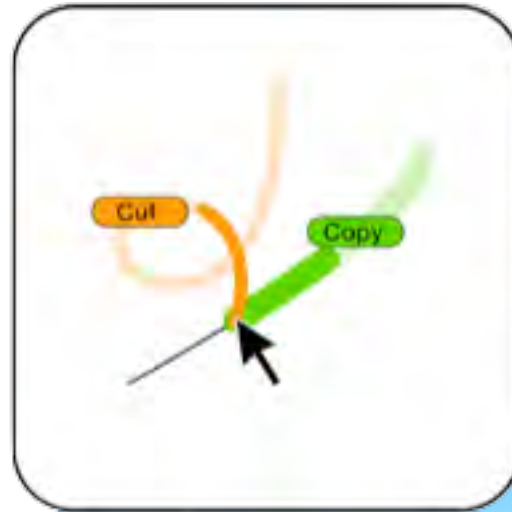
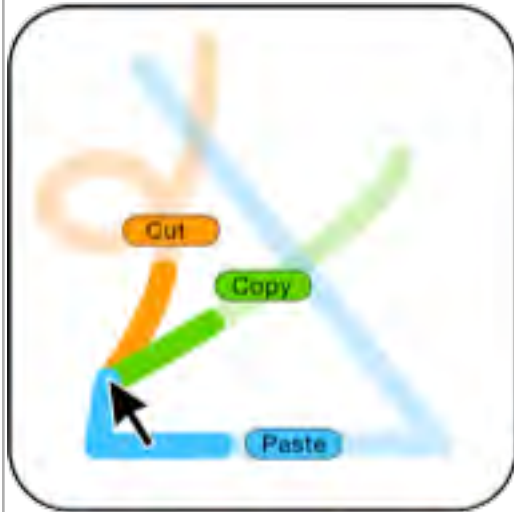


# Octopocus: Learning complex gestures

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

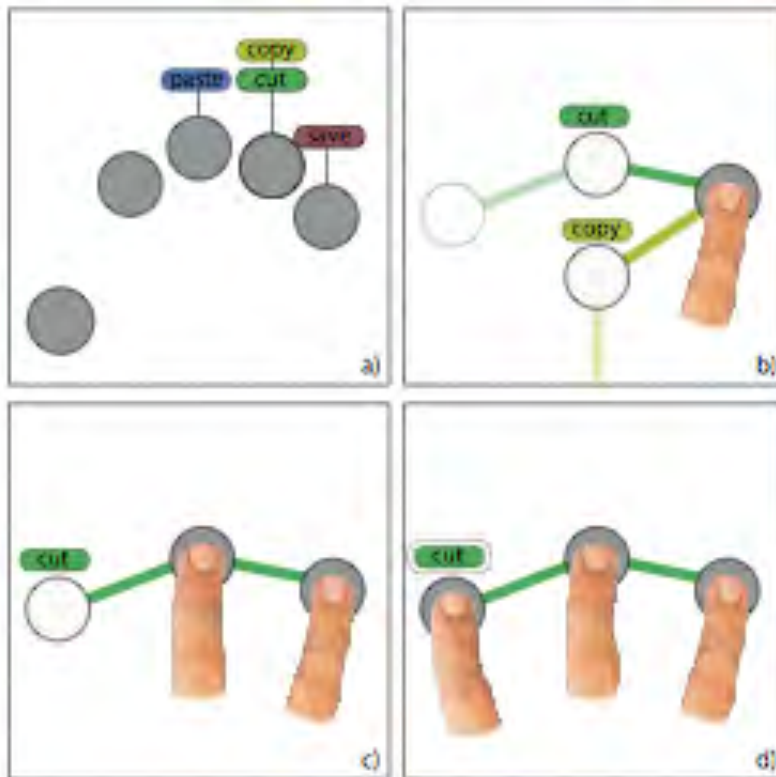
UIST 2008



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

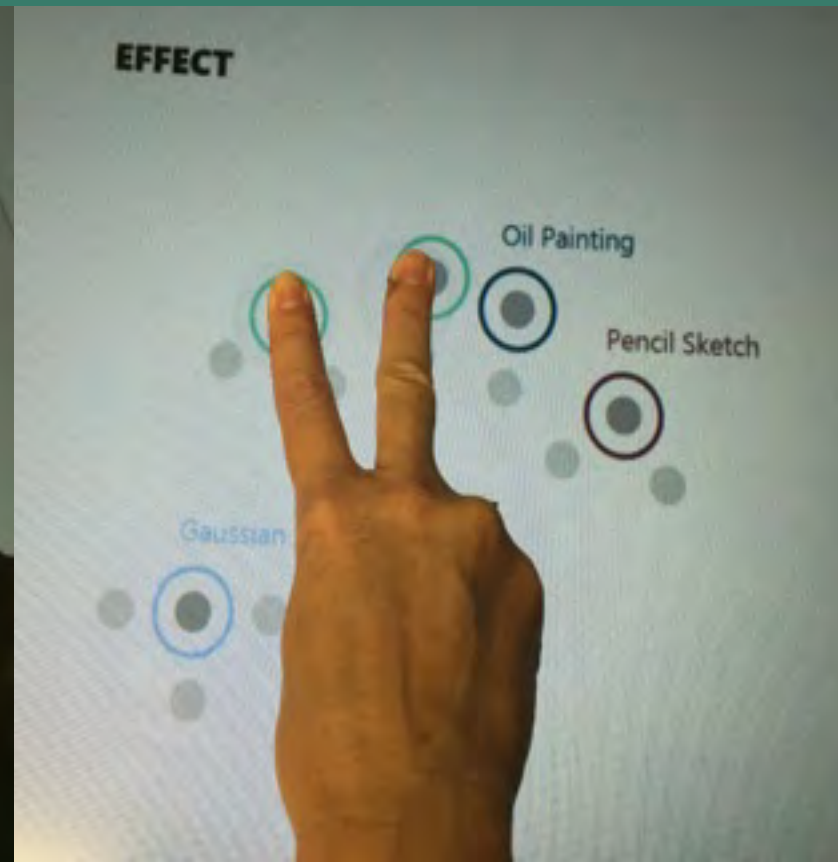
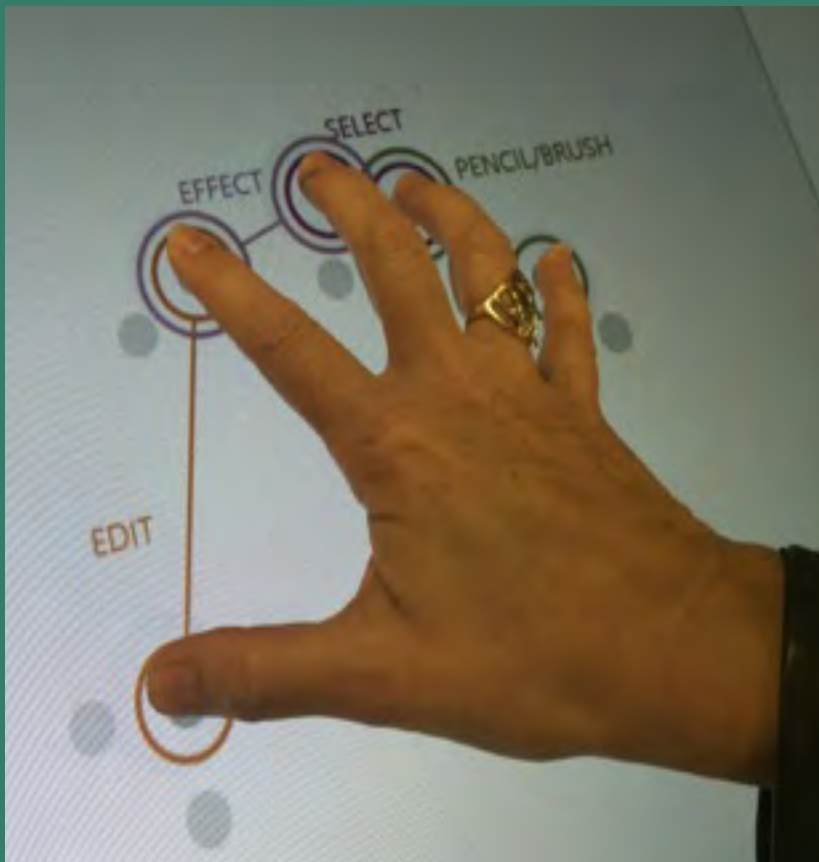


# Arpege: Design and learning of multifinger chord gestures

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 annotate flight strips

Marks can be linked to RADAR and other computer functions

Users define what marks mean

ident

tap

click

dblclick

press

endpress



Strip Editor

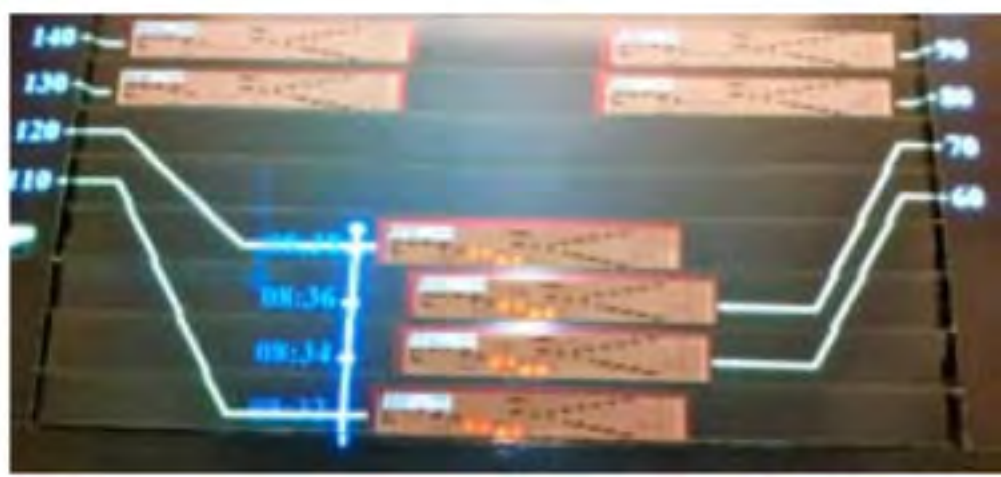
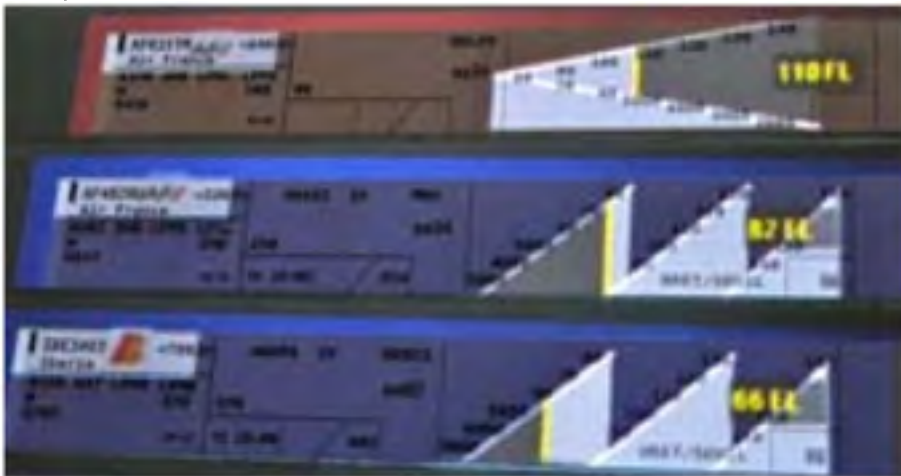
action:	mark	area:	balises	cmd:	Directe	Changer	
D L H 5 2 8 5	280		280	RIVES	L3A	TENEK	SPR
CL65	EBL EDDS 280	310		0	3	8	14
	GV 134 85		RIVES	10	10	10	10

identification    exitlevel    ILS    balises    annot    sector

OK  
Tel  
RAZ

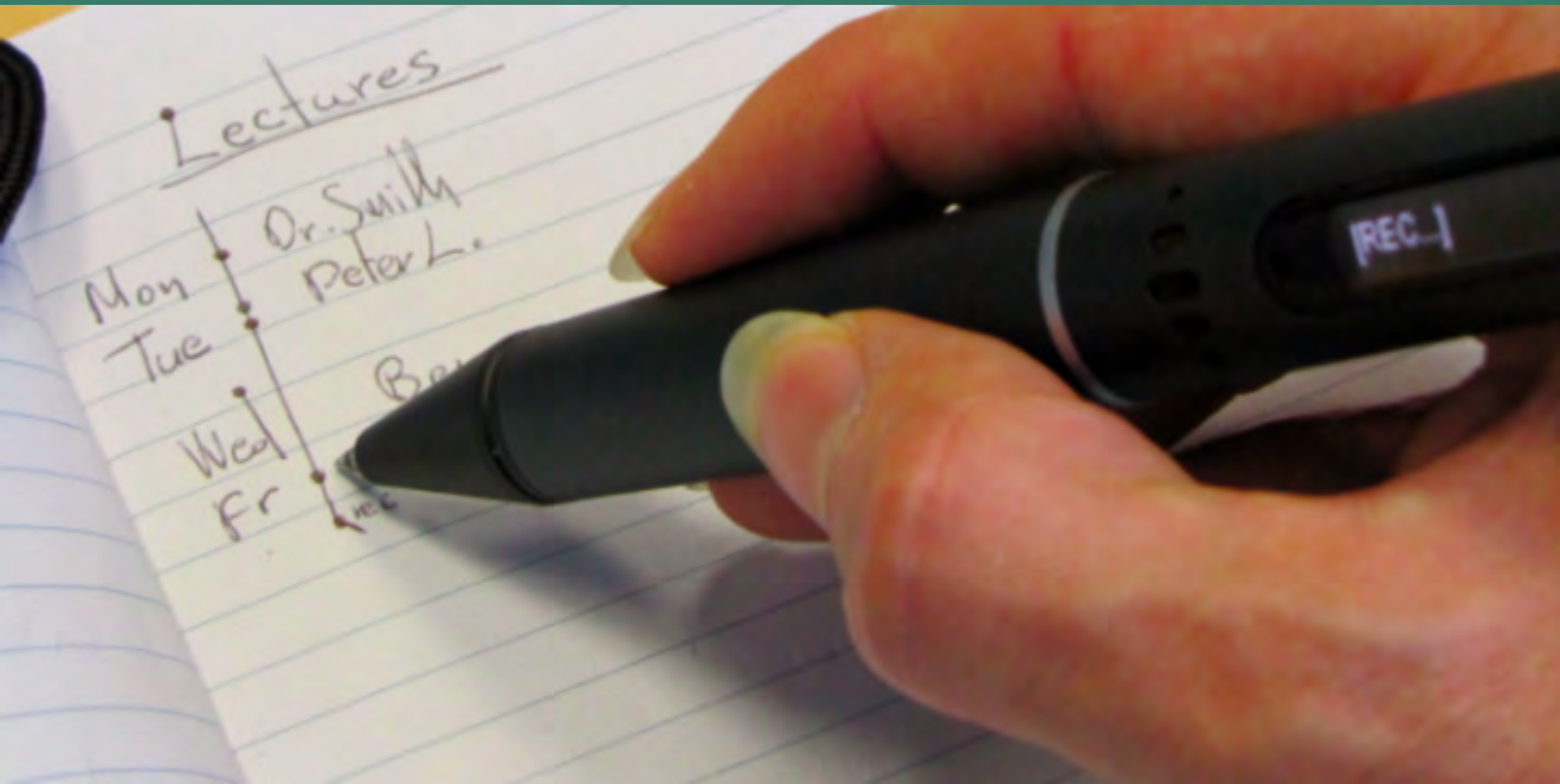
# Striptic

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



# Knotty Gestures

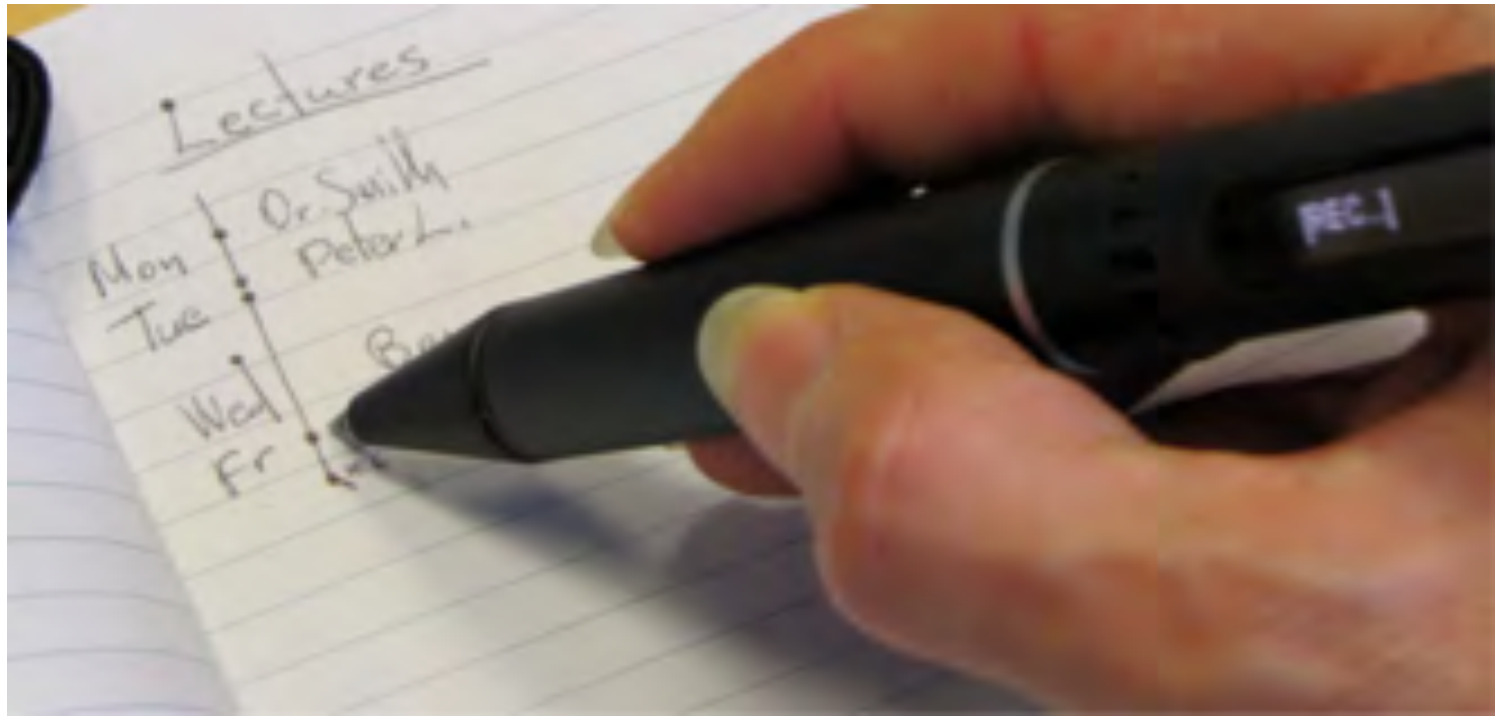
Draw a dot, define a command  
Interact while writing  
Interact with command later



# Knotty Gestures

## Interactive Paper

Users interact as they write  
or define their own gestures  
and interact with them later



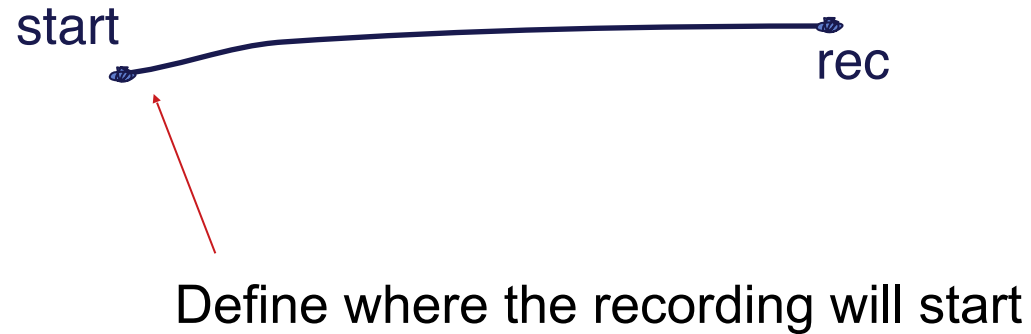
# Knotty Gestures: Creating an interactive controller

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



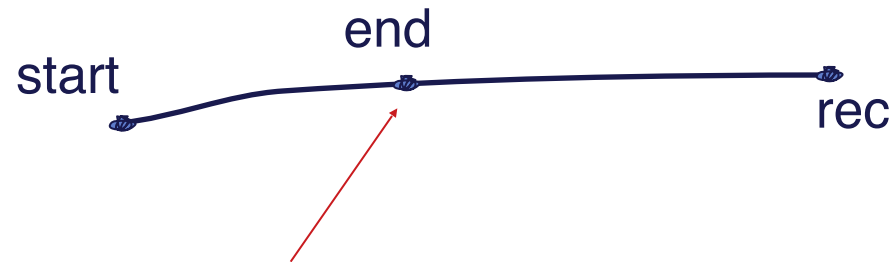
Choose "recording" to define the type of line

# Knotty Gestures: Creating an interactive controller



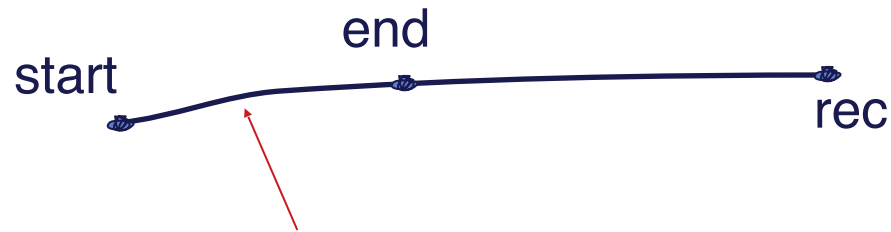


# Knotty Gestures: Creating an interactive controller



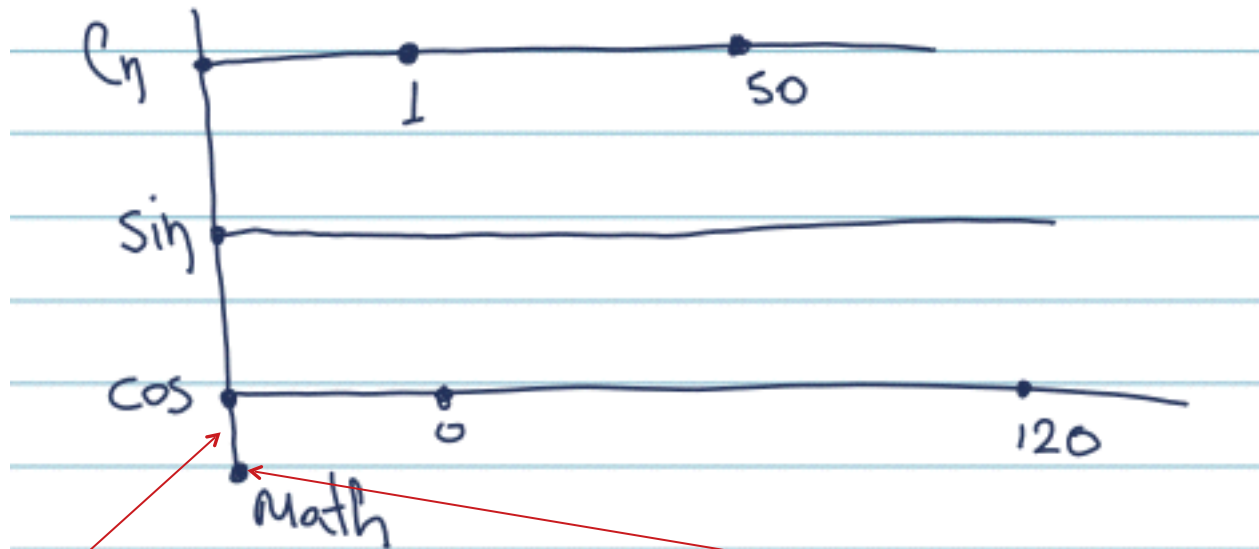
Define an end point for the recording

# Knotty Gestures: Creating an interactive controller



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

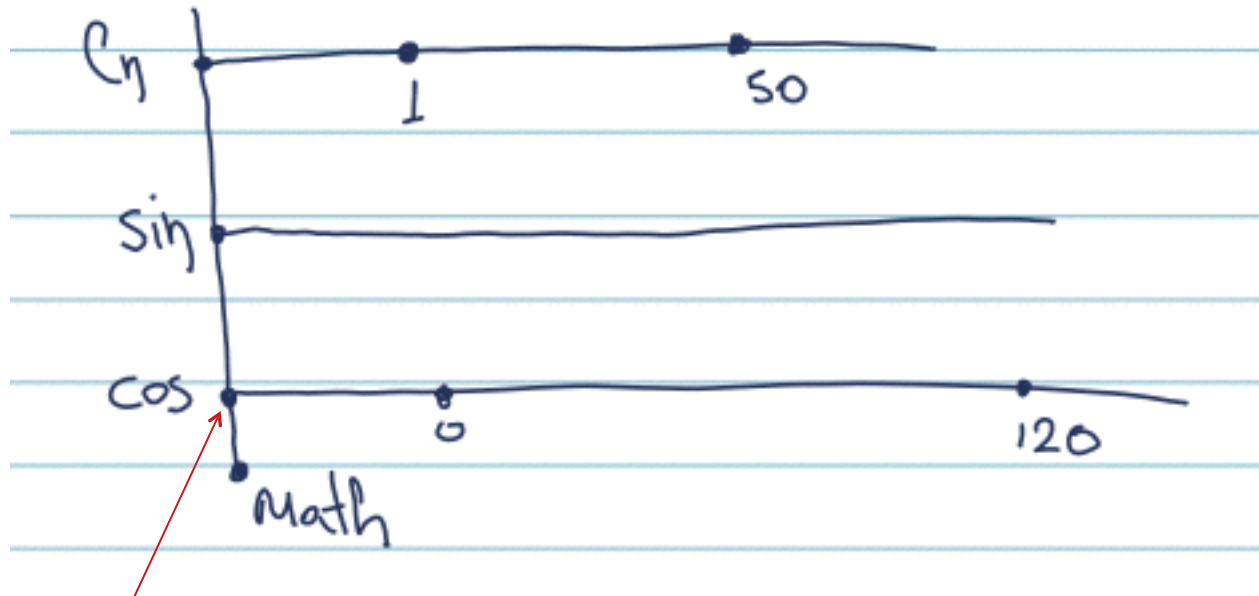
# Drawing a Math Calculator



This line acts as a base for attaching mathematical value sliders

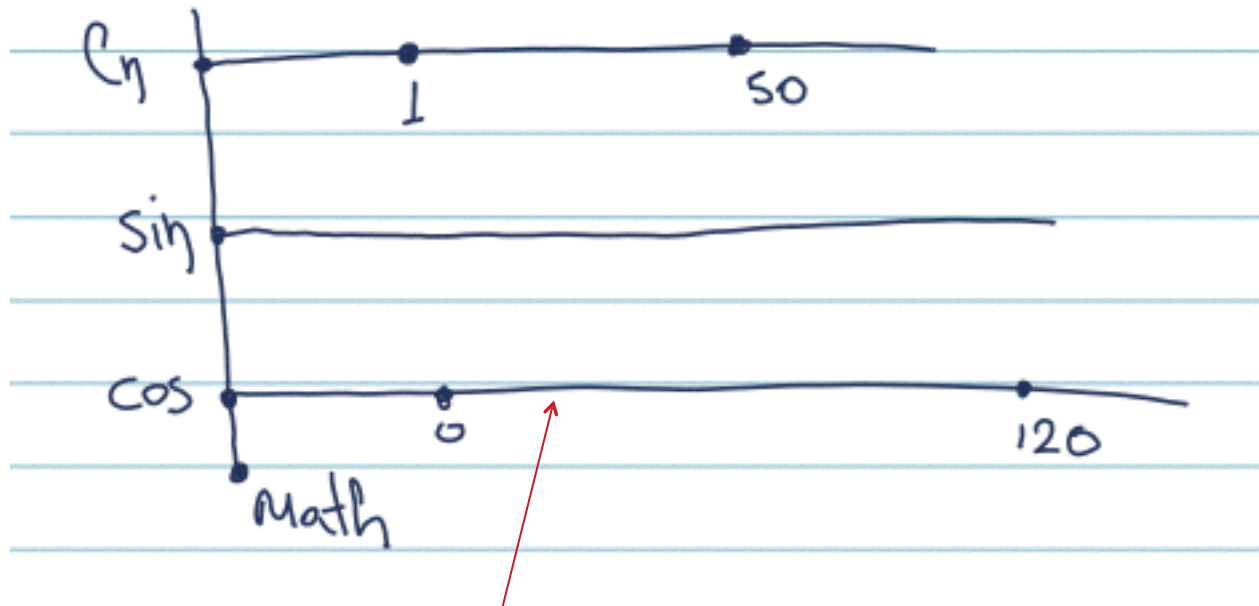
The knotty gesture at the end defines the type

# Drawing a Math Calculator



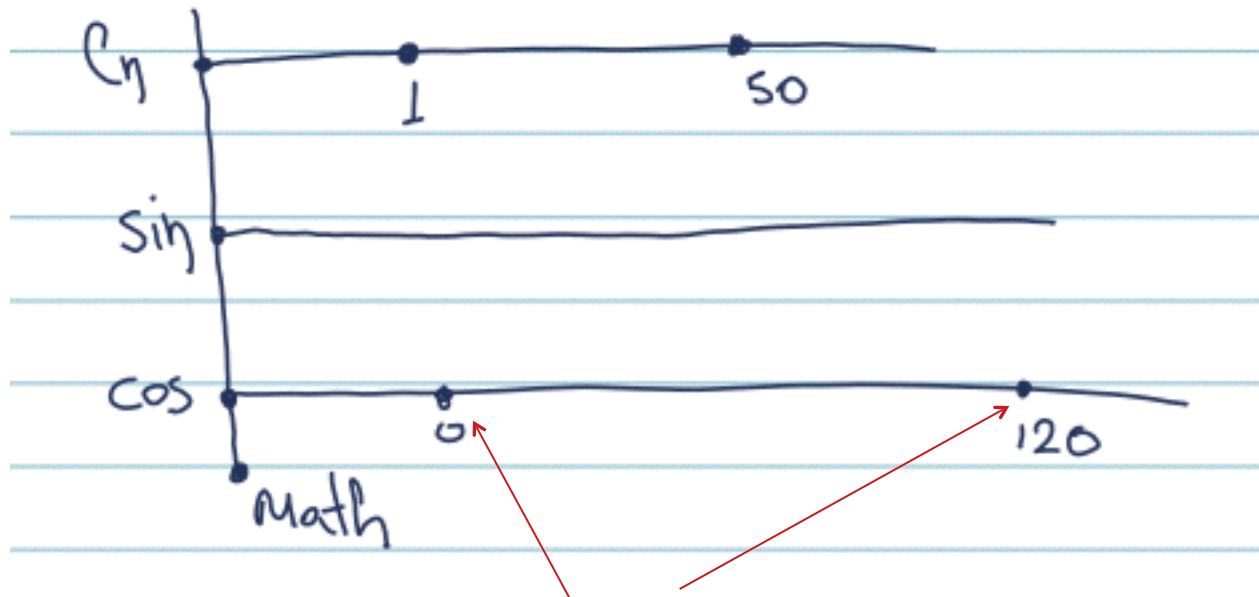
Any knot drawn on line lets the user select a mathematical function

# 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



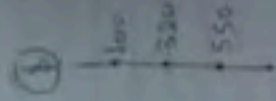
Knots may define ranges or act as traces of past interactions with specific values

$$\sqrt{53} + \sqrt{22} = 11.54$$

$$\cos^2 \theta + \sin^2 \theta = 2.516$$

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

$\cos^2 \theta$



## 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 *Musink* support the creative process?

What are the design implications for *Musink v2*?



# Semi-Structured Delayed interpretation

Key insights:

Spatial structure on paper

improves recognition  
under user's control

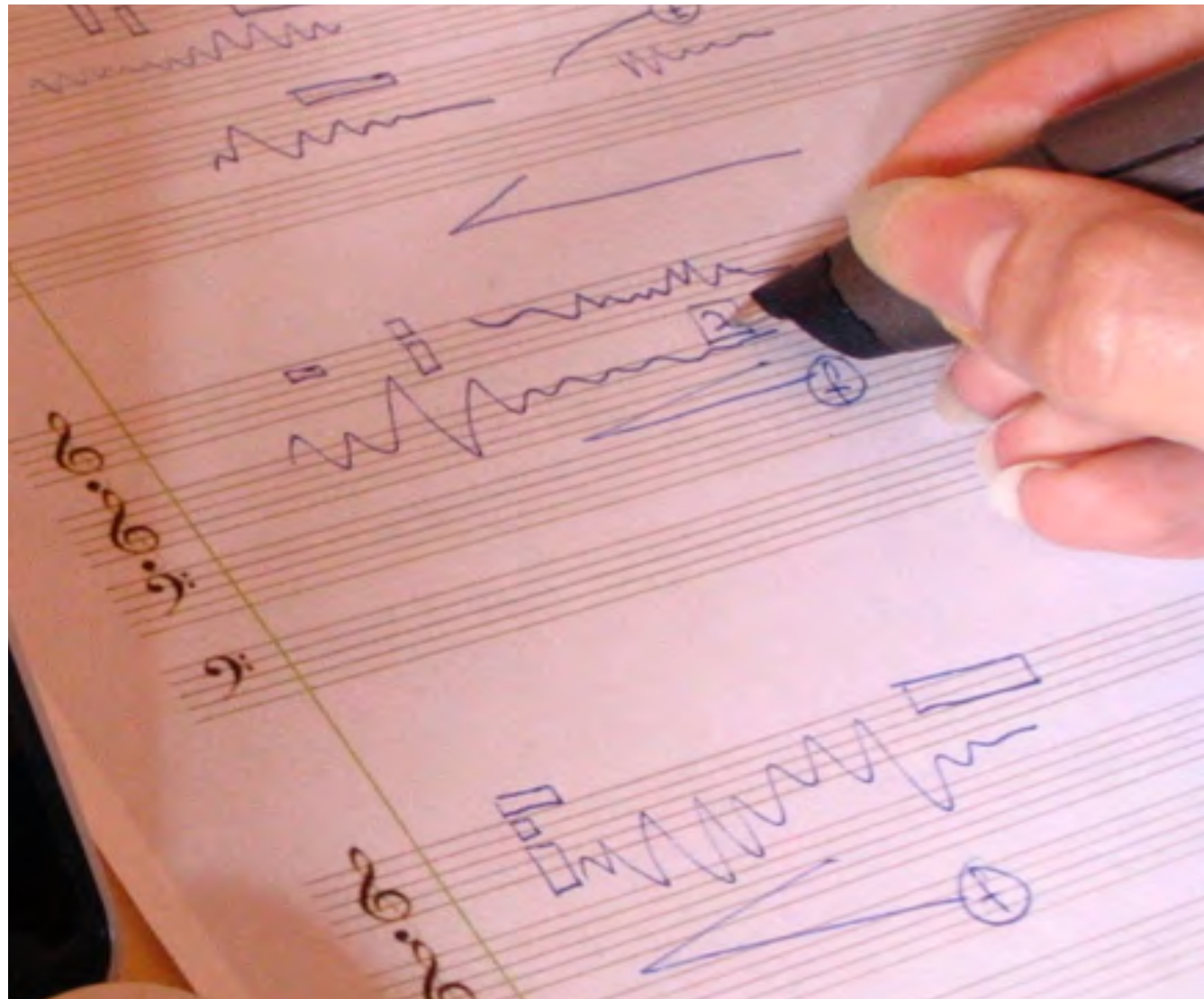
Recognition need not be immediate

users decide *when* to interpret  
interpretation *changes over time*



# MusInk Define meaning of gestures over time

User decides if and when to interpret each gesture







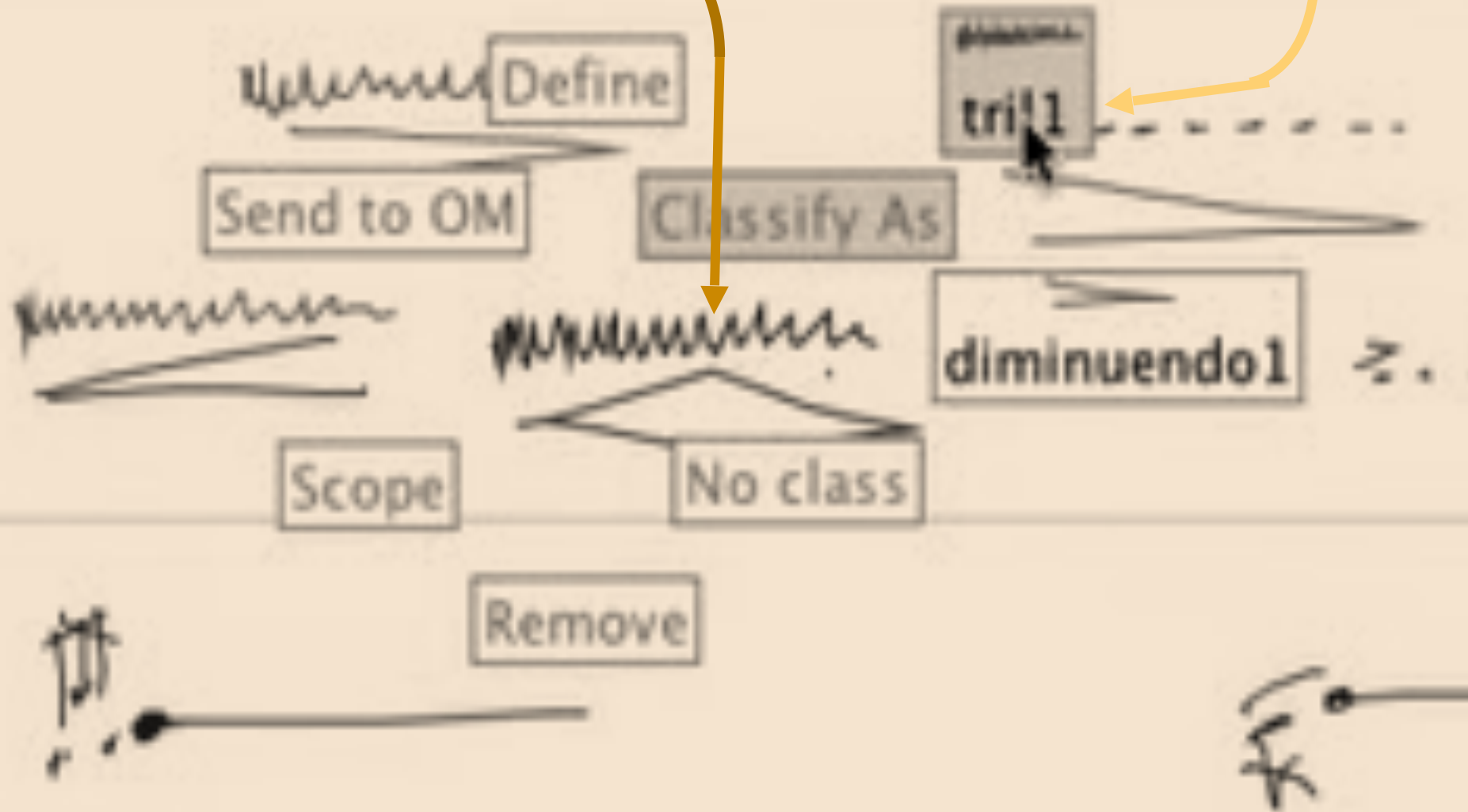
Create interactive annotations

A musical score with handwritten annotations in blue ink. The annotations include a circled '5' with a downward arrow, a box labeled 'accel' with a bracket, a box labeled 'mf', a box labeled 'Solo fm', and a circled '2' with an upward arrow. There are also blue wavy lines and brackets indicating specific musical phrases or dynamics.

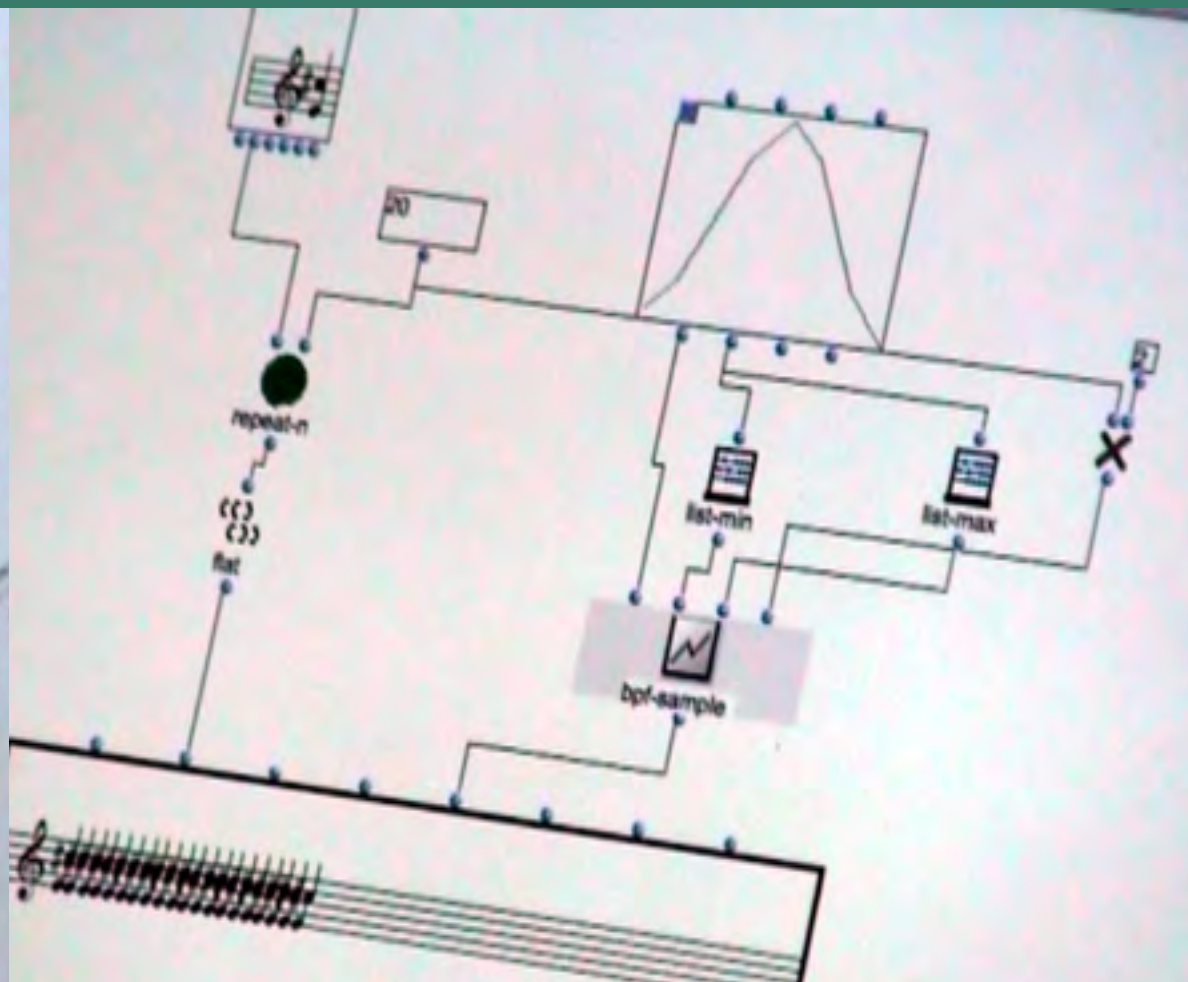
A musical score with handwritten annotations in blue ink. The annotations include a circled '5' with a downward arrow, a box labeled 'mf', and a box labeled 'Solo fm'. There are also blue wavy lines and brackets indicating specific musical phrases or dynamics.

A musical score with handwritten annotations in blue ink. The annotations include a circled '5' with a downward arrow, a box labeled 'mf', and a box labeled 'Solo fm'. There are also blue wavy lines and brackets indicating specific musical phrases or dynamics.

Reclassify a 'squiggle' and turn it into a trill

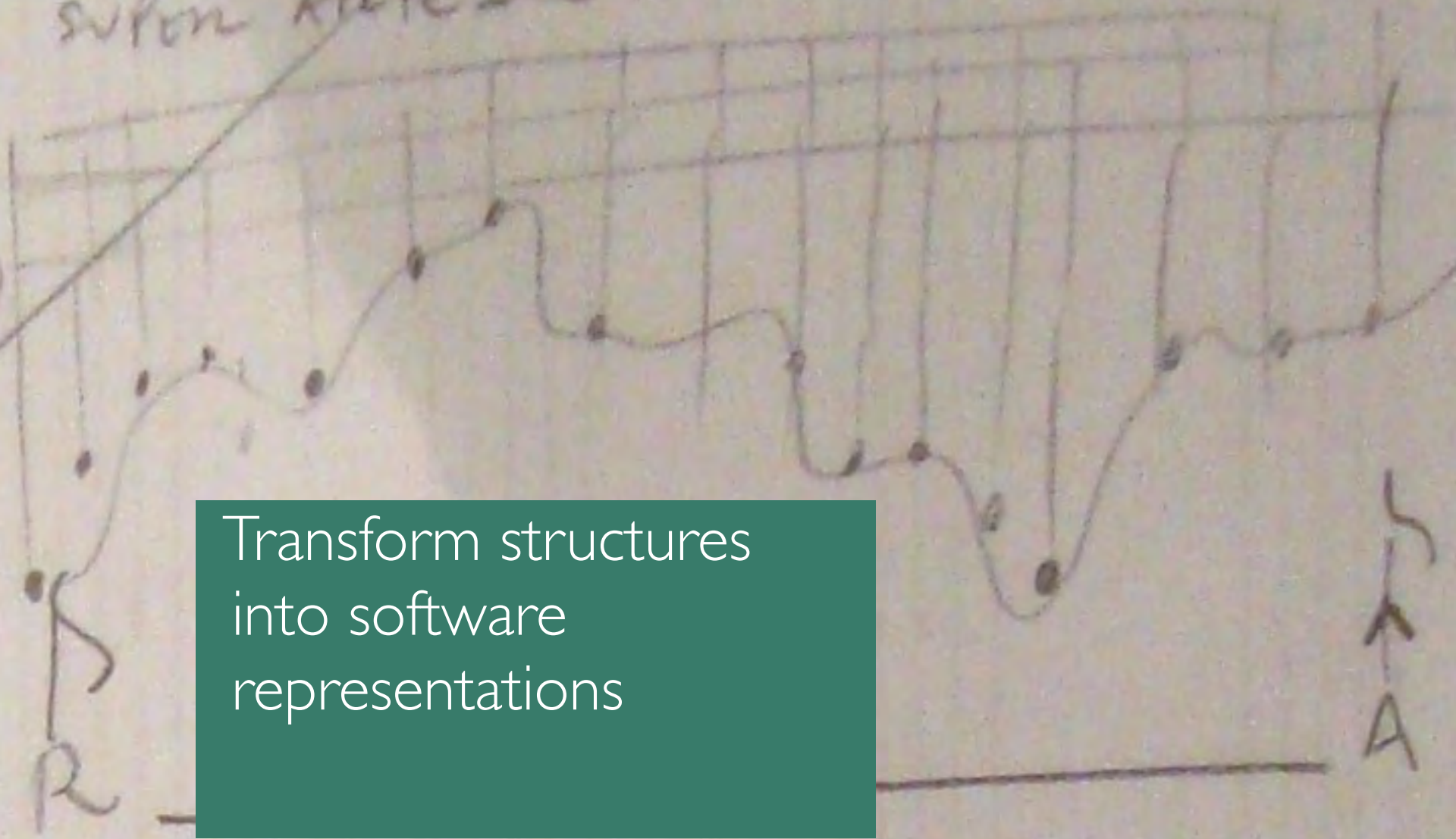


From symbols to wave forms:  
Interpret a tremolo gesture  
as a waveform by *OpenMusic*

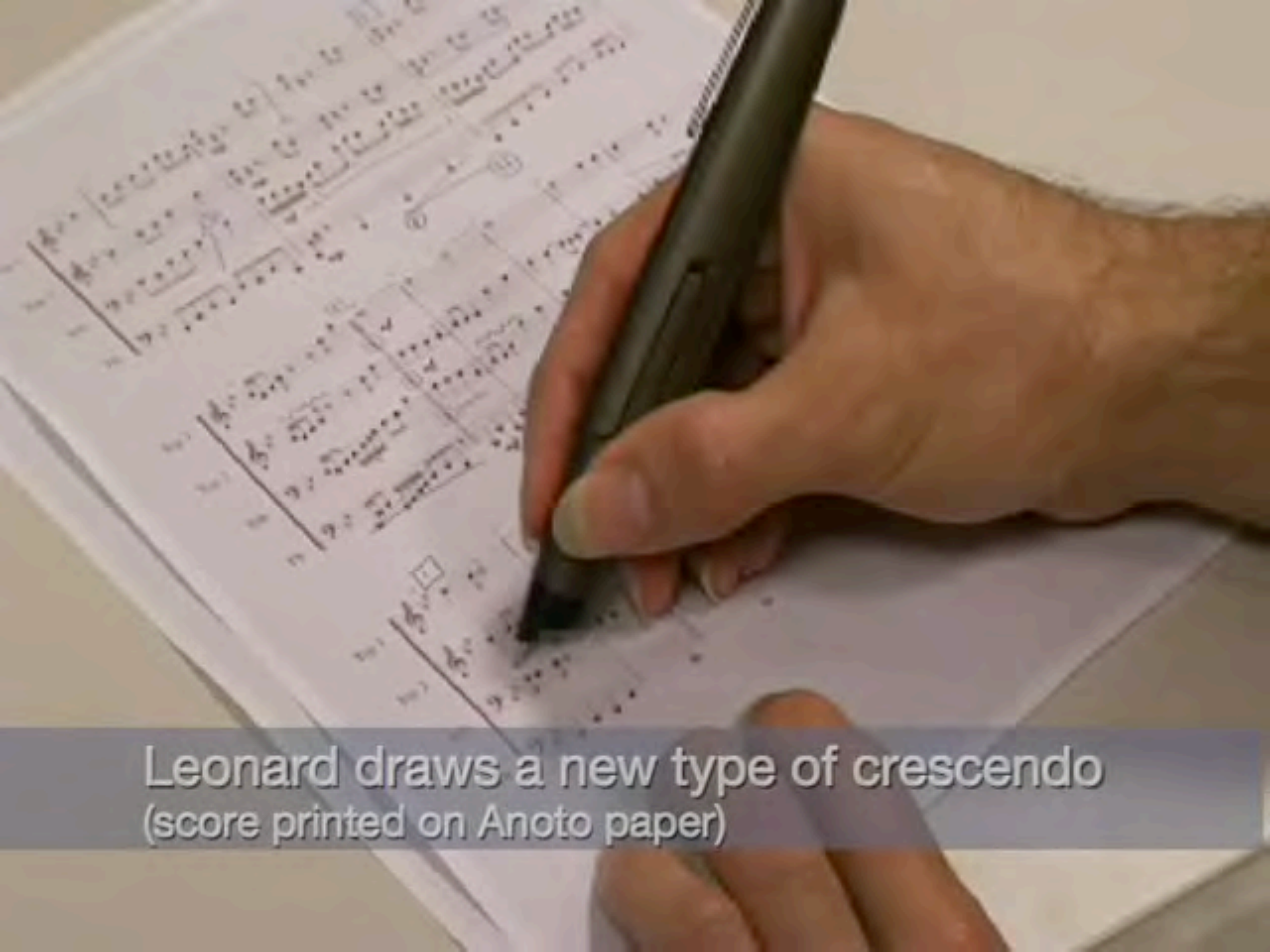




SUPON KRIPESIO



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



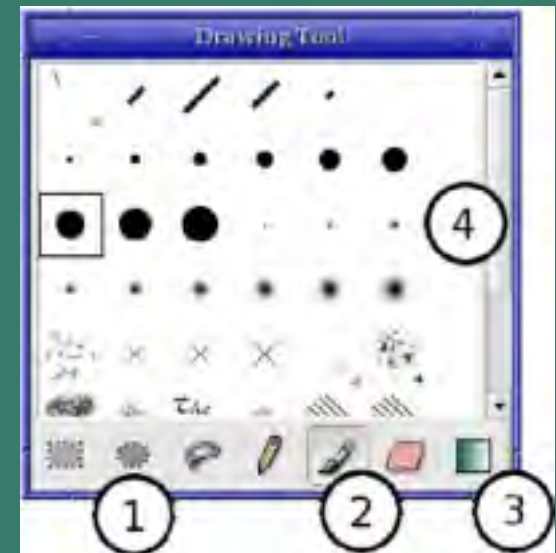
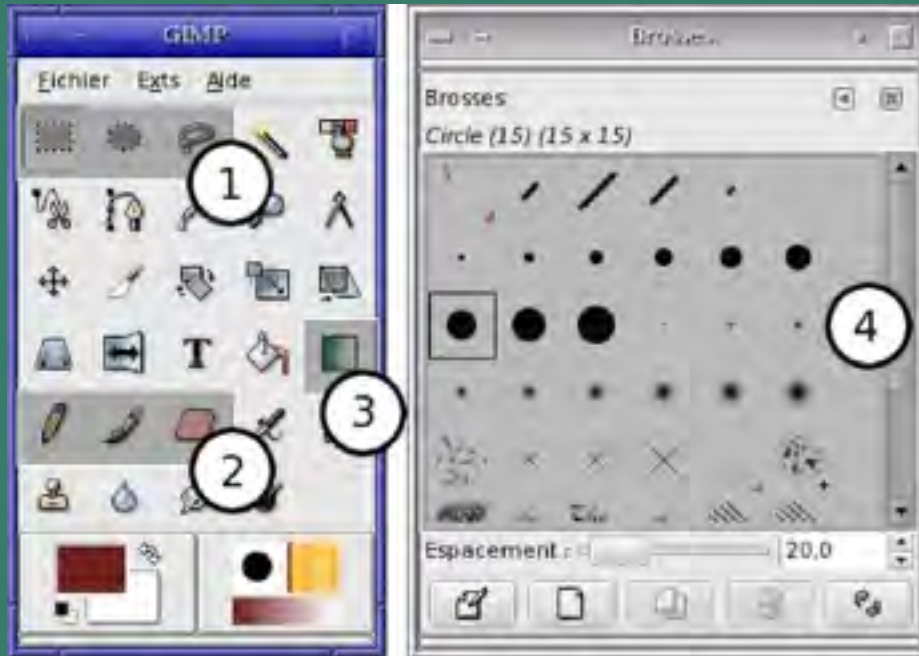
connectors



# 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

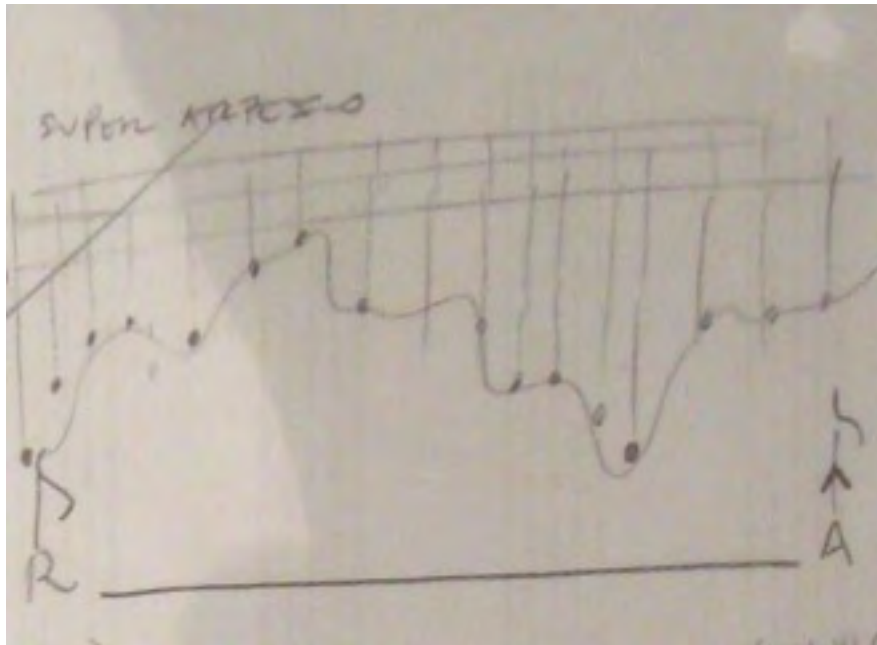


# Substrates

Define the structures and rules  
Ways to interpret the data

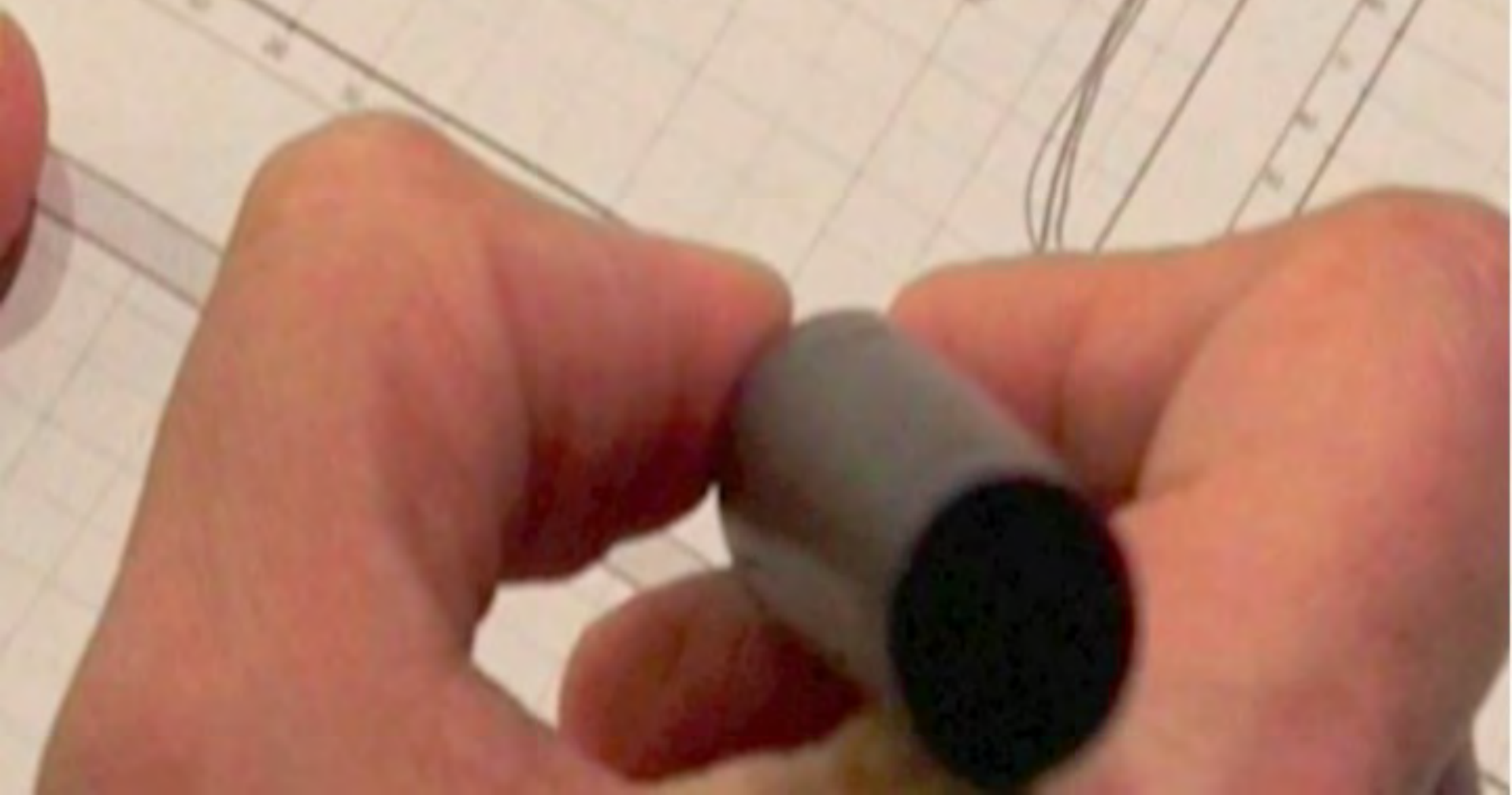
# Different structures

to facilitate  
interpretation



A handwritten musical score consisting of five staves. The top staff is a bass clef staff with a 3/8 time signature. The second staff is a treble clef staff with a 3/8 time signature, containing a note with "mod2" written above it and the handwritten text "voc buffer 4" to its right. Below this staff are two boxes labeled "A5" and "A6". The third staff is a treble clef staff with a wavy line. The fourth staff is a treble clef staff with a horizontal line and an arrow pointing left. The fifth staff is a treble clef staff with a horizontal line.

Worm





# Paper Substrates: create own language & structure

Composers  
create new structures  
for interpreting and  
composing music

The diagram illustrates a multi-layered musical structure. At the top, a hexagonal grid contains various musical notes such as do#, sib, sol, fa, mi, re, and fa#. A digital waveform is overlaid on this grid. Below the grid, a musical score is shown with two staves (treble and bass clef). The notes and lines in the score are colored to match the grid and waveform. A box labeled '16' is on the left, and numbered circles 1, 2, 3, and 4 are scattered throughout the diagram.

92

136

Handwritten musical notation on a staff, featuring various notes, rests, and markings. The notation is somewhat abstract and appears to be a sketch or a specific style of notation.

Handwritten musical notation on a staff, consisting of several vertical lines and some notes, possibly representing a specific rhythmic or melodic pattern.

Handwritten musical notation on a staff with a treble clef, showing notes and some markings. Below the staff are some circular diagrams or symbols.

Handwritten musical notation on a staff, featuring a large diagonal line that crosses the staff, possibly indicating a specific musical concept or a structural element.

Handwritten musical notation on a staff with a treble clef, showing notes and some markings.

Handwritten musical notation on a staff with a treble clef, showing notes and some markings.

Handwritten musical notation on a staff with a treble clef, showing notes and some markings.

- 1 per per left case
- ~~speculation~~ speculation
- direction not f
- 1 - blue
- 1 - blue pedestrian
- blue & blue
- blue - a better place

Composers create their own reusable structures

Handwritten musical notation on a staff with a treble clef, showing notes and some markings.

Handwritten musical notation on a staff with a treble clef, showing notes and some markings.

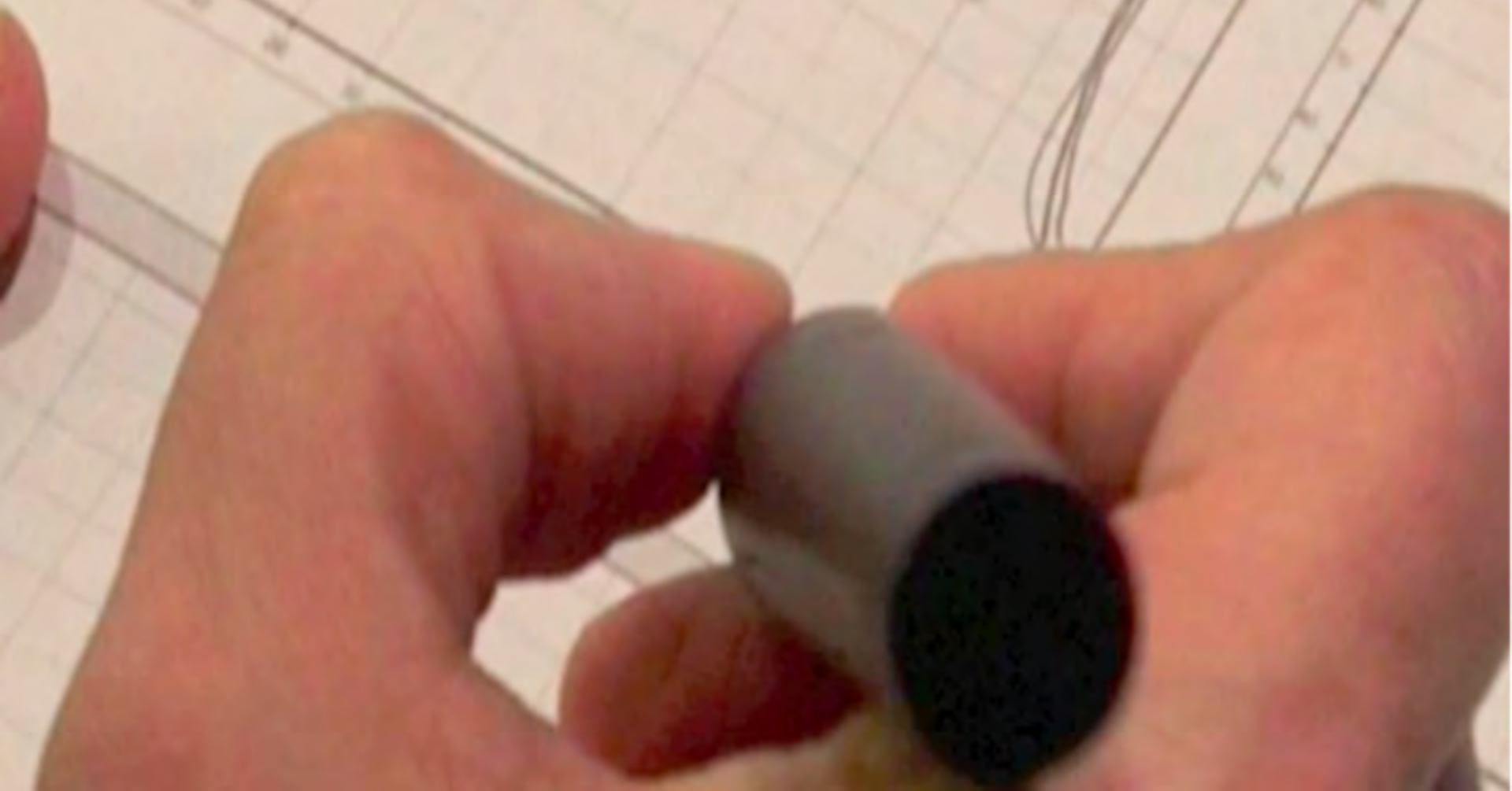
Handwritten musical notation on a staff with a treble clef, showing notes and some markings.

Handwritten musical notation on a staff with a treble clef, showing notes and some markings.

Handwritten musical notation on a staff with a treble clef, showing notes and some markings.

Handwritten musical notation on a staff with a treble clef, showing notes and some markings.

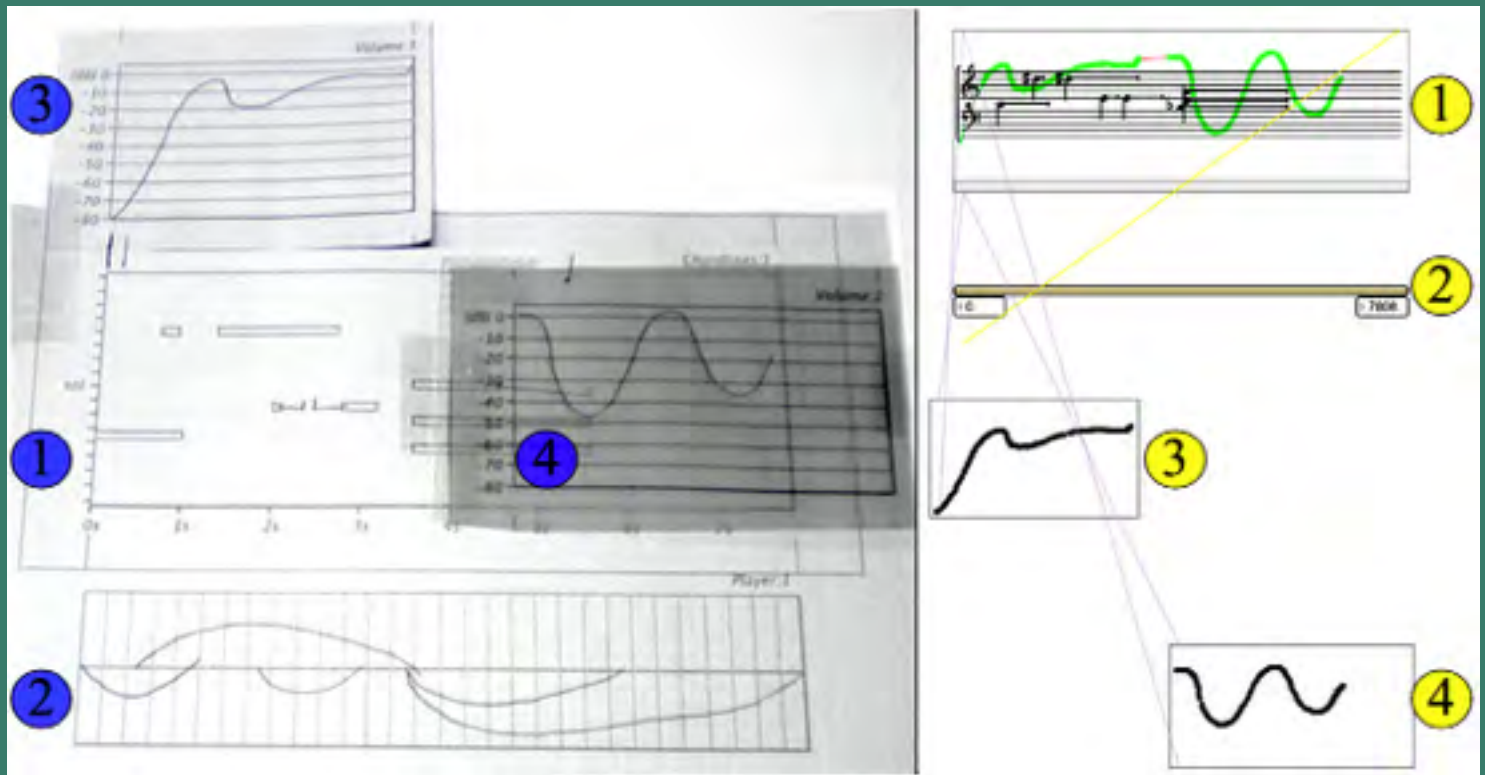
Worm



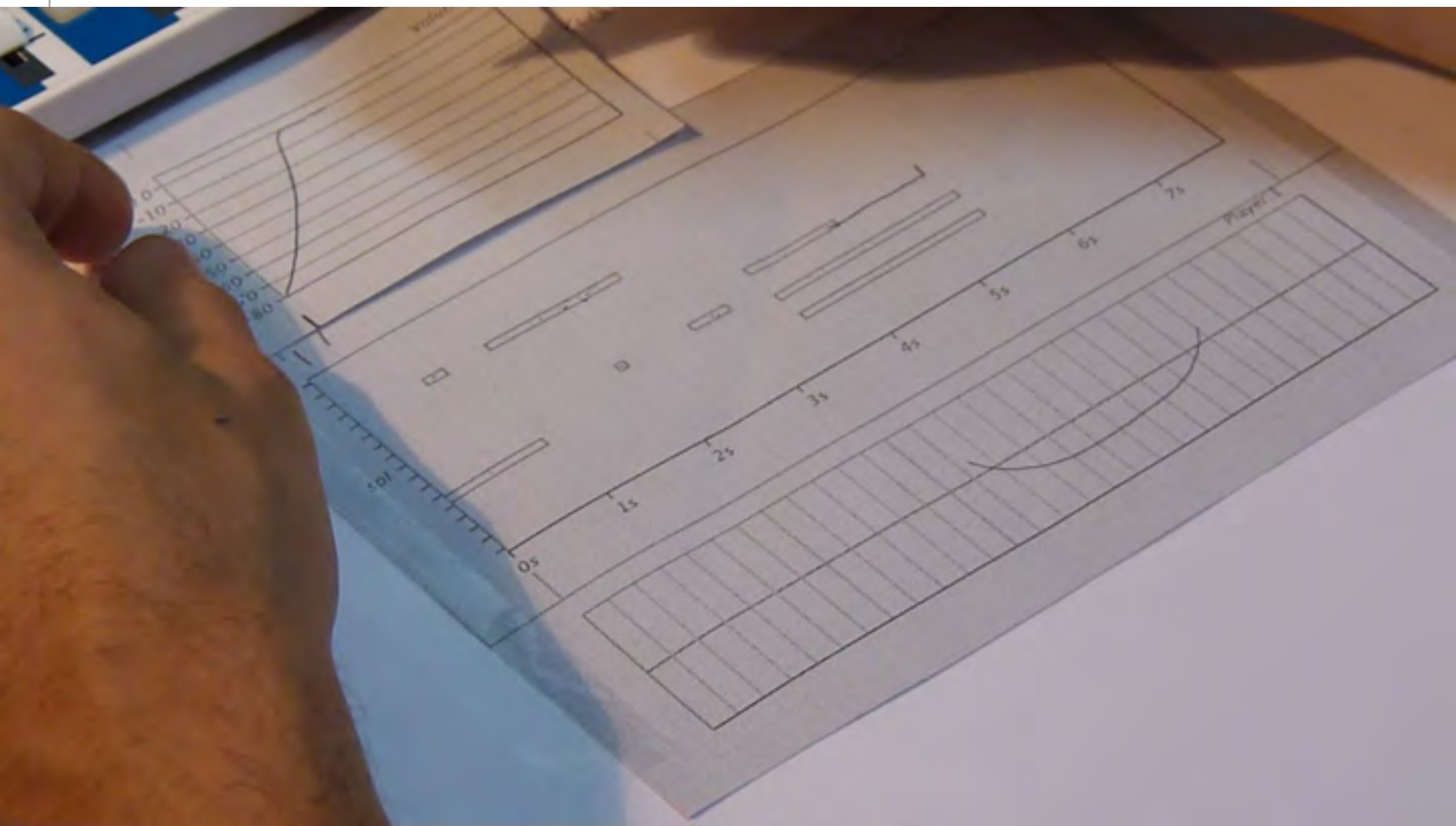


# Paper Substrates

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

The screenshot displays the PaperTonnetz software interface. At the top, a menu bar includes options: New, Open, Save As, Save, Print, activate, Erase, Network, Interfaces, and Pen Status. Below the menu is a ruler with centimeter markings from 0 to 29. The main workspace contains three pitch relationship diagrams:

- T[1,2,9] Bass:** A 5x5 grid of hexagons with pitch labels: Row 1: A, B, D, E, G; Row 2: Ab, C#, F#, F#; Row 3: Bb, Eb, F; Row 4: G, C, D; Row 5: F#, A, E. Below this grid are labels: F, AD, Bb, C#, Eb.
- T[3,4,5] Chords:** A 7x7 grid of circles with pitch labels: Row 1: Eb, Bb, D, A, C#, Ab, C; Row 2: G, F#, Bb, F; Row 3: C, B, Eb, D, A; Row 4: A, Ab, G, F#, Eb; Row 5: F#, C#, C, B, Eb; Row 6: Eb, Bb, A, Ab, C; Row 7: C, G, F#, F#, A.
- C minor Melody:** A 10x10 grid of hexagons with pitch labels: Row 1: Bb, Eb, G, C, Eb, Ab, C, F, Ab, D, F; Row 2: Ab, D, F, Bb, D, G, Eb, G, C, Eb; Row 3: G, C, Eb, Ab, C, Ab, D, F, Bb, D; Row 4: F, Bb, D, G, Bb, Eb, G, Eb, Ab, C; Row 5: Eb, Ad, C, F, Ab, D, Bb, G, F, Bb; Row 6: G, C, Eb, C, Ab, F, Ab, F, D, F; Row 7: D, Bb, G, Eb, Eb, G, C, Eb, C, Ab; Row 8: C, F, Ab, F, Bb, D, G, Bb, Eb, G; Row 9: Bb, Eb, G, Eb, Ab, C, Ab, F, D, F; Row 10: Ab, F, D, Bb, G, Eb, G, Eb, Eb; Row 11: G, Eb, G, C, Eb, Ab, C, Ab, D.

At the bottom of the workspace, there is a horizontal row of seven empty rectangular boxes. On the right side, there is a vertical toolbar with various icons for drawing and editing.

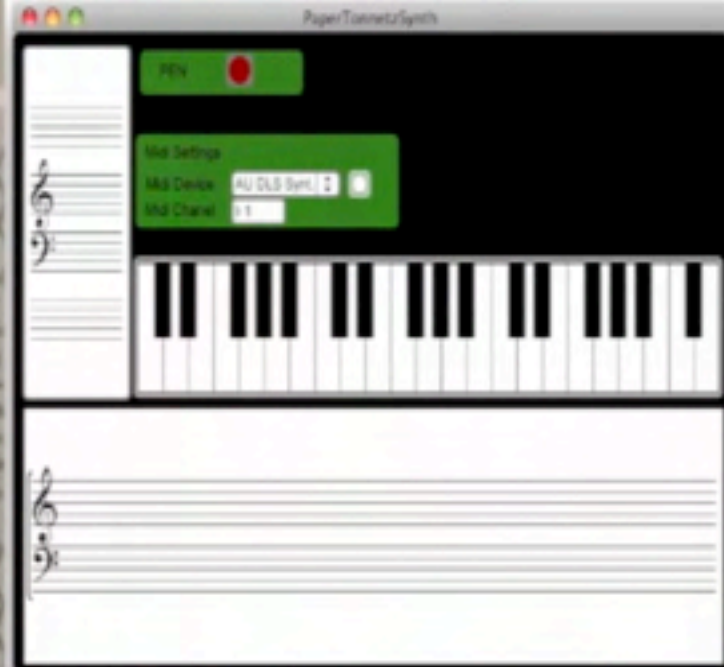
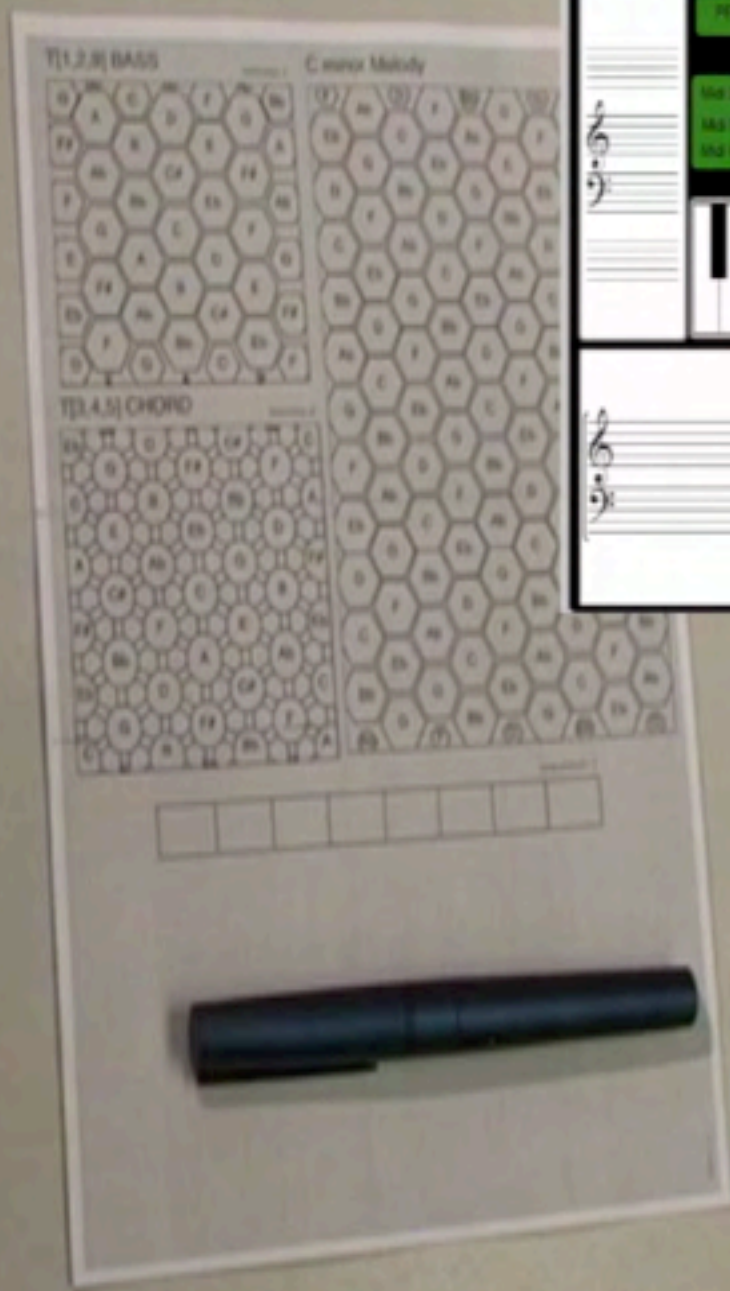
# PaperTonnetz

## Supporting Music Composition with Interactive Paper

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

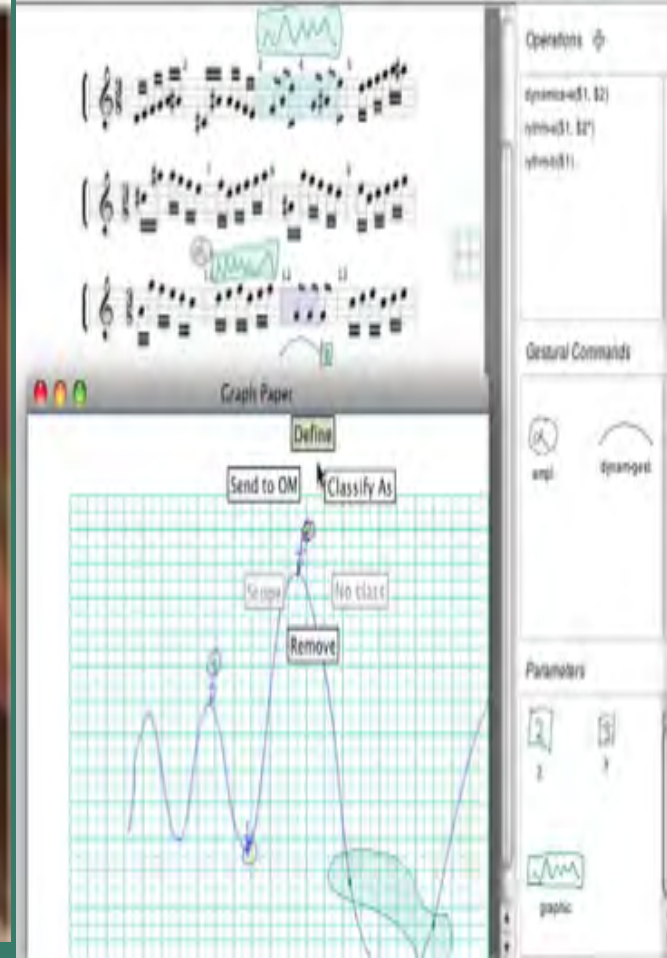
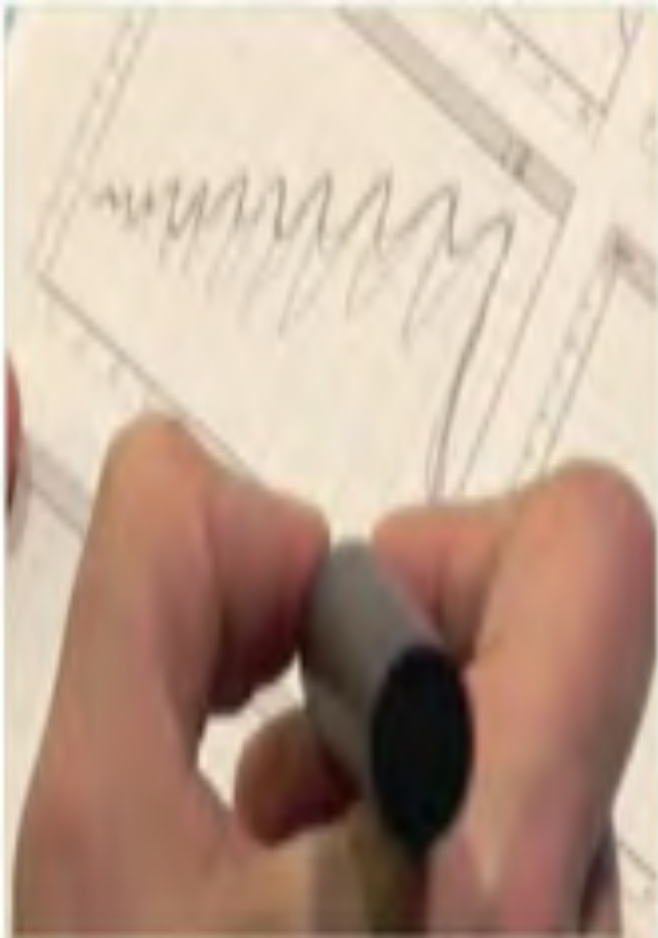
INRIA, IRCAM, LACL



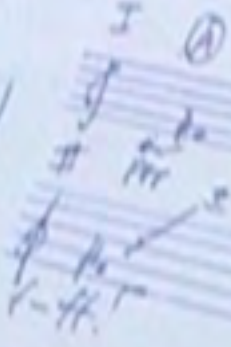
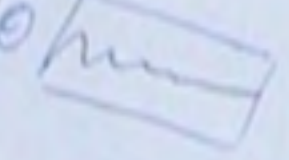
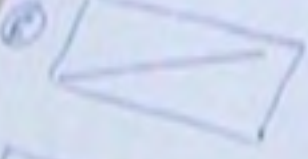


Paper Substrates

Composer create their own  
reusable musical structures



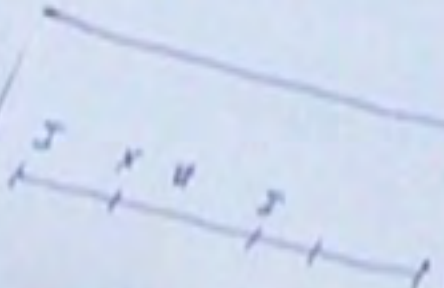
13



RULE

#1 follow a bpf

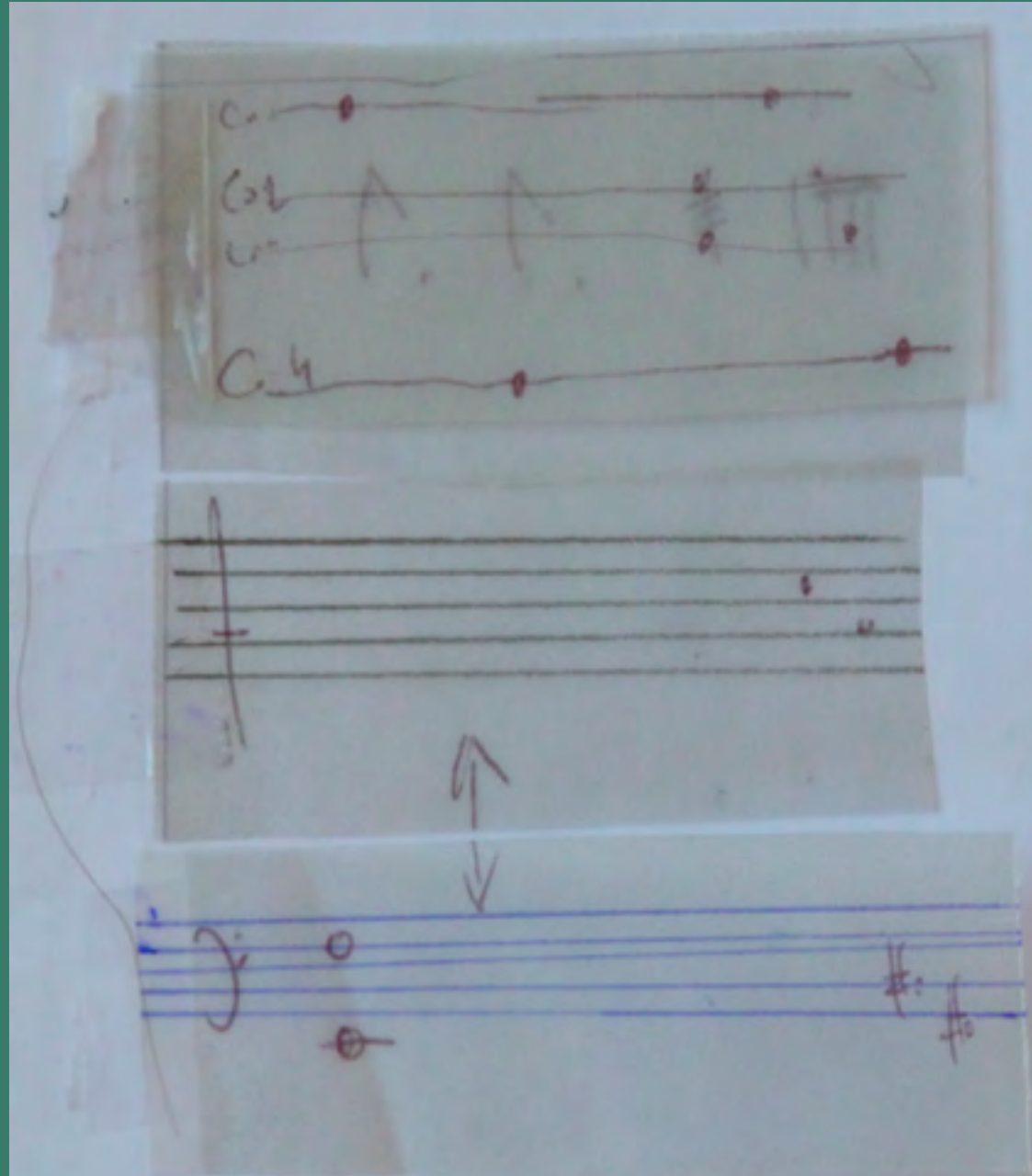
establish relationships among them



the global set-class

Hand-drawn musical notation on a staff with notes and stems.

Arrange  
and  
Link  
substrates





Arrange  
and  
Link  
substrates

to

composition  
software

The screenshot displays a music composition software interface. At the top, three staves of musical notation are shown, with handwritten annotations including circled numbers (1-13) and a green waveform graphic. Below the notation is a window titled "Graph Paper" containing a grid with a blue waveform and several labeled boxes: "Define", "Send to OM", "Classify As", "Scope", "Remove", and "No class". To the right of the graphing area is a vertical sidebar with three sections: "Operations" containing "dynamics-w(\$1, \$2)", "rhythm-a(\$1, \$2)", and "rhythm-b(\$1)"; "Gestural Commands" containing "amp" and "dynam-gest" with corresponding icons; and "Parameters" containing "2" and "3" with icons, and "graphic" with a waveform icon.

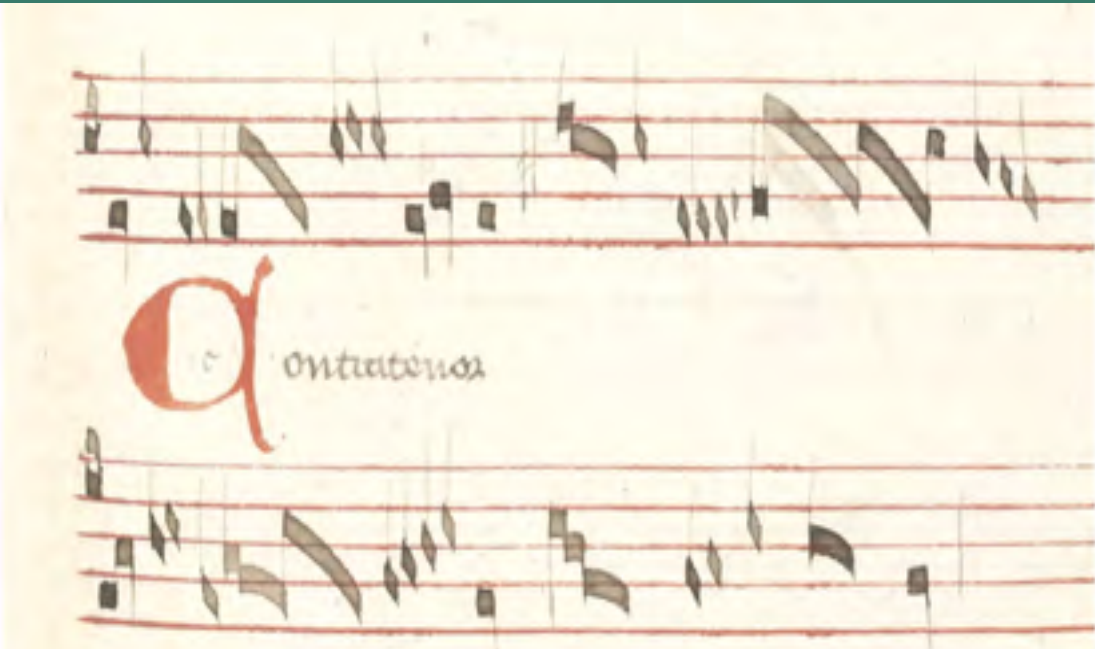
# **Interactive Paper Substrates to Support Musical Creation**

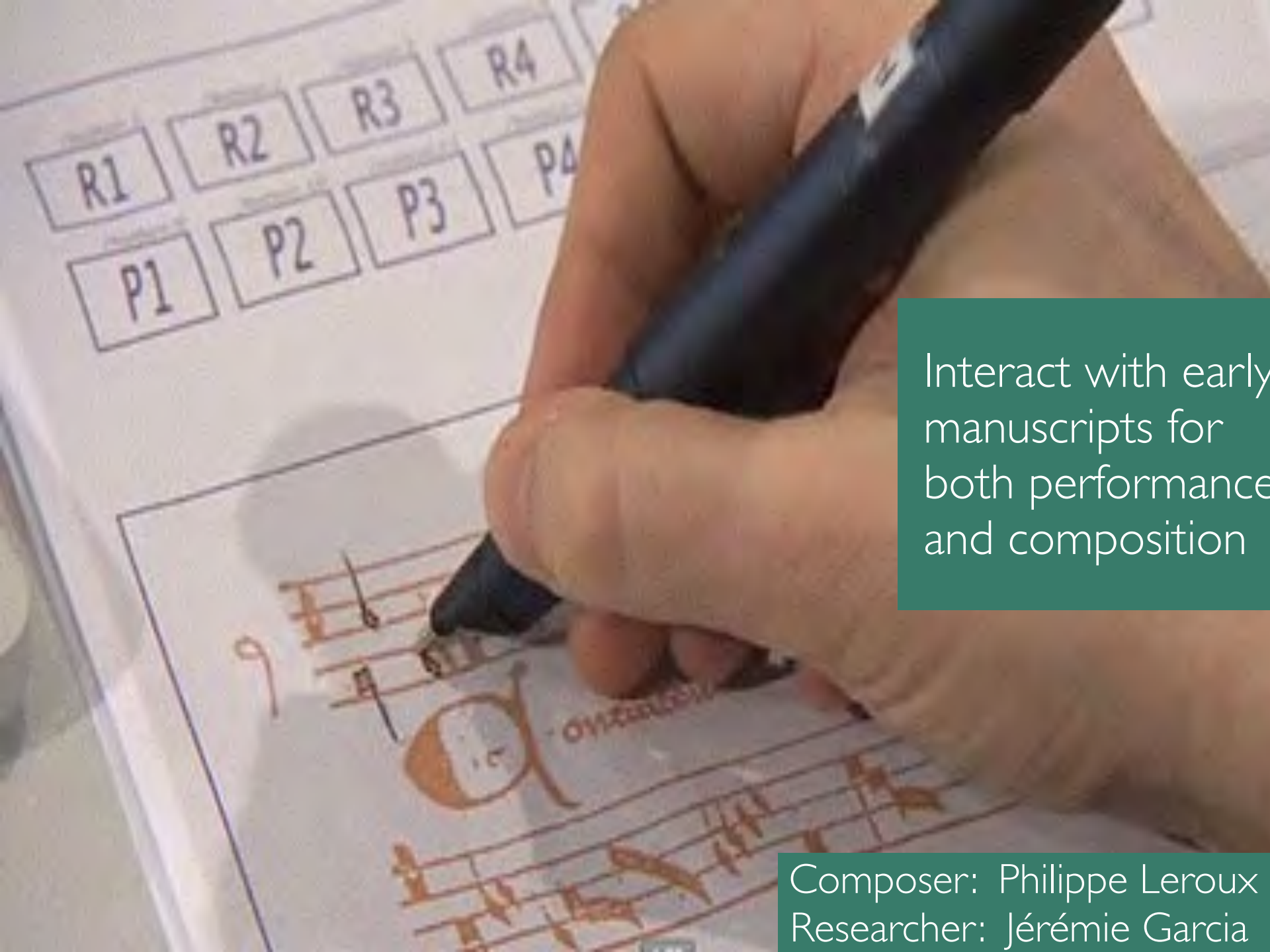
**Jérémie Garcia, Theophanis Tsandilas, Carlos Agon & Wendy E. Mackay**

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

Quid Sit Musicus  
Philippe Leroux

13<sup>th</sup> 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

Computer as *servant*  
Delegate tasks

Computer as *medium*  
Communicate



Human-  
Computer  
Interaction

Artificial  
Intelligence

Mediated  
Communication

# Human-Computer Partnerships

*Human-in-the-loop*

# A 'simple' human-computer partnership

User types – Google suggests – User chooses



The image shows a screenshot of a Google search results page. The search bar at the top contains the word "google". Below the search bar, a list of suggestions is displayed: "google", "google maps", "google translate", "google earth", and "google images". The search results section shows "About 5,700,000,000 results (0.07 seconds)". The first result is for "Google", with a description: "Enables users to search the Web, Usenet, and images. Features include PageRank, caching and translation of results, and an option to find similar pages." The second result is for "Google Images", with a description: "The most comprehensive image search on the web." The third result is for "Google Maps", with a description: "Find local businesses, view maps and get driving directions in Google Maps." The fourth result is for "News for google", with a sub-heading "Google Goes Gaming With Search Puzzles" and a description: "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 ...".

**Google**

Everything  
Images  
Videos  
News  
Shopping  
Realtime  
More

San Francisco, CA  
Change location

Any time  
Latest  
Past 24 hours  
Past 2 days  
Past week  
Past month  
Past year  
Custom range...  
More search tools

google

google  
google maps  
google translate  
google earth  
google images

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

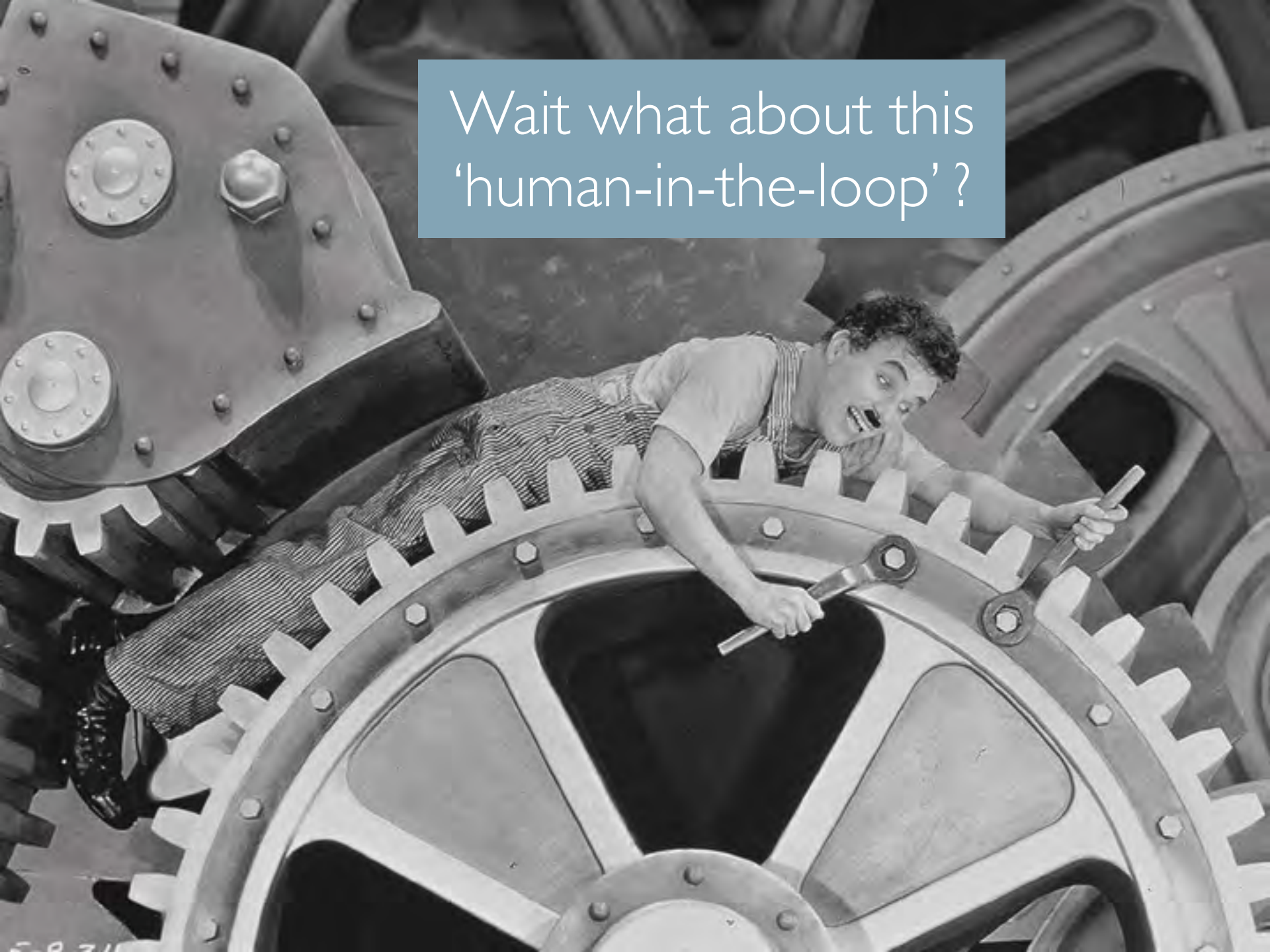
**Google**  
Enables users to search the Web, Usenet, and images. Features include PageRank, caching and translation of results, and an option to find similar pages. [Show stock quote for GOOG](#)  
[www.google.com/](#) - [Cached](#) - [Similar](#)

**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*

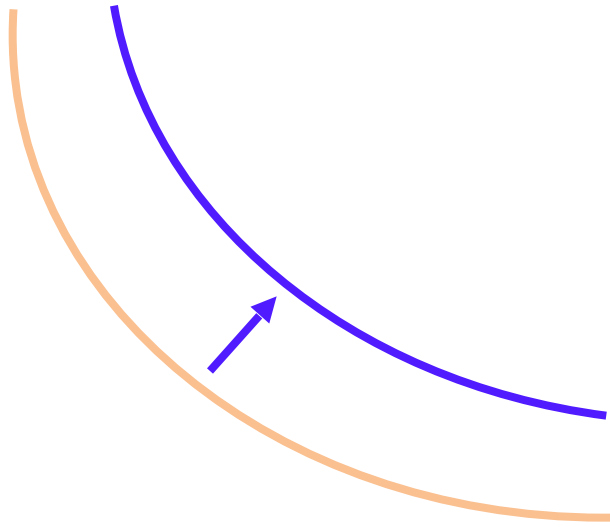
Human-computer interaction perspective:

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

# 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

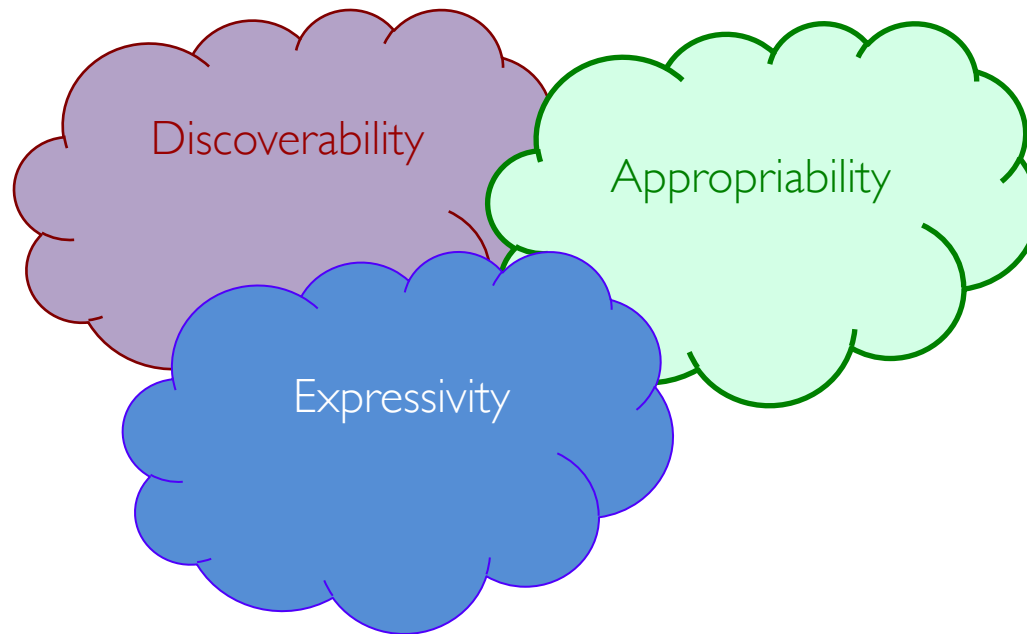
People can

*adapt to* technology

*adapt* the technology

they learn it

they appropriate it



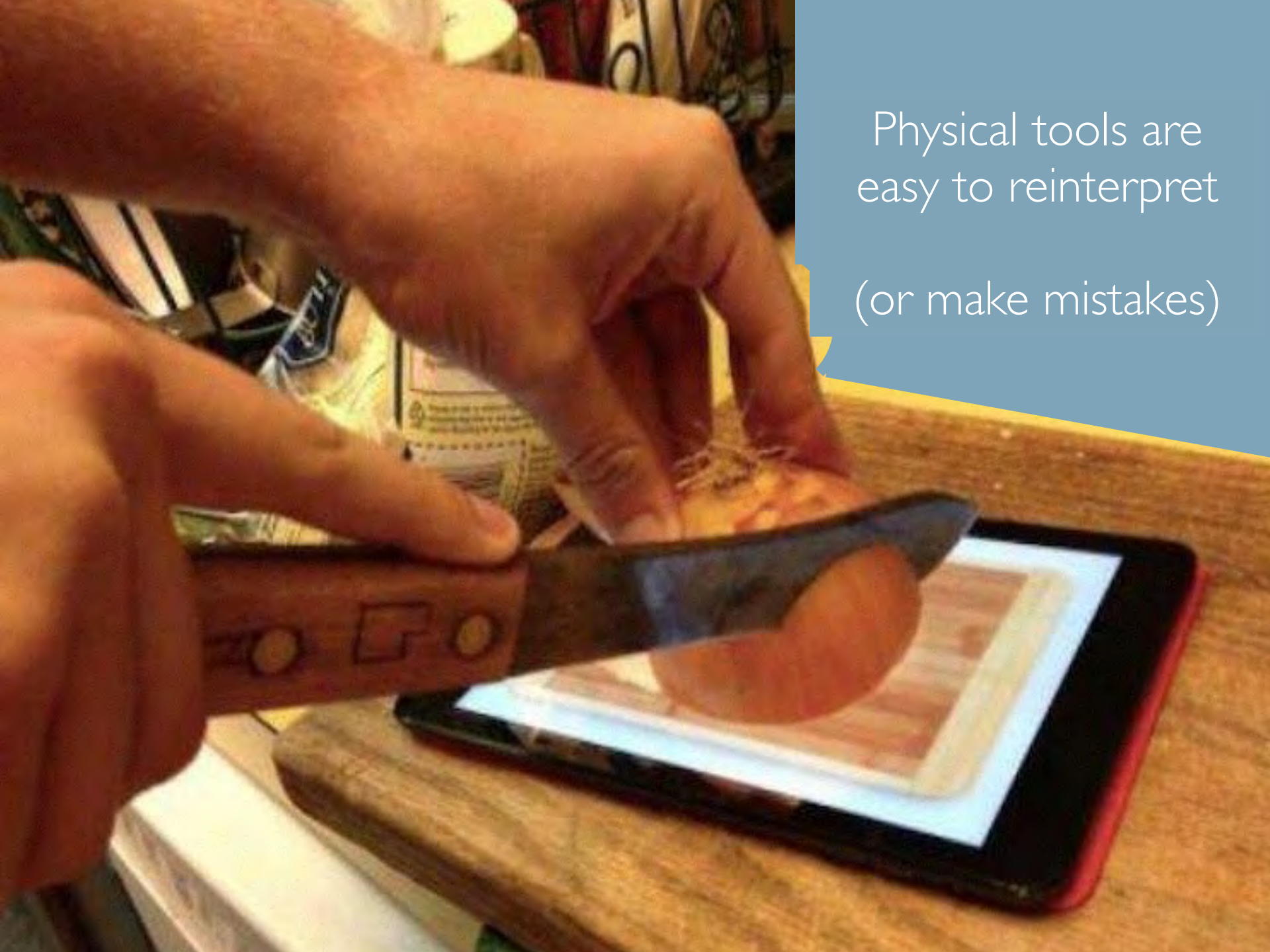
What can we learn from physical tools ?



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 !



A hominid with dark, shaggy fur is shown in a dynamic pose, holding a large bone vertically in its right hand as if about to strike. The hominid is surrounded by a pile of bones and a large animal skull on the ground. The background is a bright, cloudy sky. A semi-transparent blue box with white text is overlaid on the right side of the image.

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

*adapt* the technology

they learn it

they appropriate it

Computers can

*adapt to* people

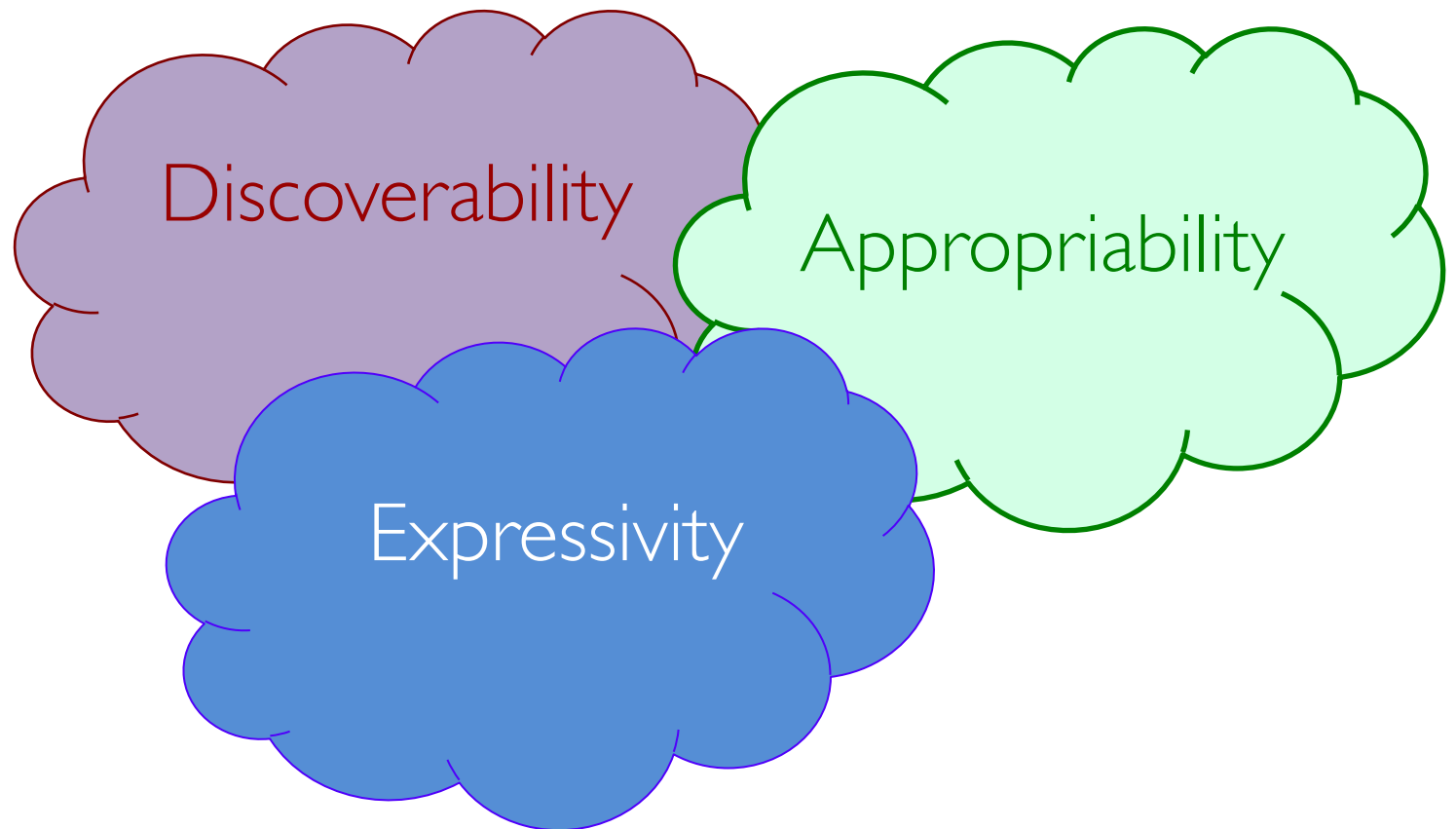
*adapt* people's behavior

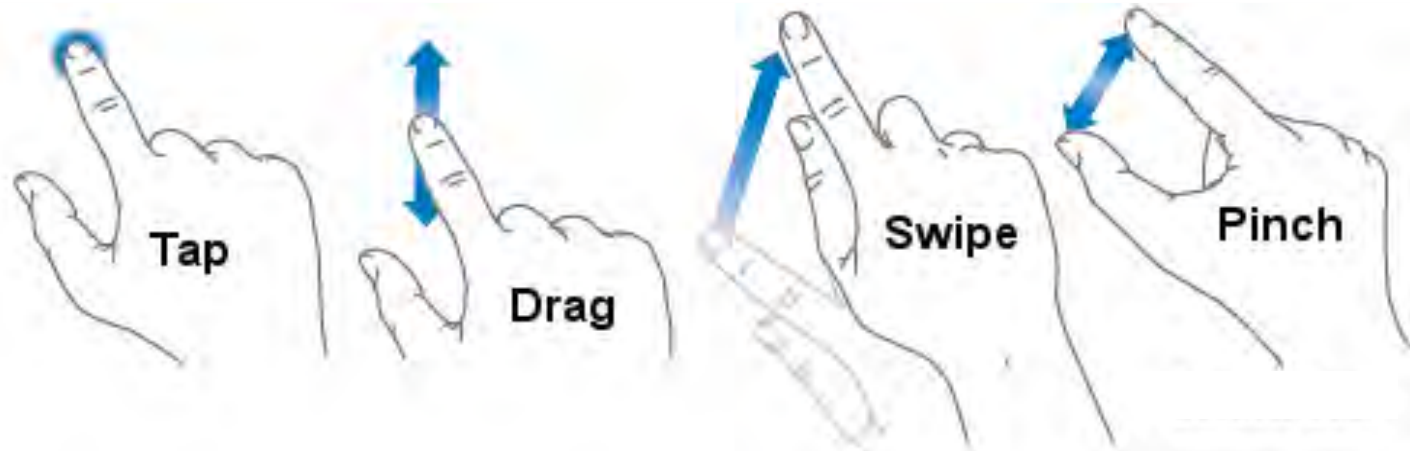
they learn (AI)

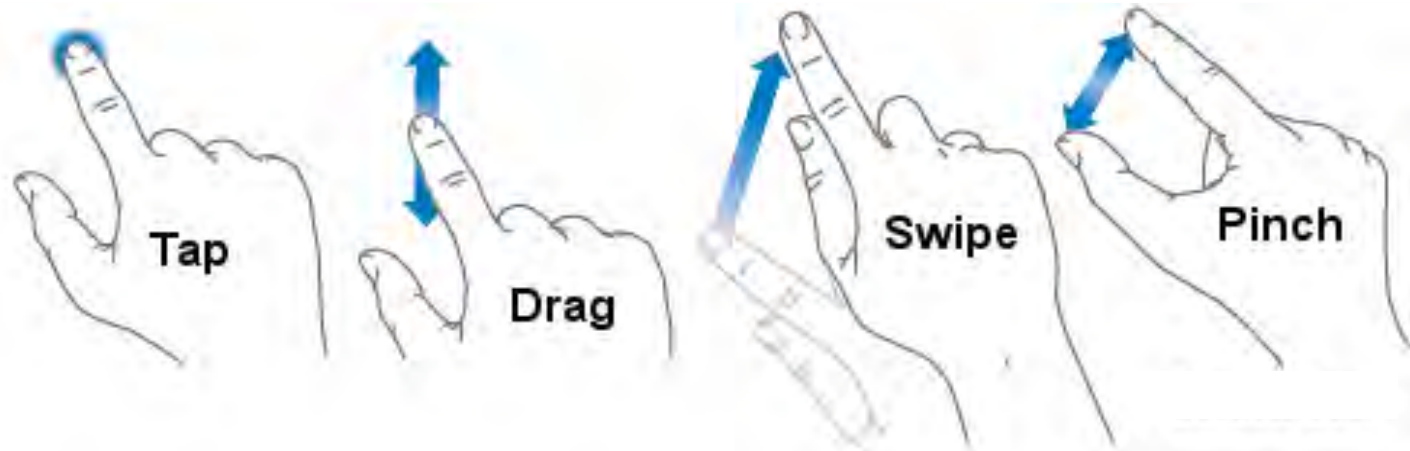
they teach

# Human-Computer Partnerships

People seek:









Discoverability

How can I learn  
new gestures  
and commands ?



# Octopocus

Learn gestures that issue commands

Progressive feedforward

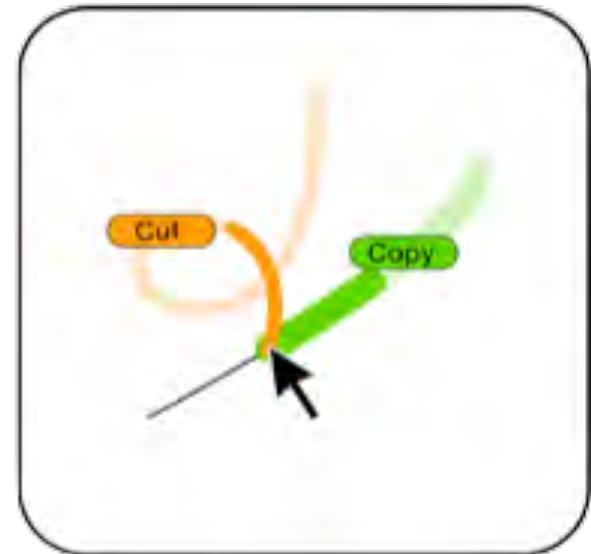
*What gestures are available ?*

Progressive feedback

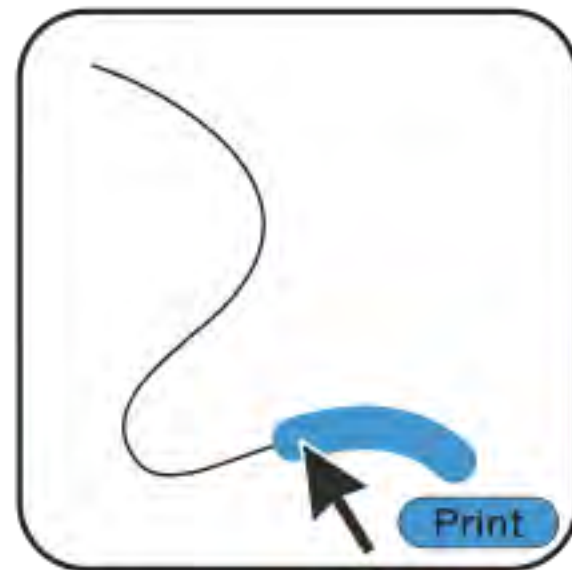
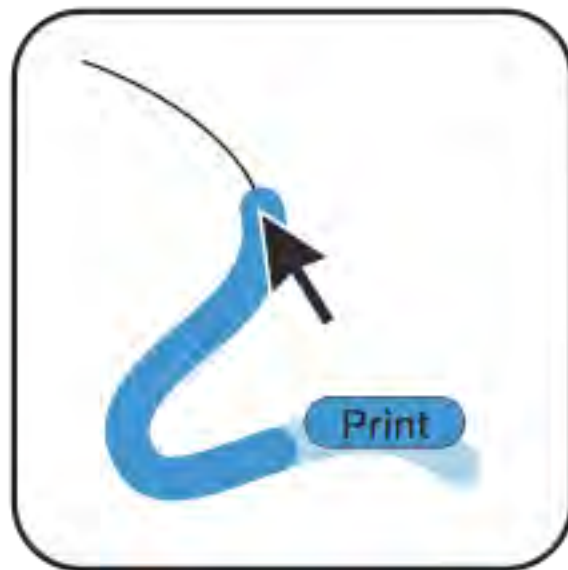
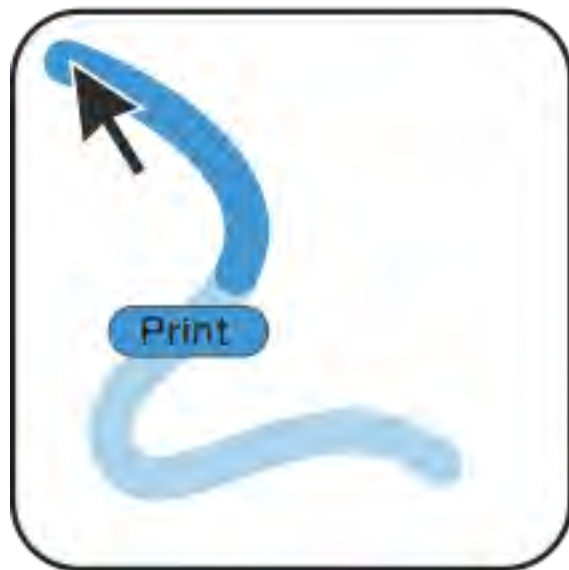
*What did the system recognize ?*

Novices pause ...  
the guide appears

Experts just do it



# Dynamic partnership



Inking the *'Help'* command



Appropriability

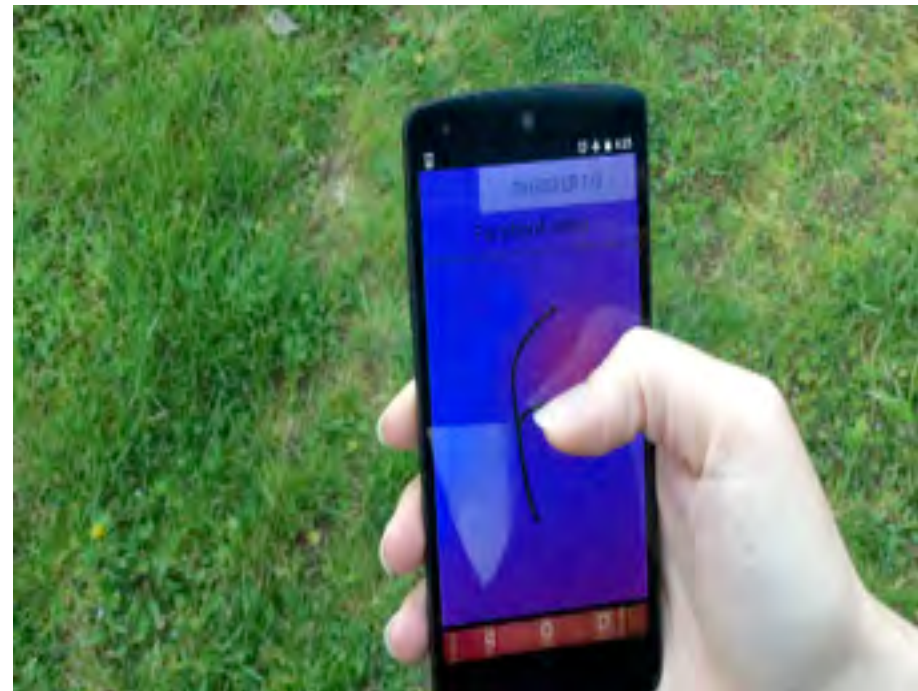
How can I define  
my own gestures ?

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

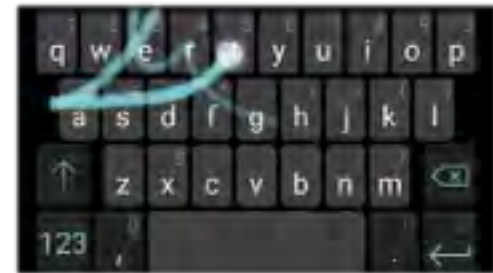
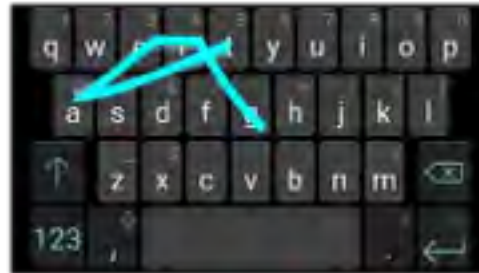
How can I  
express myself ?



Redefining gesture-typing keyboards  
to support user expression

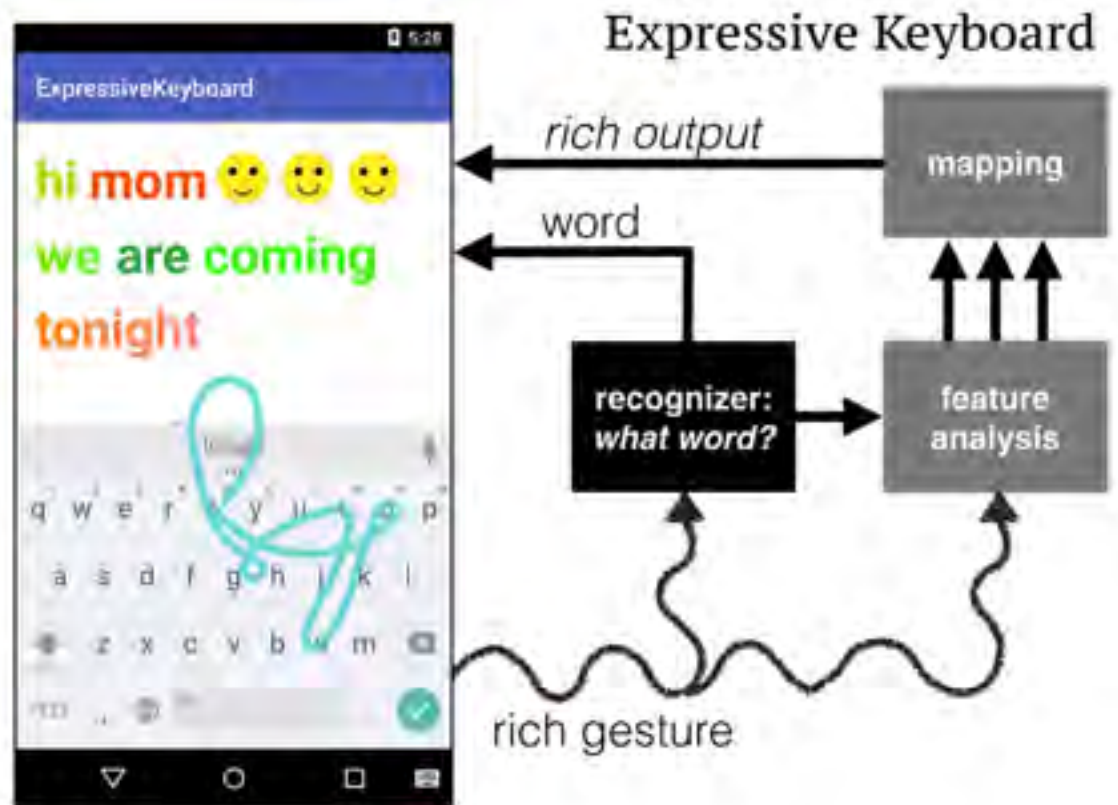
Four ways to type the word “great”

each produces the same result



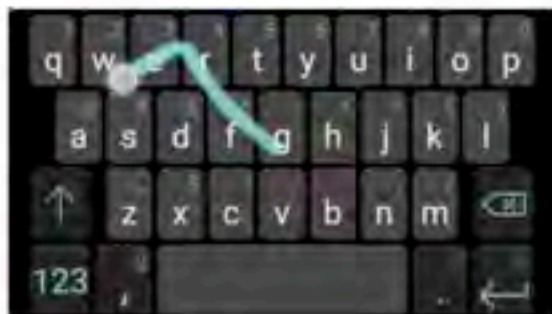
Machine learning guesses the correct word  
Gesture variations creates expressive text

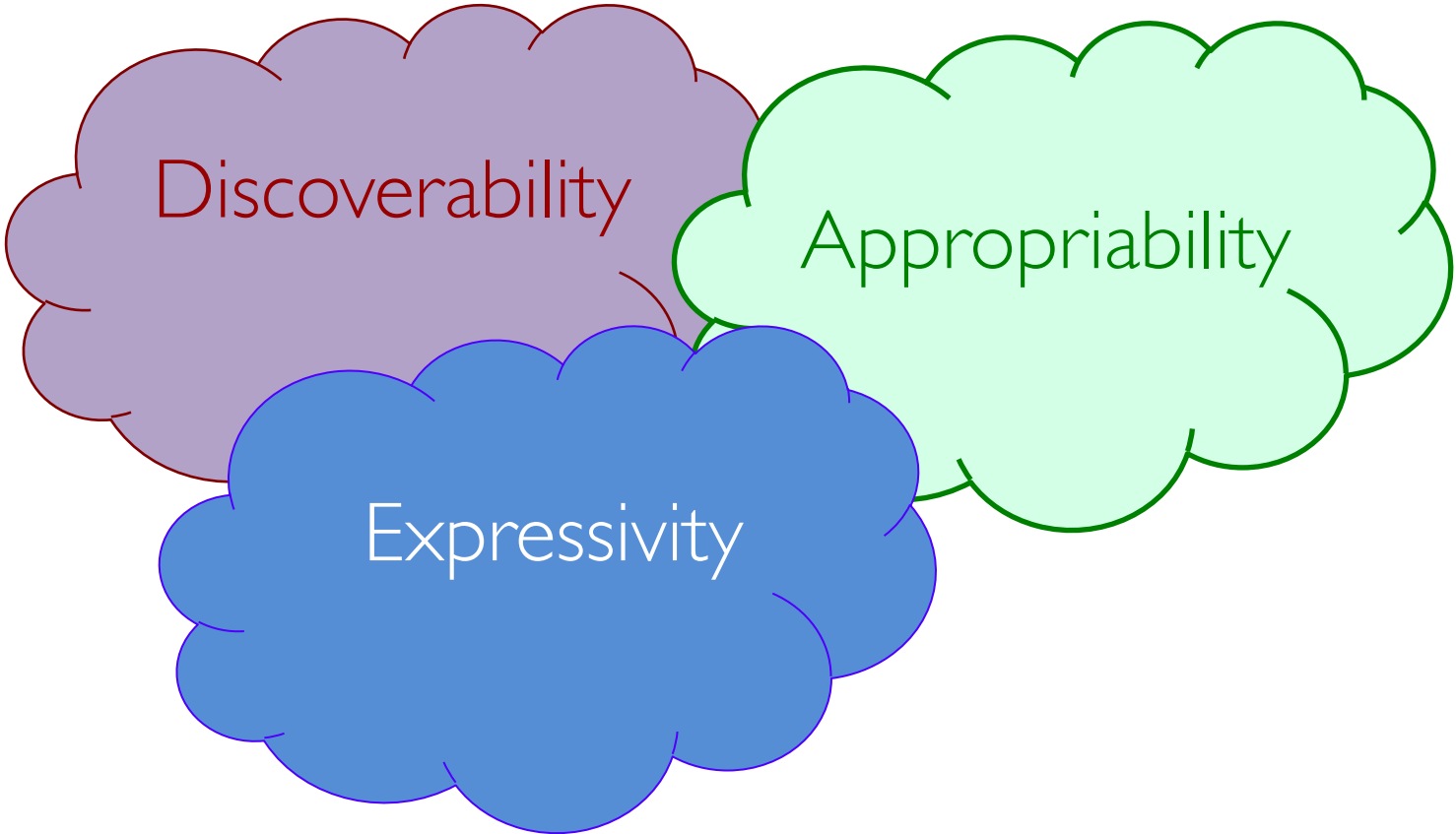
Users control  
text color  
font style  
and emojis



# Expressive Keyboard

The recognition algorithm is highly tolerant of input variation ... these are all "great"!



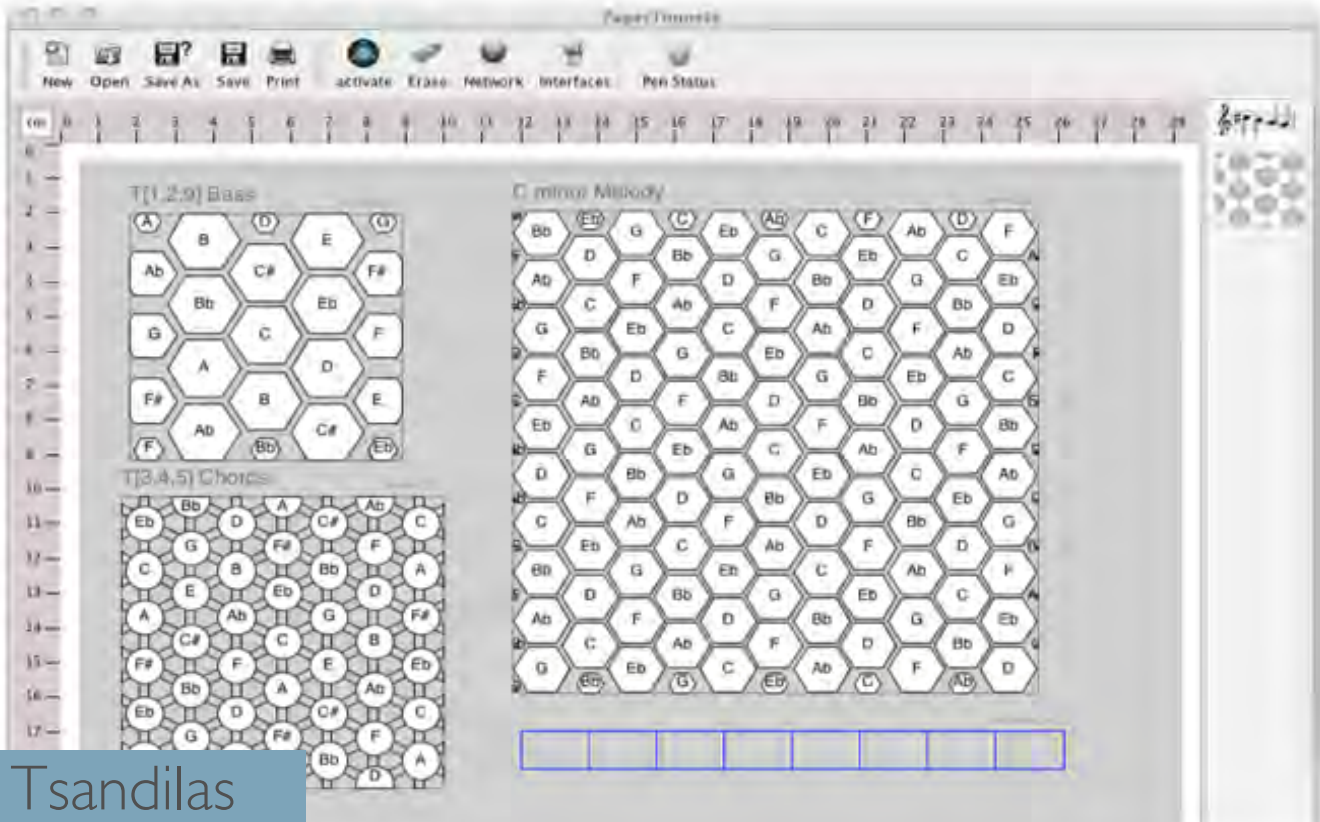




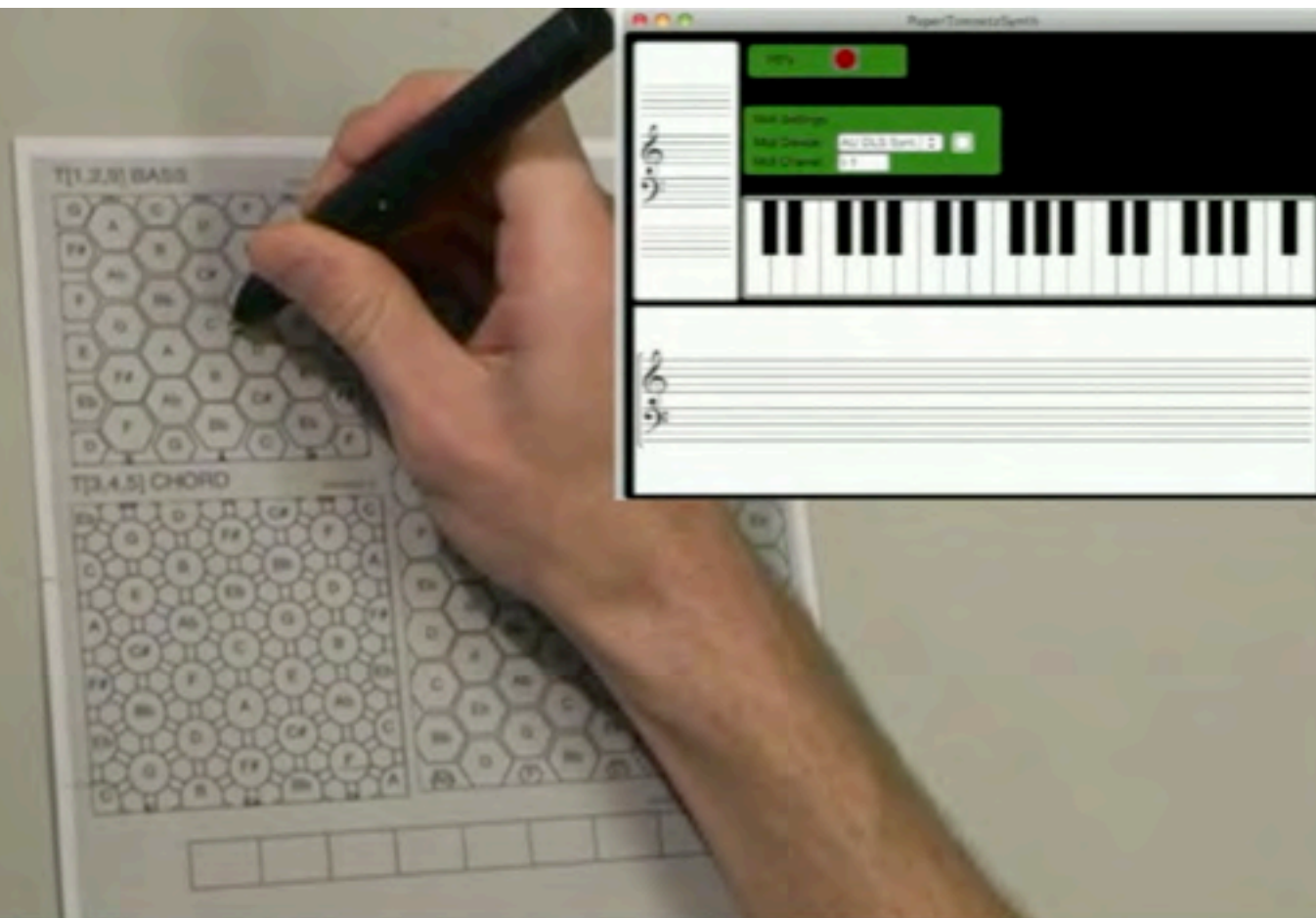
Discoverability

How can I learn  
musical relationships ?

# Paper Tonnetz



# Paper Tonnetz



The composer starts with the bass part





Appropriability

How can I define  
my own commands ?

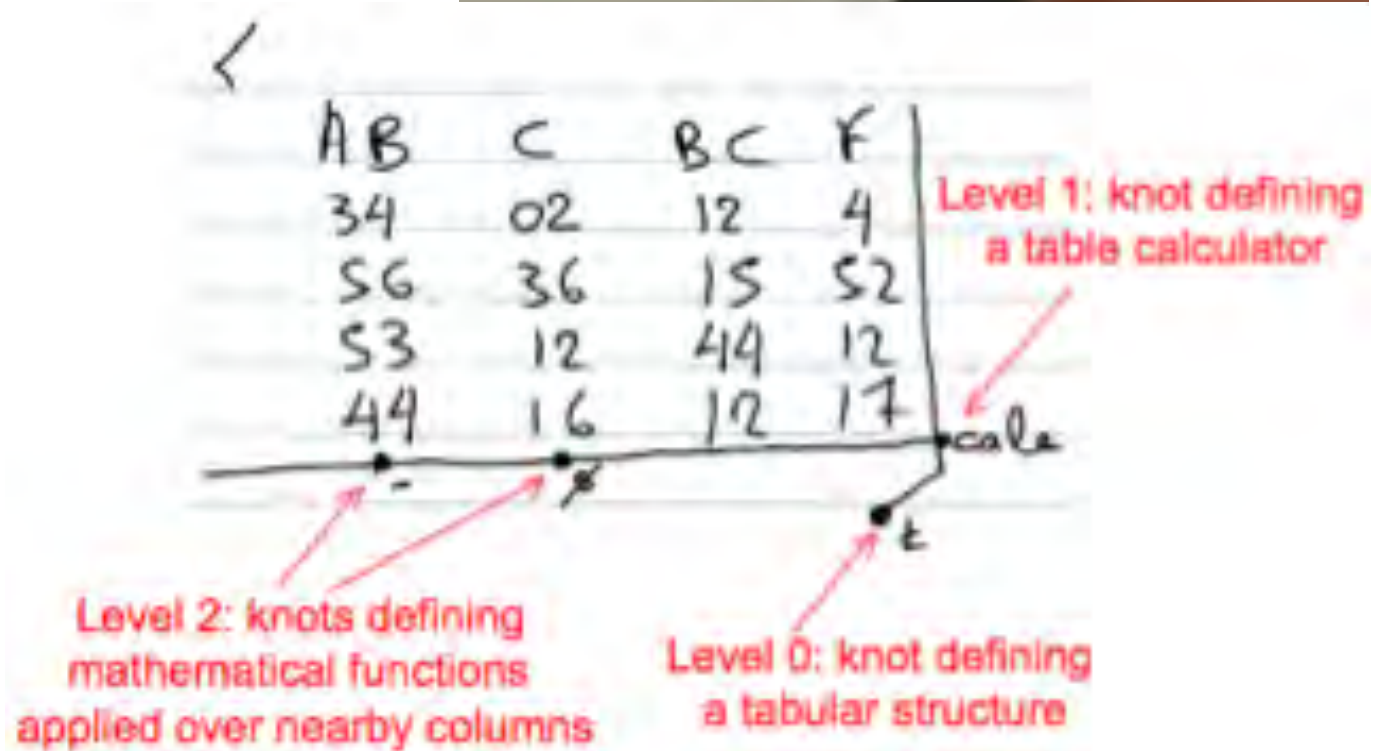
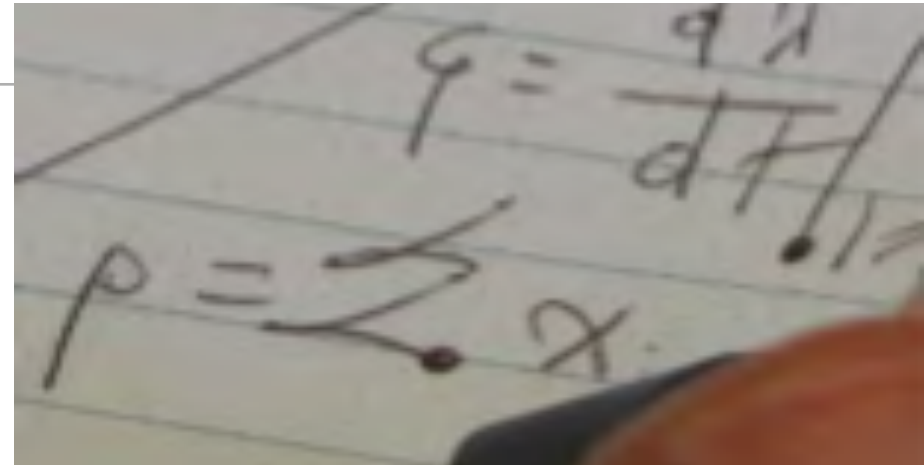
# Knotty Gestures:

AVI'10

Draw a knot to define a command  
Interact now or later



Knots can define mathematical or other relationships



# Knotty Gestures

Draw a line with a knot

Choose “recording” to define the type of line



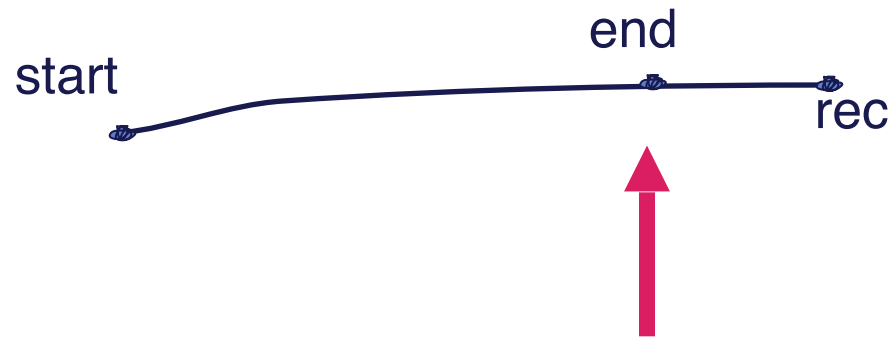
# Knotty Gestures

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



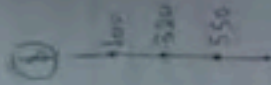


$$\sqrt{53} + \sqrt{52} = 11,34$$

$$\cos 9 + \cos 9 = 2,516$$

$$\sqrt{53} - \sqrt{52}$$

$$\cos 9$$





Expressivity

How can I  
express myself ?

13th century musical scores  
Each note indicates expression



# Paper Composer

Associate gesture characteristics  
with sounds and features

compose  
& perform



***QUID SIT MUSICUS?***  
**BY PHILIPPE LEROUX**