ECCS

Human-Computer Partnerships

Wendy E. Mackay

Inria, Université Paris-Saclay

11 October 2018

What kind of ‘partnership’ ?

Take a taxi

- Driver in control

What kind of 'partnership' ?

Take a taxi

Driver in control

Drive a motorcycle

User in control

What kind of 'partnership'?

Take a taxi

Driver in control

Drive a motorcycle

User in control

Ride a horse

Shared control

Towards generative theory

Define principles of a
unified theory of interaction

Instrumental Interaction

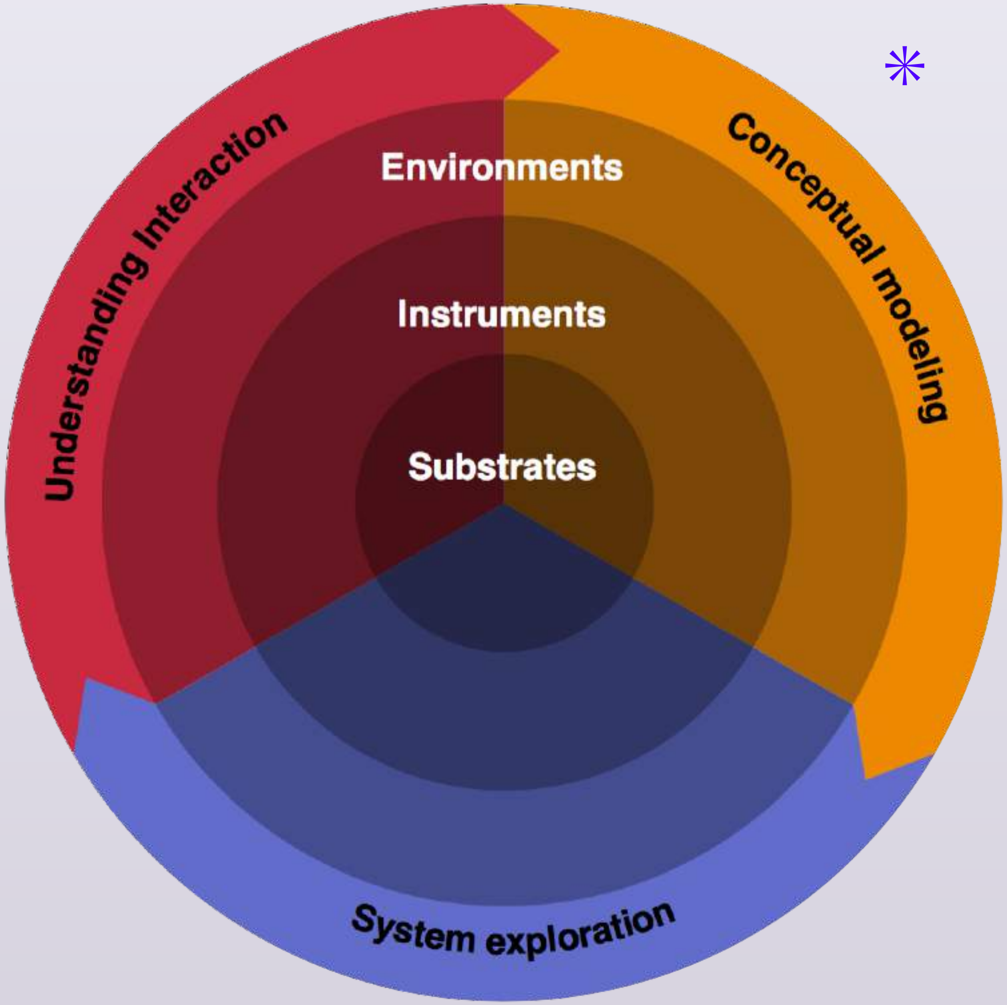
  Reification

  Polymorphism

  Reuse

Substrates

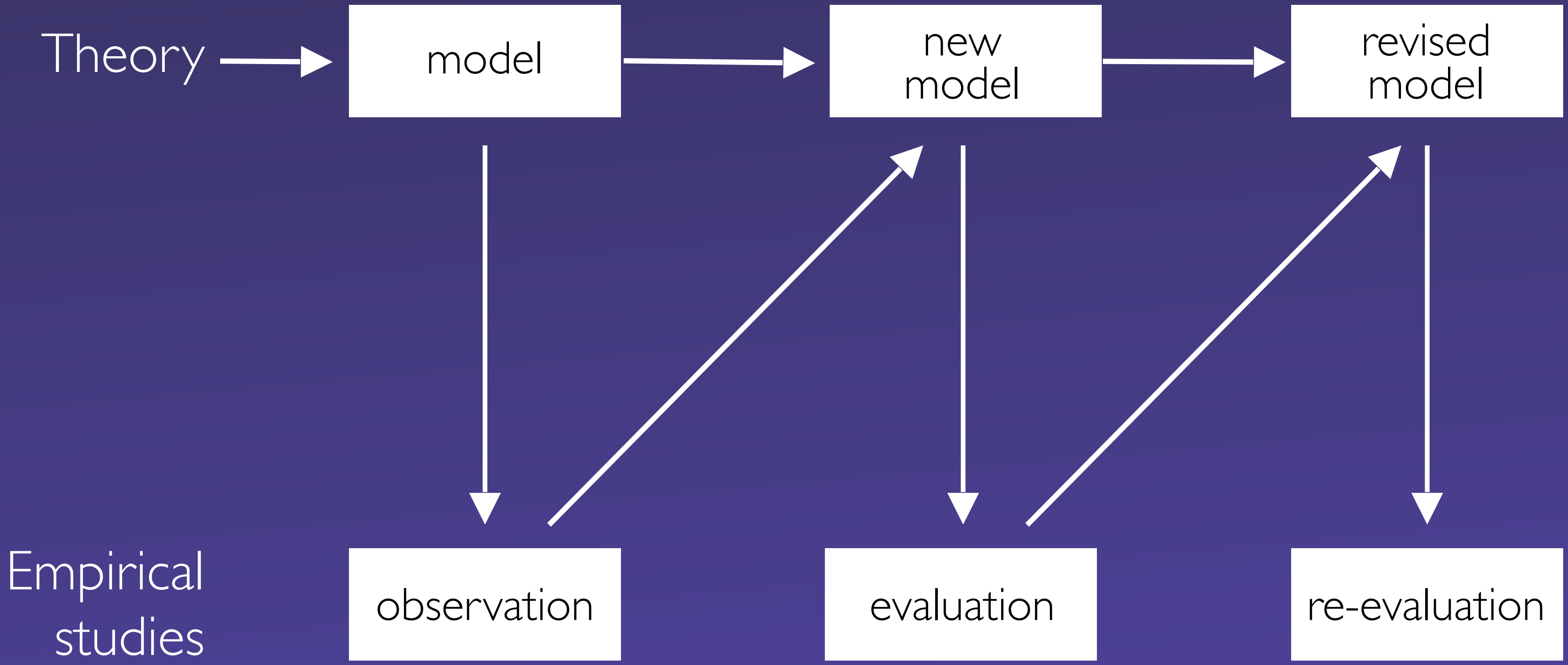
Reciprocal Co-Adaptation



*

* with Michel Beaudouin-Lafon

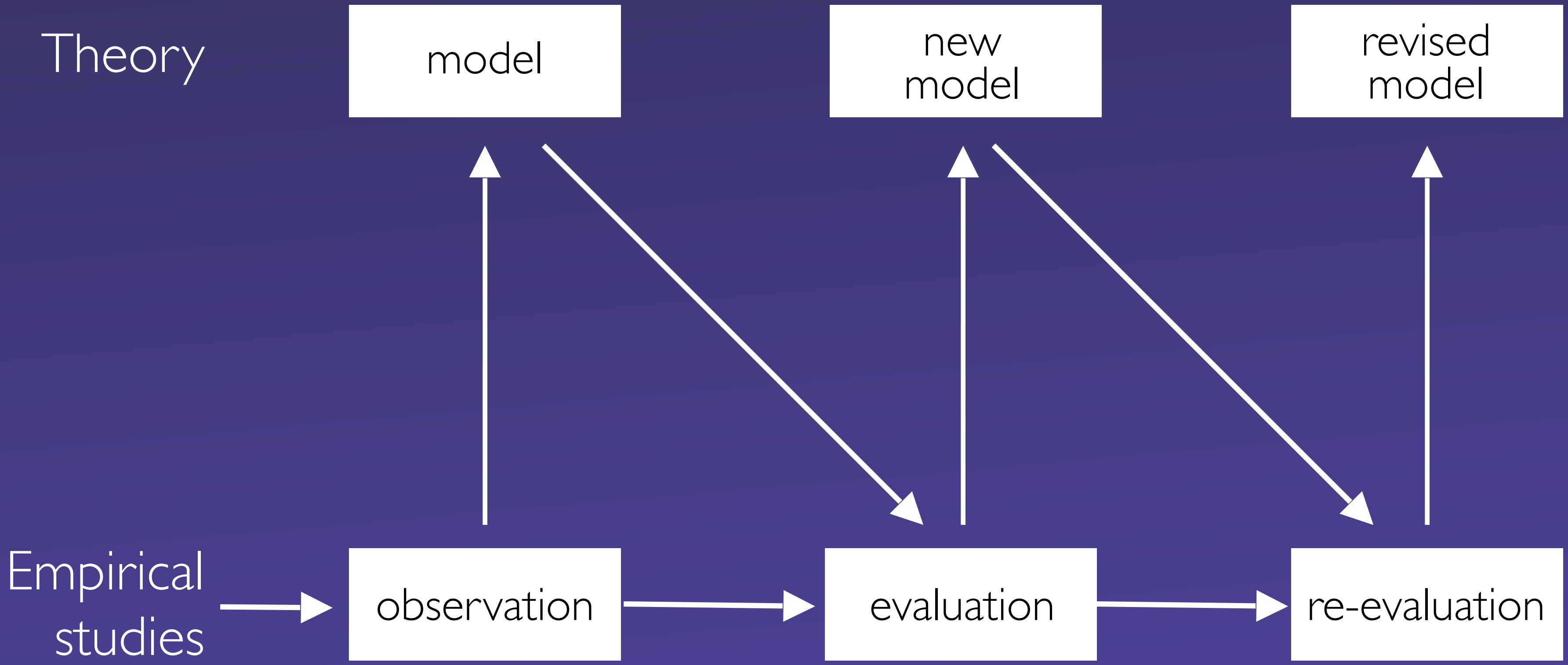
Natural Sciences: deduction



Mackay, W.E. and Fayard, A-L. (1997) ACM DIS'97

HCI, Natural Science and Design: A Framework for Triangulation Across Disciplines

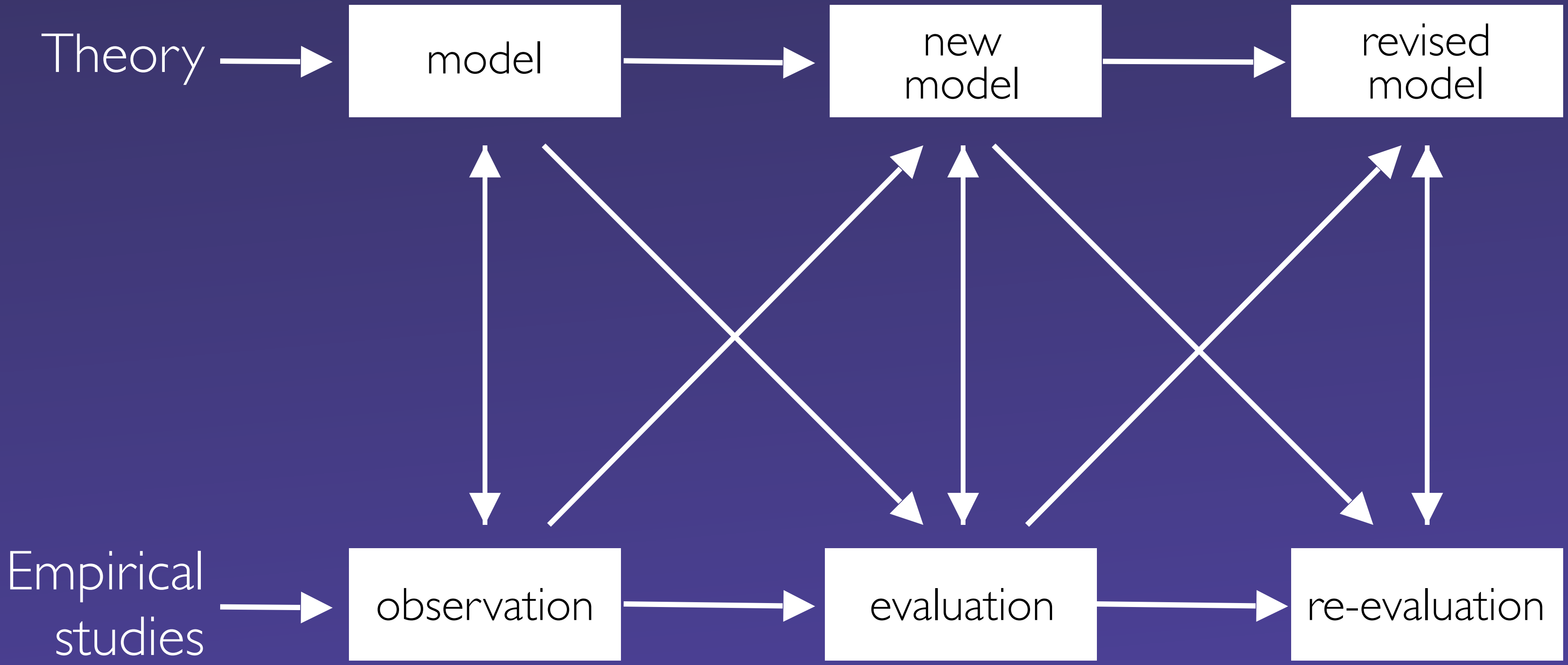
Natural Sciences: induction



Mackay, W.E. and Fayard, A-L. (1997) ACM DIS'97

HCI, Natural Science and Design: A Framework for Triangulation Across Disciplines

All natural sciences are cyclic



Theory

model

new model

revised model

Empirical studies

observation

evaluation

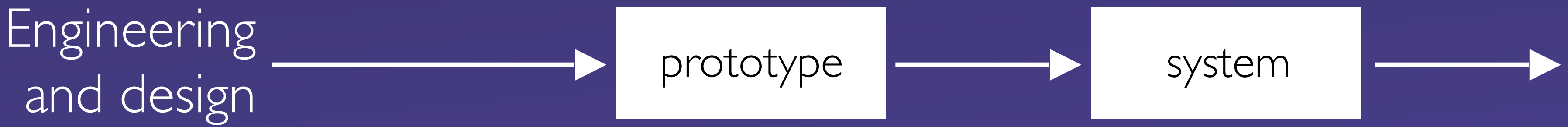
re-evaluation

Mackay, W.E. and Fayard, A-L. (1997)   ACM DIS'97

HCI, Natural Science and Design: A Framework for Triangulation Across Disciplines

What about engineering and design ?

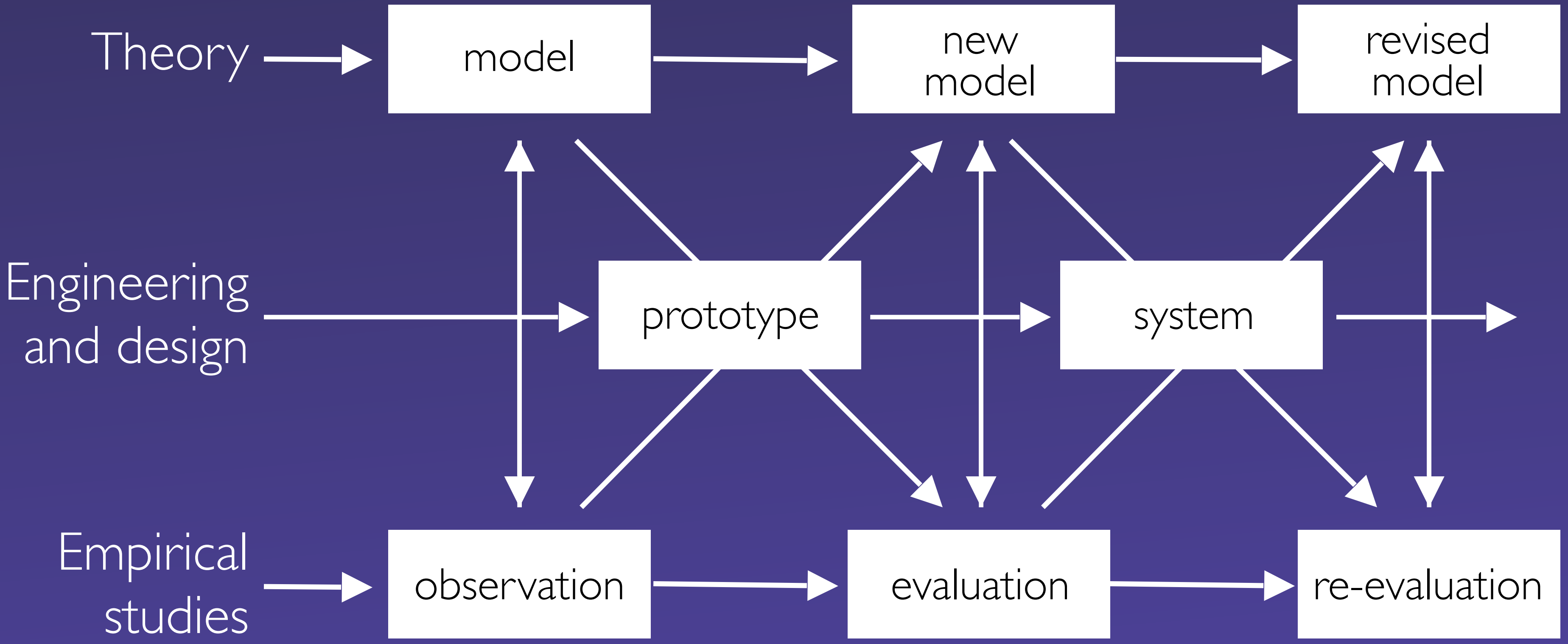
We study what we create



Mackay, W.E. and Fayard, A-L. (1997) ACM DIS'97

HCI, Natural Science and Design: A Framework for Triangulation Across Disciplines

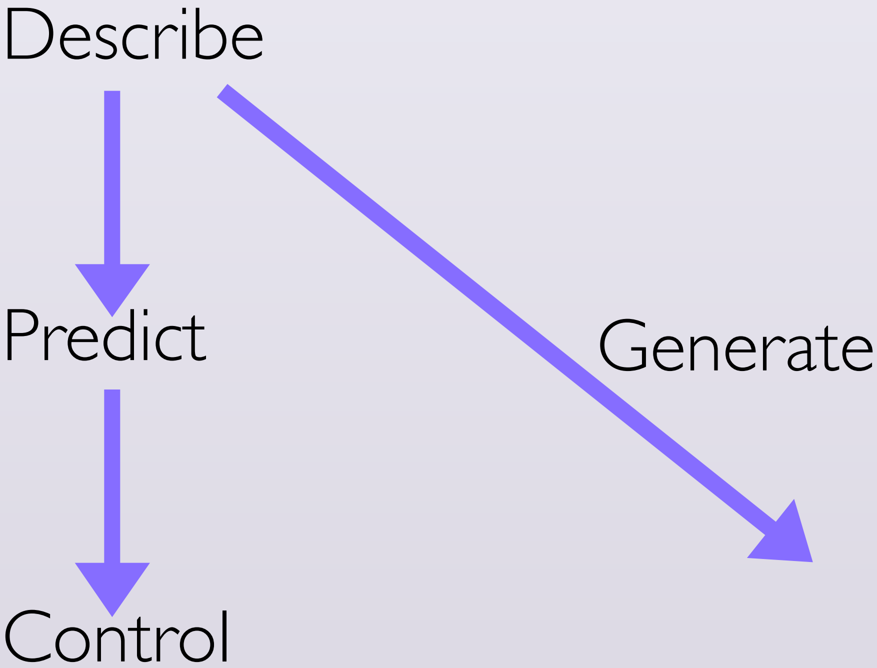
Multi-disciplinary research



Mackay, W.E. and Fayard, A-L. (1997)   ACM DIS'97

HCI, Natural Science and Design: A Framework for Triangulation Across Disciplines

Levels of theoretical power



Describe

Predict

Control

Generate

Theory, Empirical studies and Design

Natural Sciences:	Study a natural, existing phenomenon		
Deductive:	Theorical predictions	to	empirical verification
Inductive:	Empirical findings	to	theorical implications
Design:	Create a novel artifact		
Top-down:	Create architecture	then	build system
Bottom-up:	Design artifacts	then	derive architecture
HCI research:	Natural phenomena – *and* – designed artifacts		

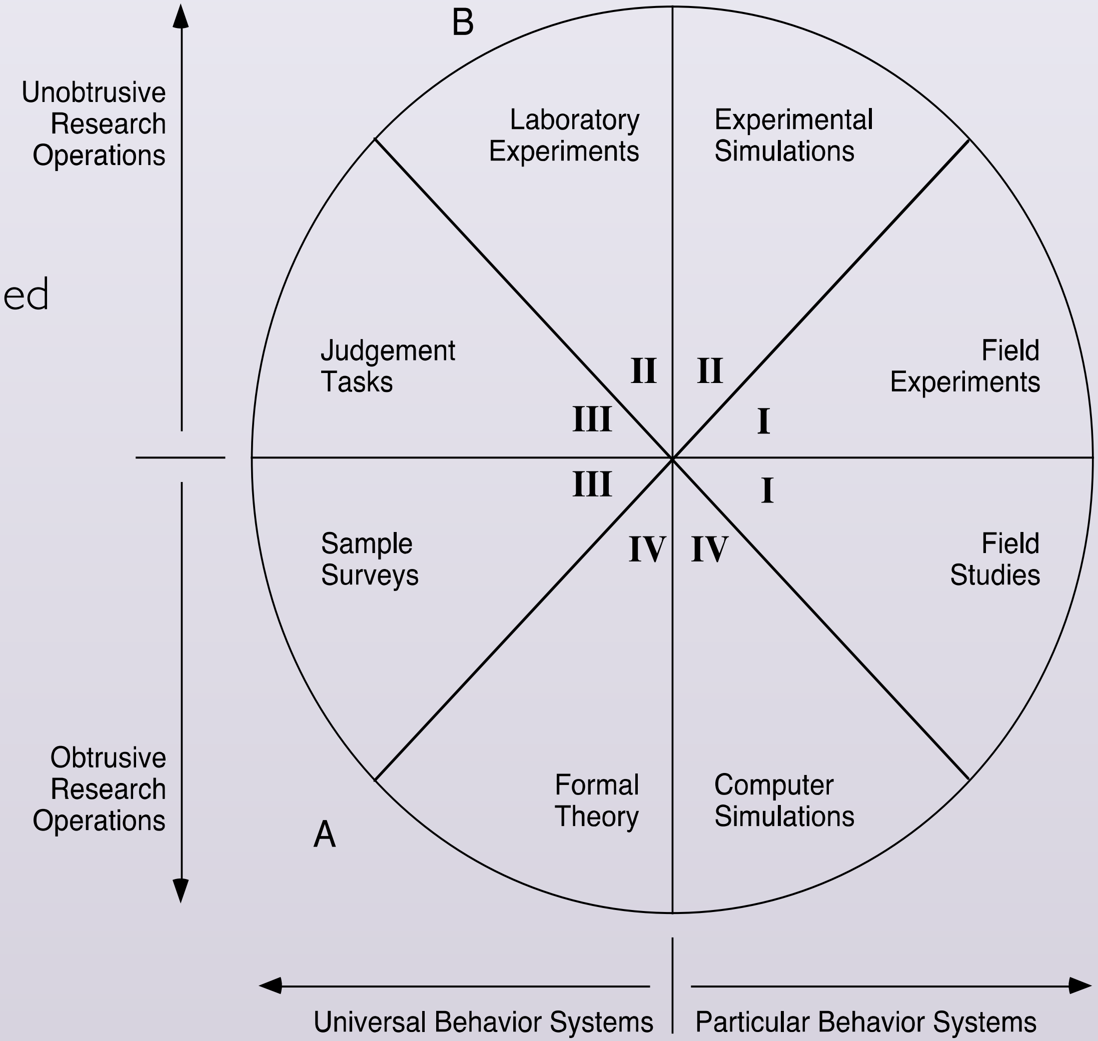
Methodology trade-offs

Types of settings:

I. Settings in natural systems
II. Contrived or created settings
III. Contrived or created settings
IV. No behavior observation needed

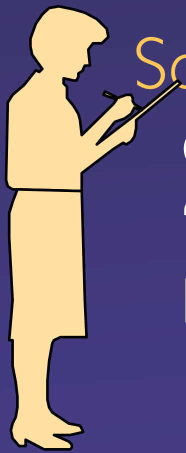
Major concern is:

A. Generality over actors
B. Precise measure of behavior
C. System character of context



Runkel & McGrath, 1972

Perspectives on understanding users



Scientific perspective

Collect data about users

‘Objective’ analysis

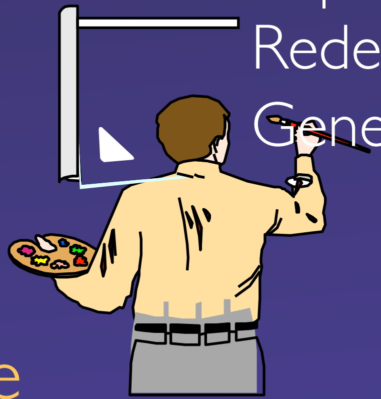
Inform designers

Design perspective

Inspire ideas

Redefine problem

Generate innovations

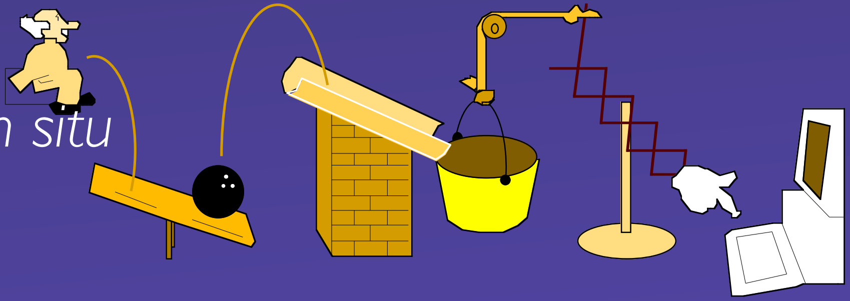


Engineering perspective

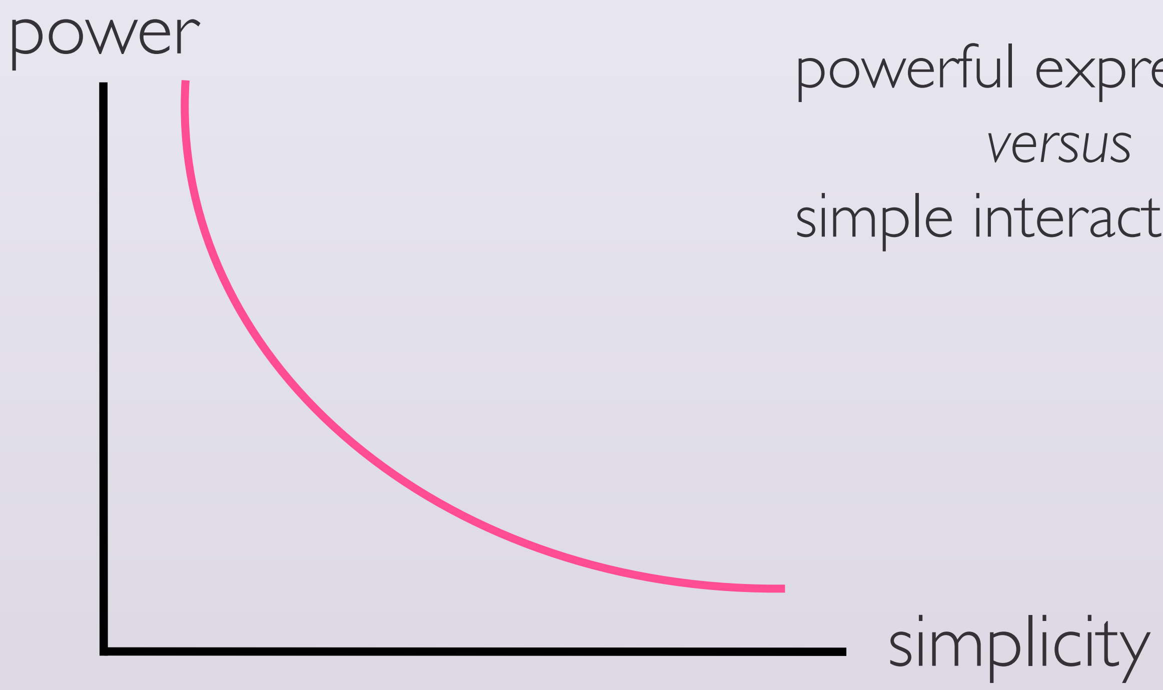
Address a given problem

Make trade-offs

Ensure it works *in situ*



HCI Design Trade-offs



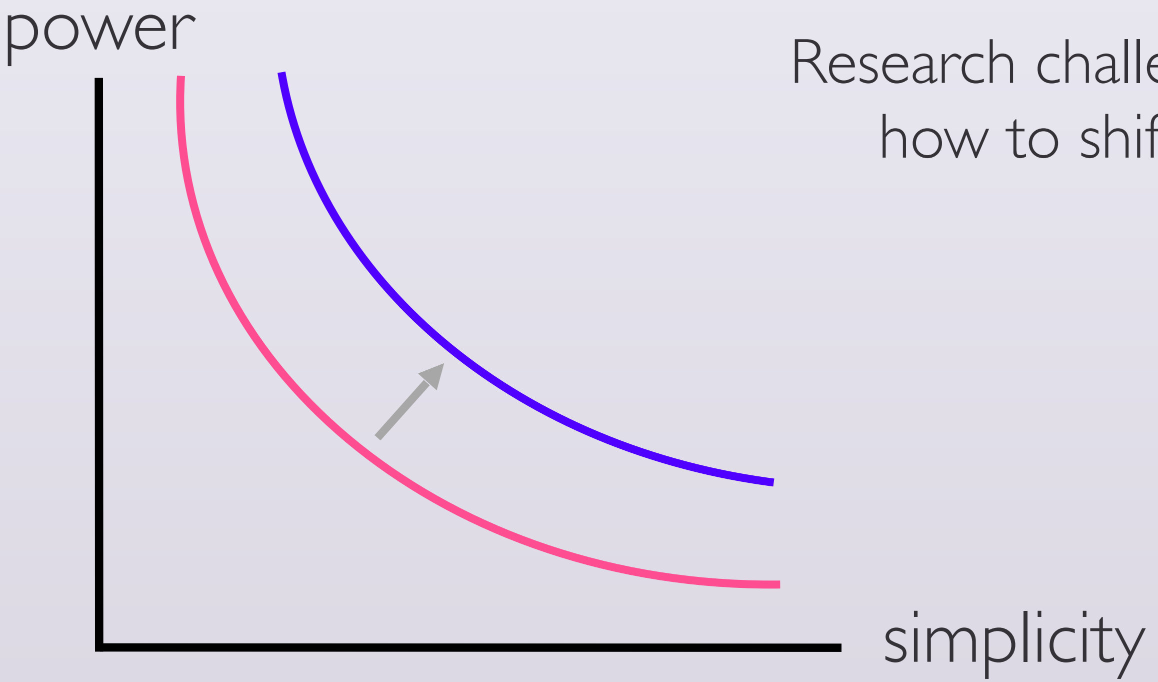
powerful expression

versus

simple interaction

Simple things should be simple,
complex things should be possible

HCI Design Trade-offs



Research challenge:
how to shift the curve?

Towards generative theory

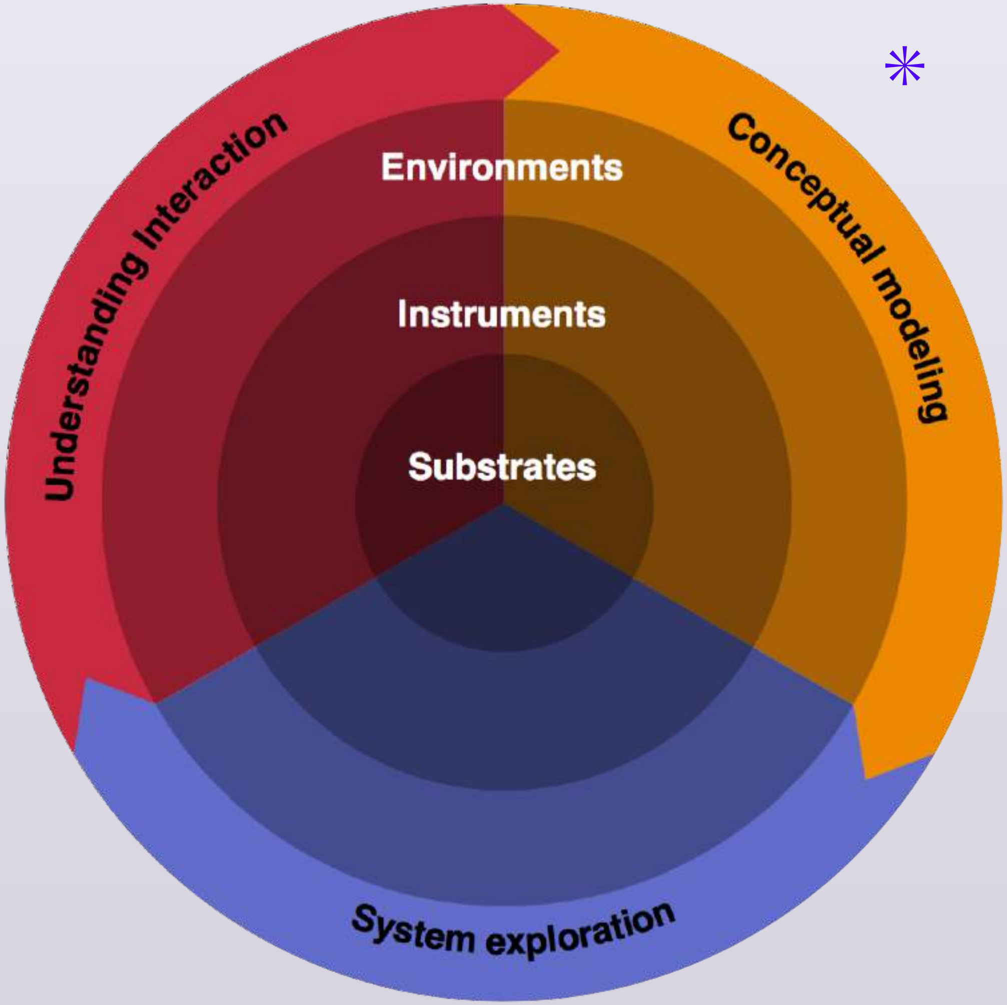
Define principles of a
unified theory of interaction

Instrumental Interaction

- Reification
- Polymorphism
- Reuse

Substrates

Reciprocal Co-Adaptation



*

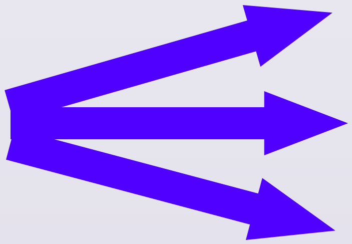
* with Michel Beaudouin-Lafon

Generative power: Three design principles

Reification

extends the notion of

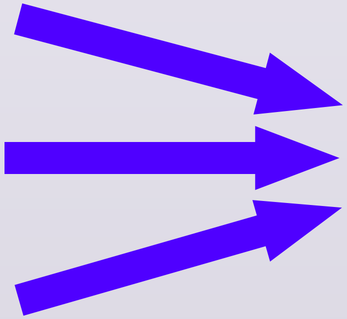
what constitutes an object



Polymorphism

extends the power of commands

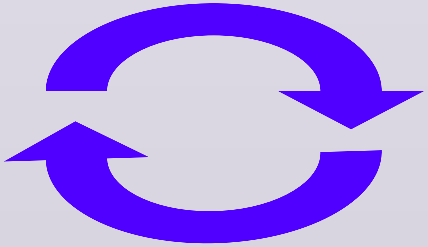
with respect to these objects



Reuse

provides a way of capturing and

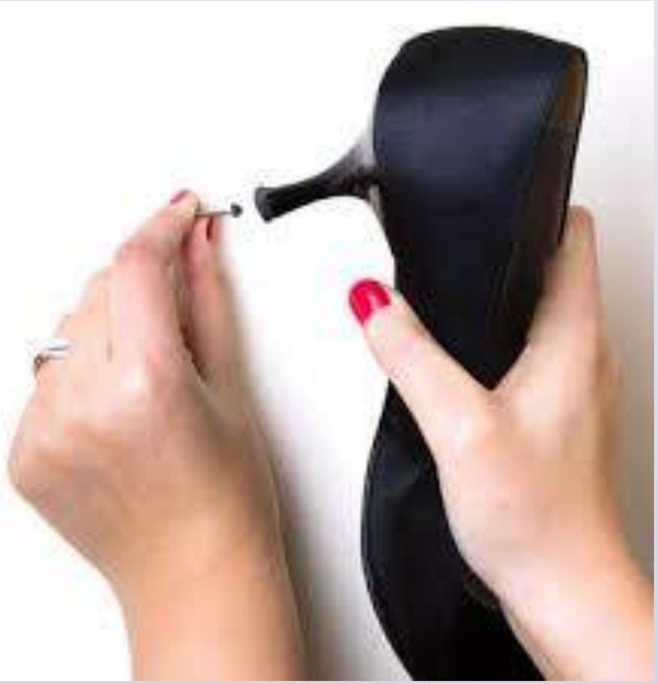
reusing interaction patterns



Physical tools have affordances

Physical tools have affordances

we can improvise ...



Physical tools have affordances

we can improvise ...

Physical affordances

any object can become an instrument

any instrument can solve multiple problems

Why isn't software like this ?

Our relationships with tools

Physical tools: follow the laws of physics

users can easily learn them

users can appropriate them

Computer tools: follow the whims of programmers

users must learn and relearn them

users easily break them

Goal: make **interaction** a first-class computational object

Software tools

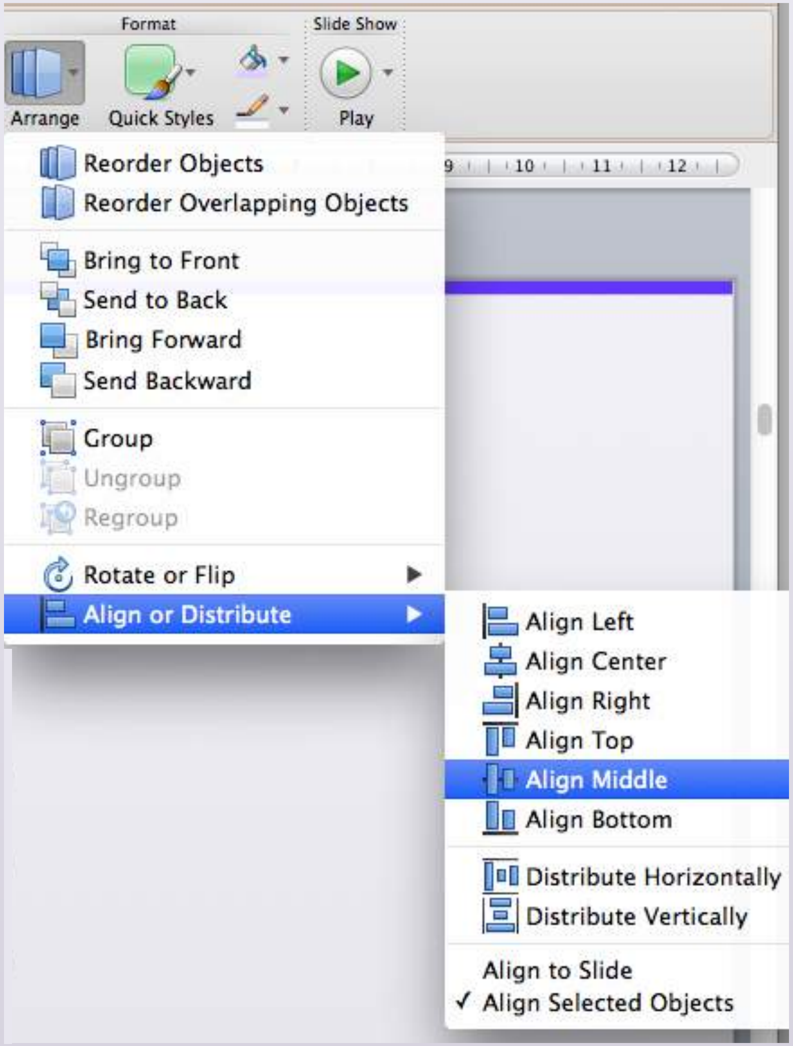
Example: Powerpoint

Alignment and distribution

=

Cumbersome buttons

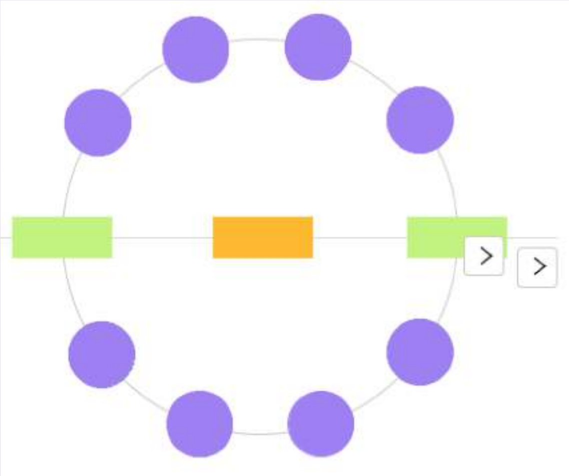
and pull-down menus

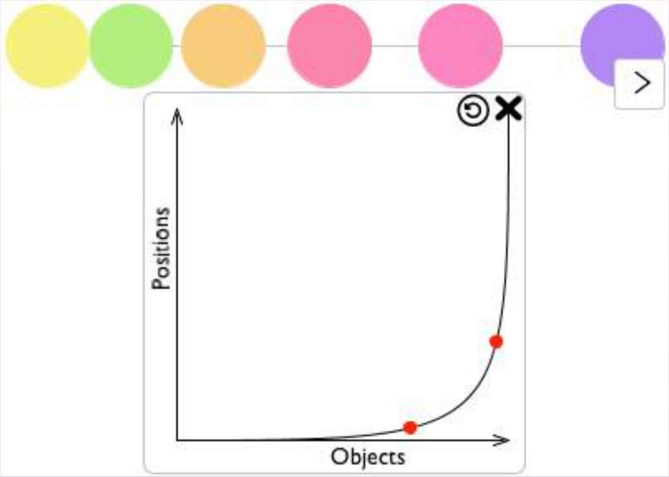


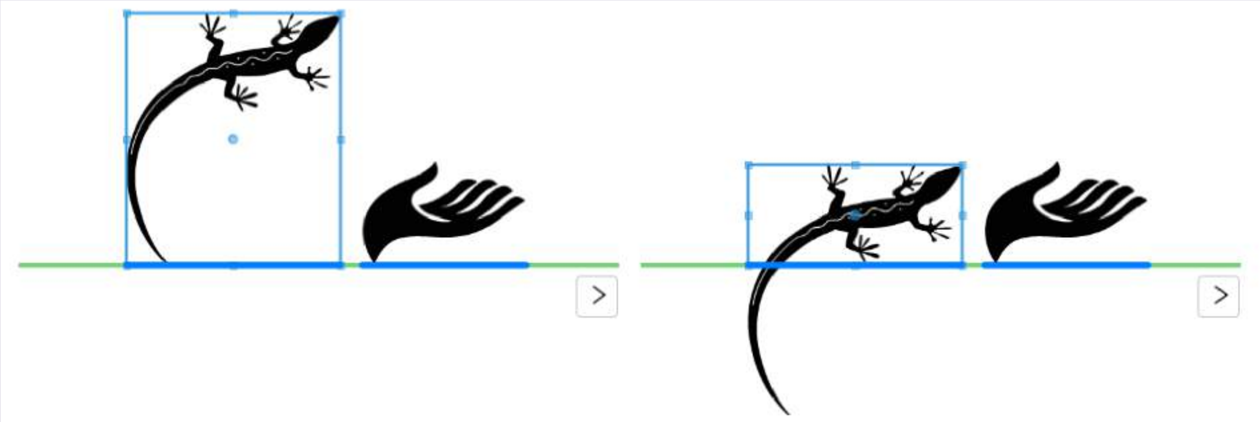
StickyLines: Use key principles to

Reify :

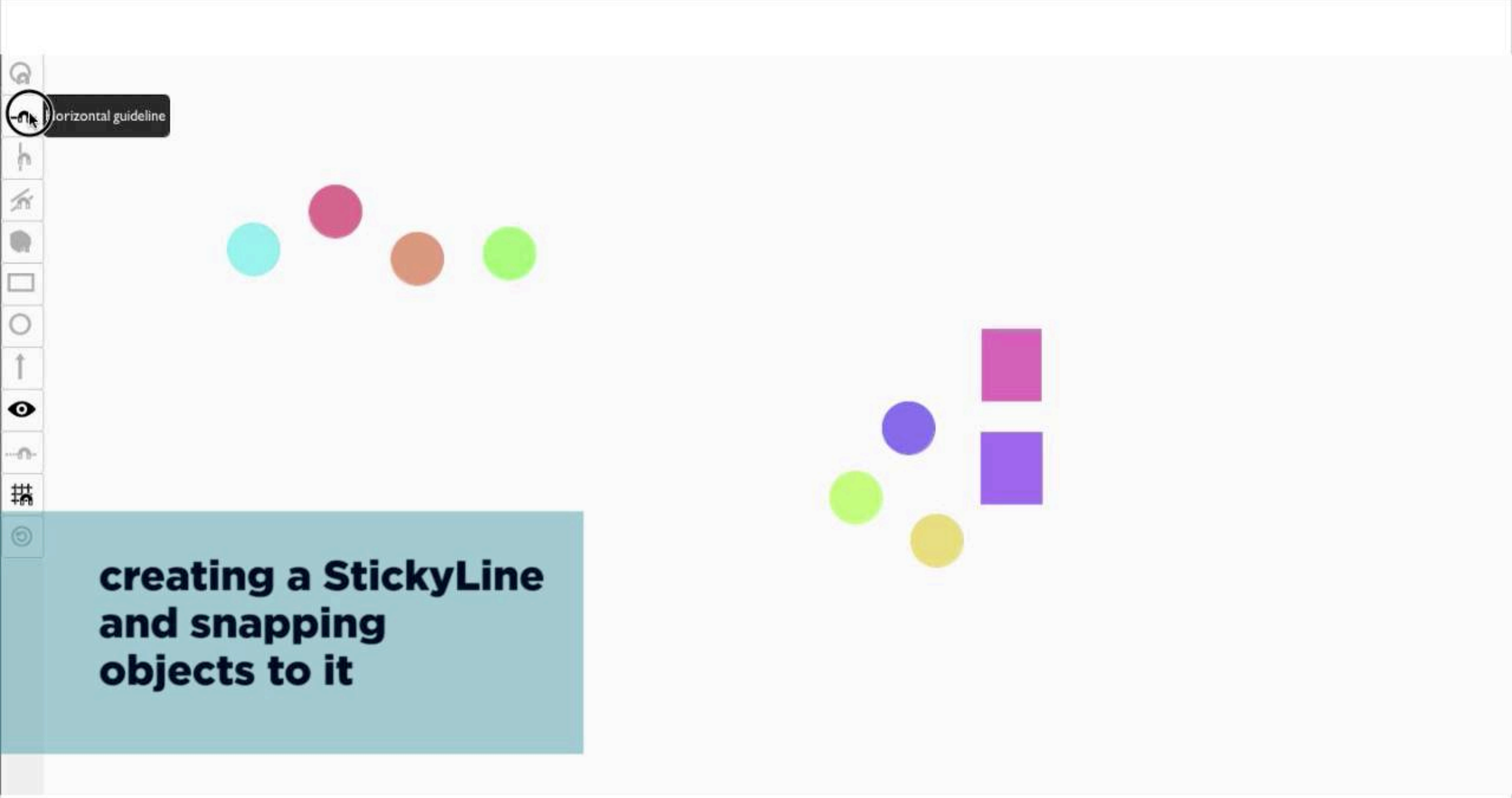
- alignment
- distribution
- ‘tweaks’







StickyLines



creating a StickyLine
and snapping
objects to it

Towards generative theory

Define principles of a
unified theory of interaction

Instrumental Interaction

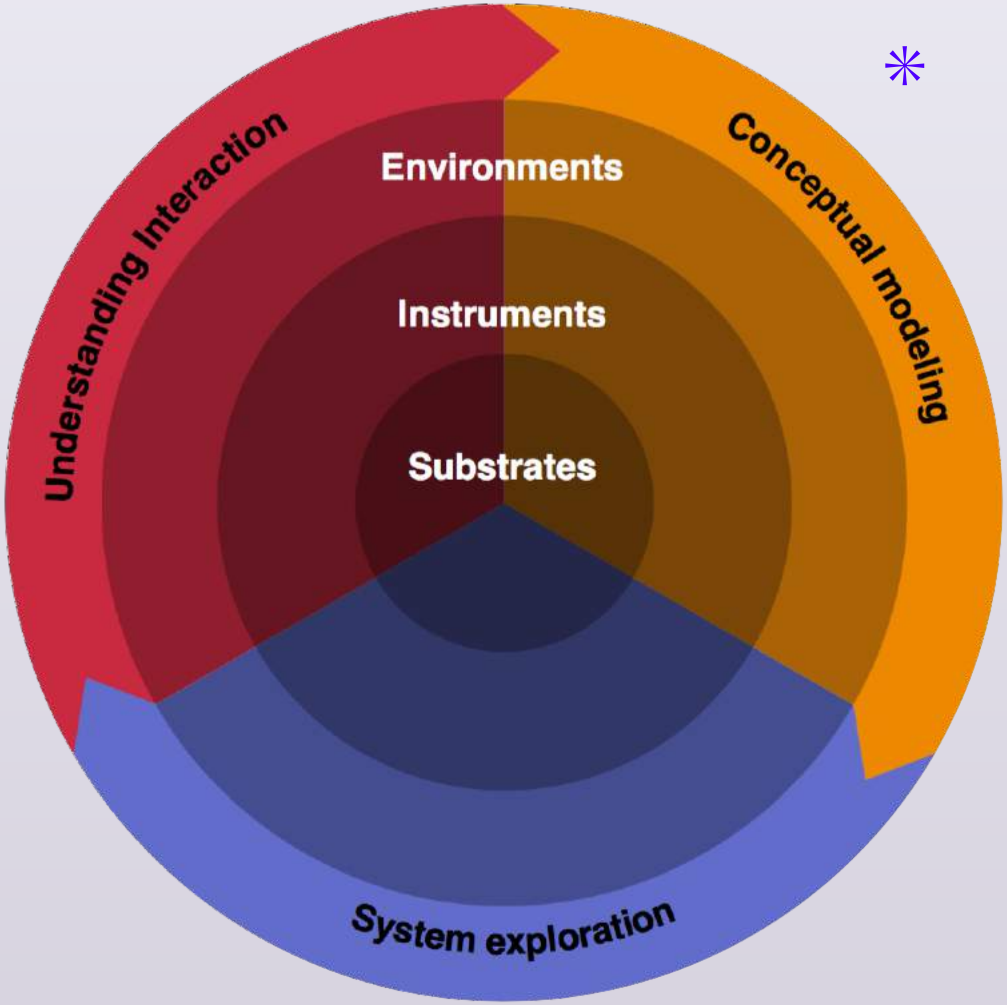
  Reification

  Polymorphism

  Reuse

Substrates

Reciprocal Co-Adaptation



*

* with Michel Beaudouin-Lafon

Webstrates

Any web document (HTML) served by the Webstrates server

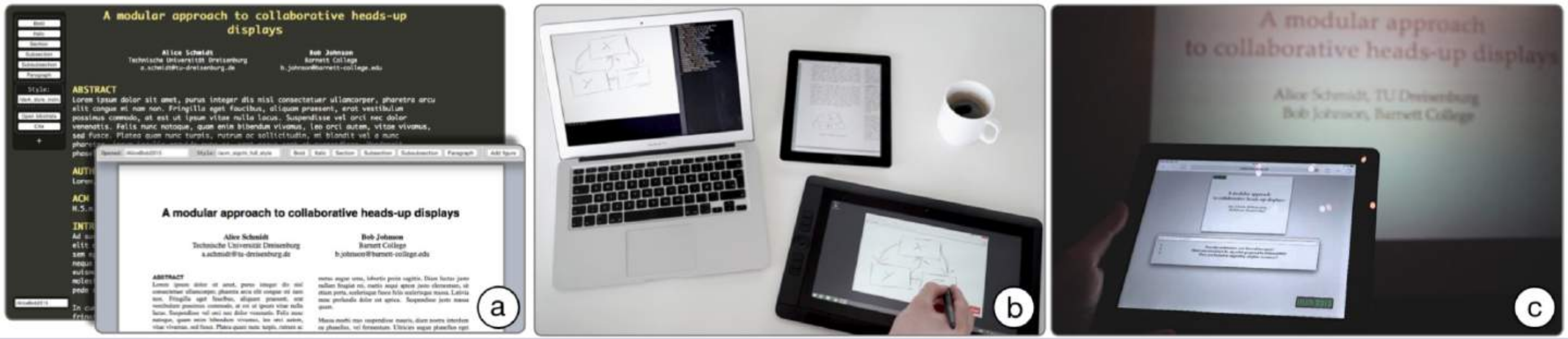
    is shared by everyone who looks at it in a regular web browser

Any changes are immediately visible to everyone.

Unlike google docs

    Create your own editor (just a doc) with own tools (ditto)

Edit the same doc with your personal editor and tool

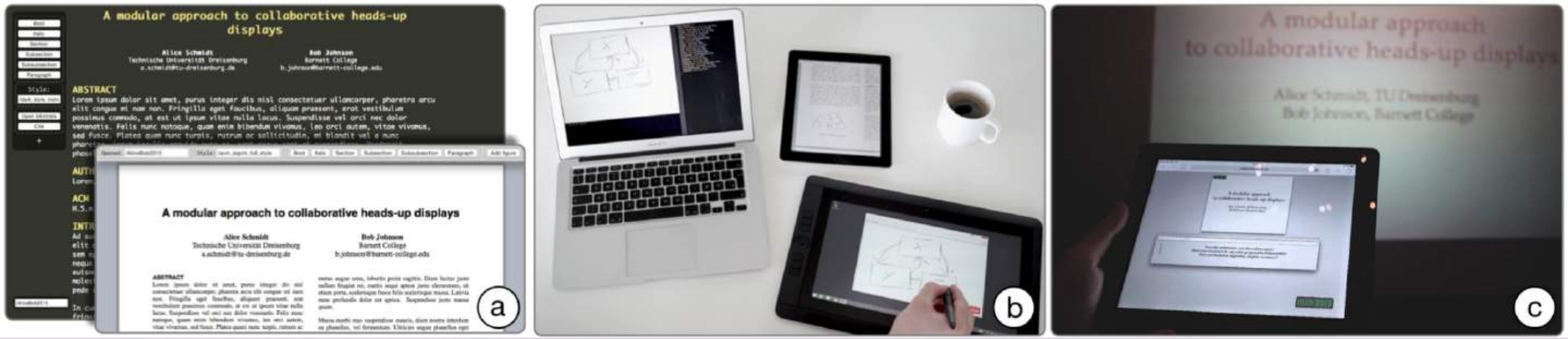

A modular approach to collaborative heads-up displays
Alice Schmidt
Technische Universität Dreisenburg
Bob Johnson
Barnett College
ABSTRACT
a
b
A modular approach to collaborative heads-up displays
Alice Schmidt, TU Dreisenburg
Bob Johnson, Barnett College
c

Webstrates

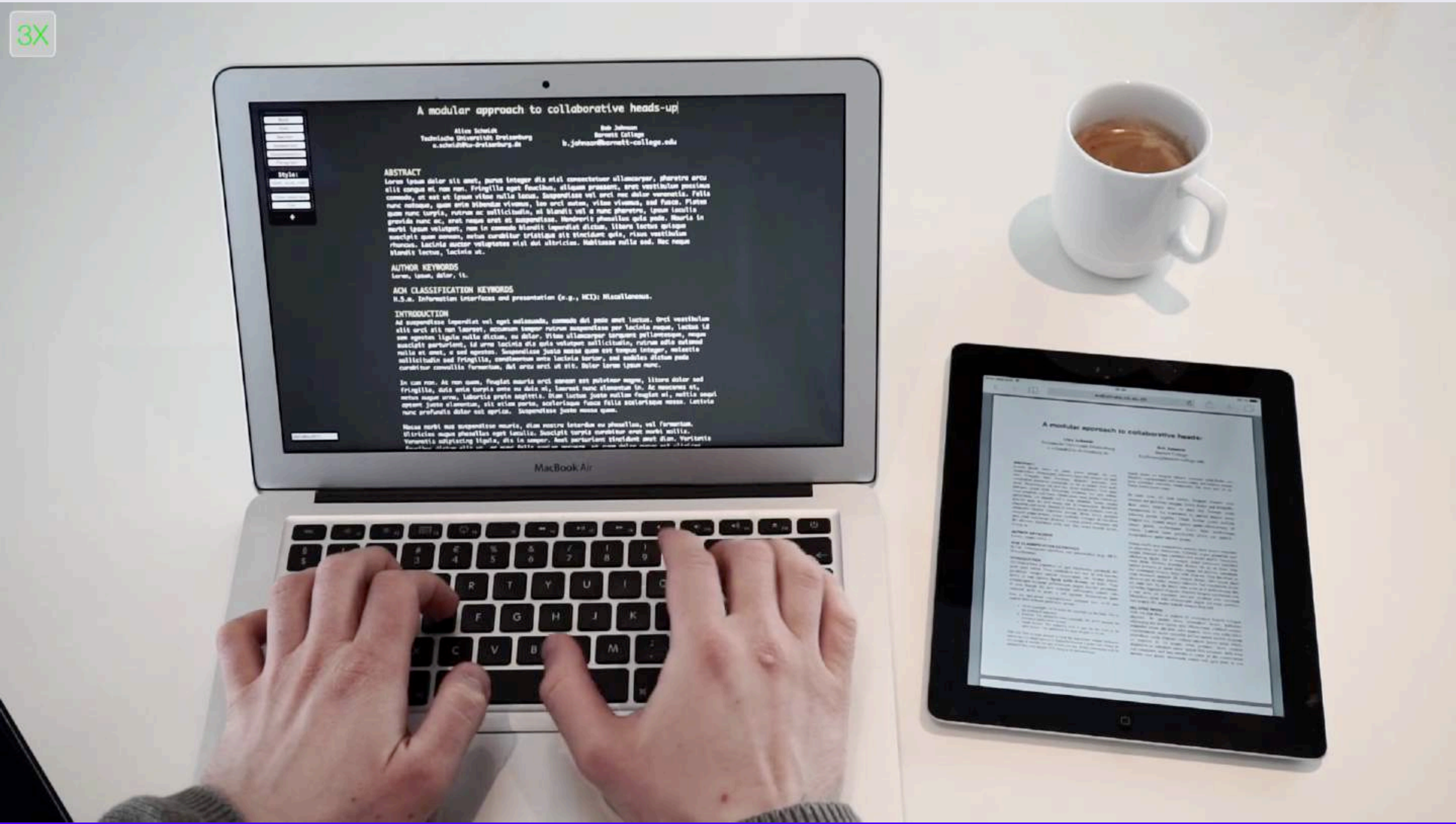
Shareable dynamic media :

malleable by users, who appropriate them
shareable among users, who collaborate on them
distributable across diverse devices and platforms

Users interacts with one document, with personal editors


A modular approach to collaborative heads-up displays
Alice Schmidt
Technische Universität Dreisenburg
Bob Johnson
Barnett College
ABSTRACT
A modular approach to collaborative heads-up displays
Alice Schmidt, TU Dreisenburg
Bob Johnson, Barnett College
a
b
c

Webstrates


3X
A modular approach to collaborative heads-up
ABSTRACT
AUTHOR KEYWORDS
ACM CLASSIFICATION KEYWORDS
INTRODUCTION
MacBook Air
A modular approach to collaborative heads-

Towards generative theory

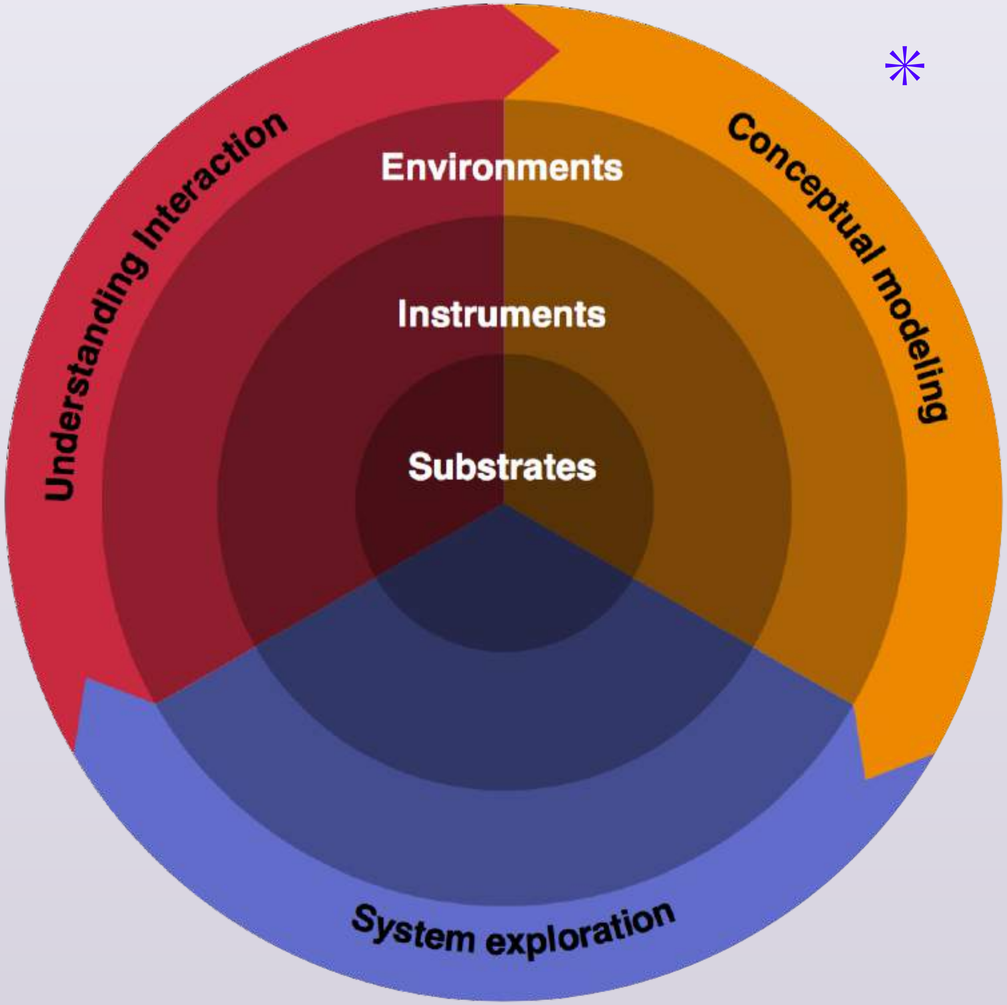
Define principles of a
unified theory of interaction

Instrumental Interaction

- Reification
- Polymorphism
- Reuse

Substrates

Reciprocal Co-Adaptation



*

* with Michel Beaudouin-Lafon

How we interact with computers

Computer as ***tool***

Empower users

Human-Computer Interaction

Computer as ***servant***

Delegate tasks

Artificial Intelligence

Computer as ***medium***

Communicate

Mediated Communication

Human-Computer Partnerships

Combine:

computer as a tool

to augment human capabilities

and

computer as a servant

to take over certain tasks

Keep the user in control

Competing perspectives

Human-in-the-loop

Machine learning perspective:

Human is *input to the* ***algorithm***

‘human-in-the-loop’ ?

Competing perspectives

Human-in-the-loop

Machine learning perspective:

Human is *input to the* ***algorithm***

Computer-in-the-loop

HCI perspective:

Algorithm is input to *inform the* ***user***

Human-Computer Partnerships

Instead of just creating models of *users*
to inform the *system*

Shouldn't we create models of the *system*
to inform the *user?*

Together, they can create effective
human-computer partnerships

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

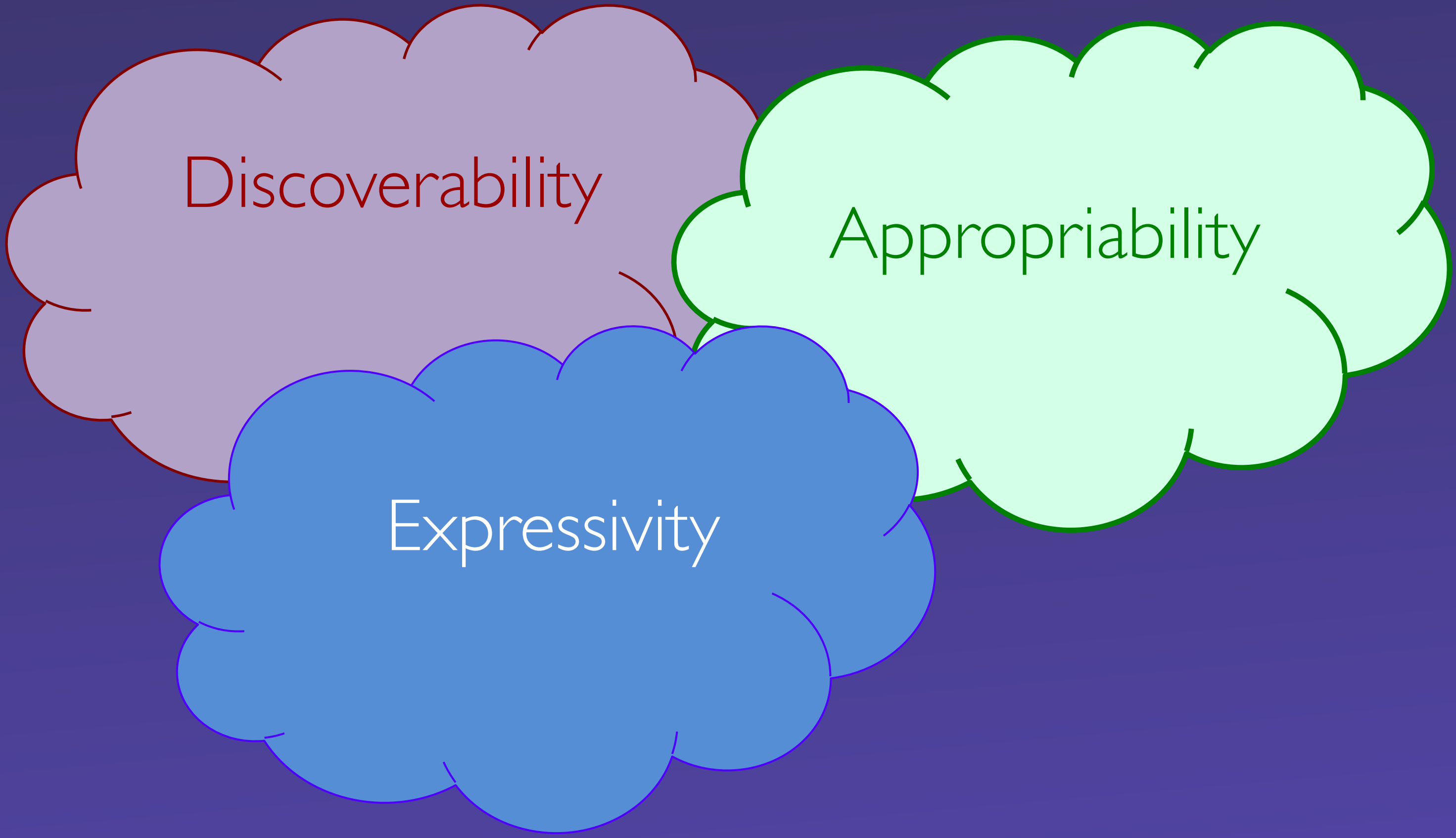
Computers modify human behavior

… training (or persuasion)

Human-Computer Partnerships

People *adapt to* technology they *learn* it

adapt the technology they *appropriate* it

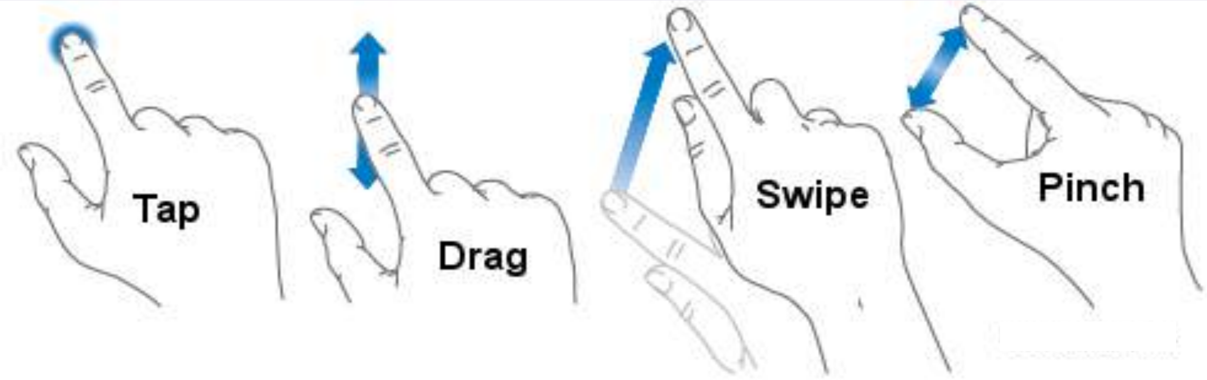


Discoverability

Appropriability

Expressivity

Smart phones are easy to use



... but interaction is more limited

Why can’t users learn to ‘play’ phones ?

Users should be able to progress
from novice to virtuoso

Towards generative theory*

Define principles of a *unified theory of interaction*

Instrumental Interaction

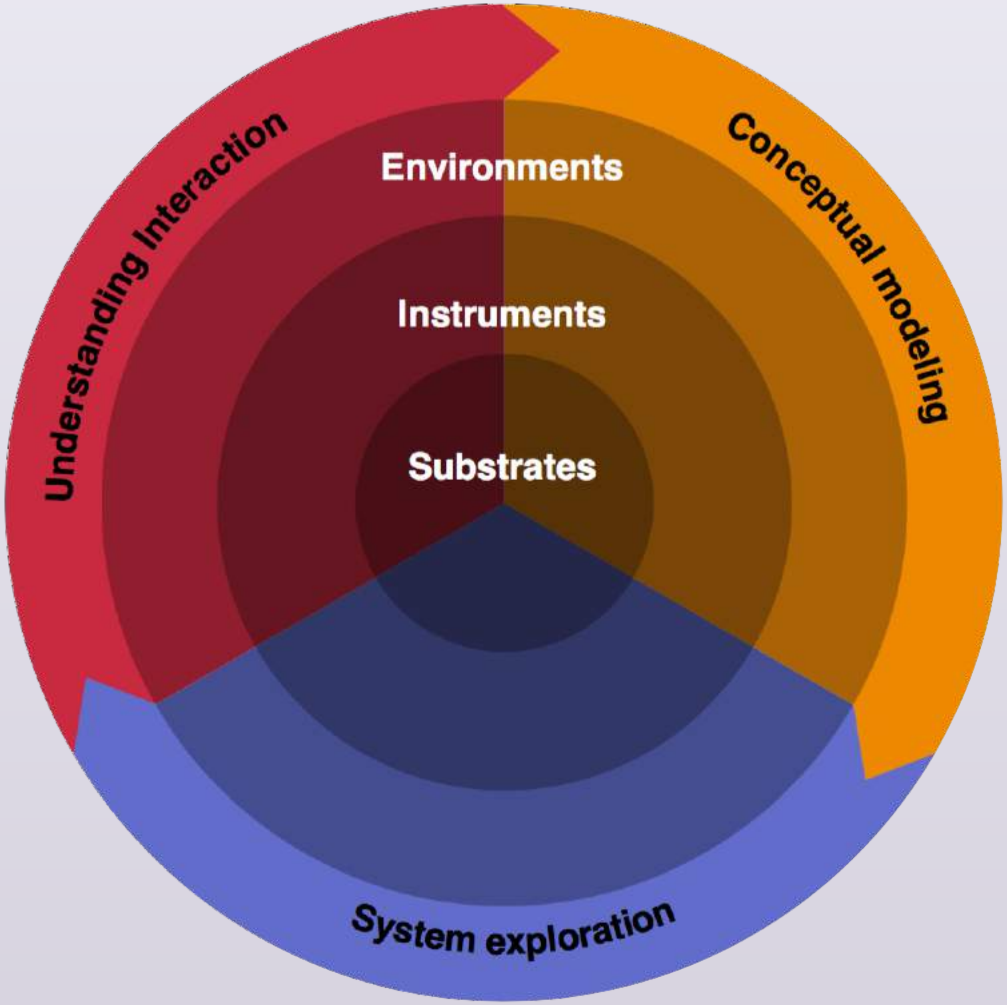
Reification

Polymorphism

Reuse

Substrates

Reciprocal Co-Adaptation



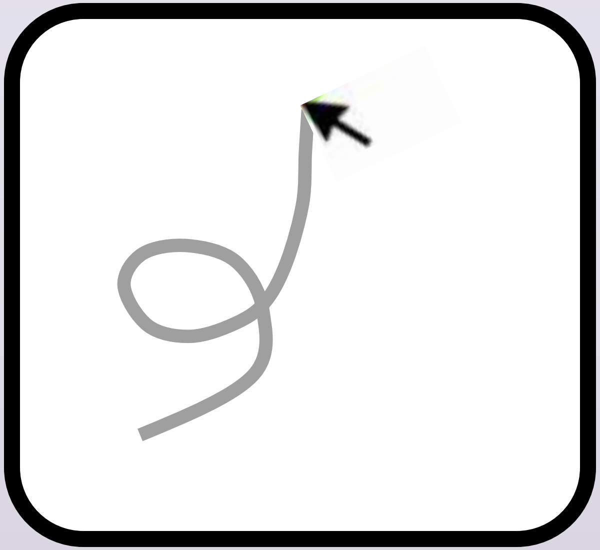
* with Michel Beaudouin-Lafon

Discoverability

How can I learn
which gesture
executes which command?

Octopocus

Experts just perform the gesture



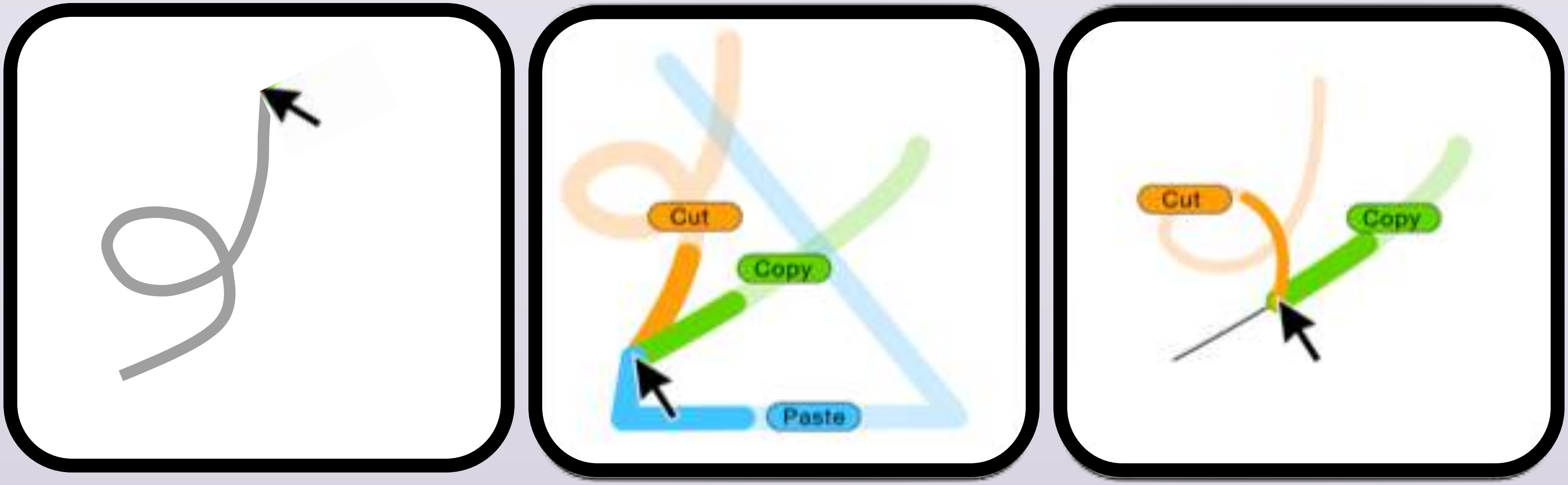
Bau & Mackay, UIST'09

Octopocus

Experts just perform the gesture

Novices **pause** . . .

and the Octopocus guide appears



Bau & Mackay, UIST’09

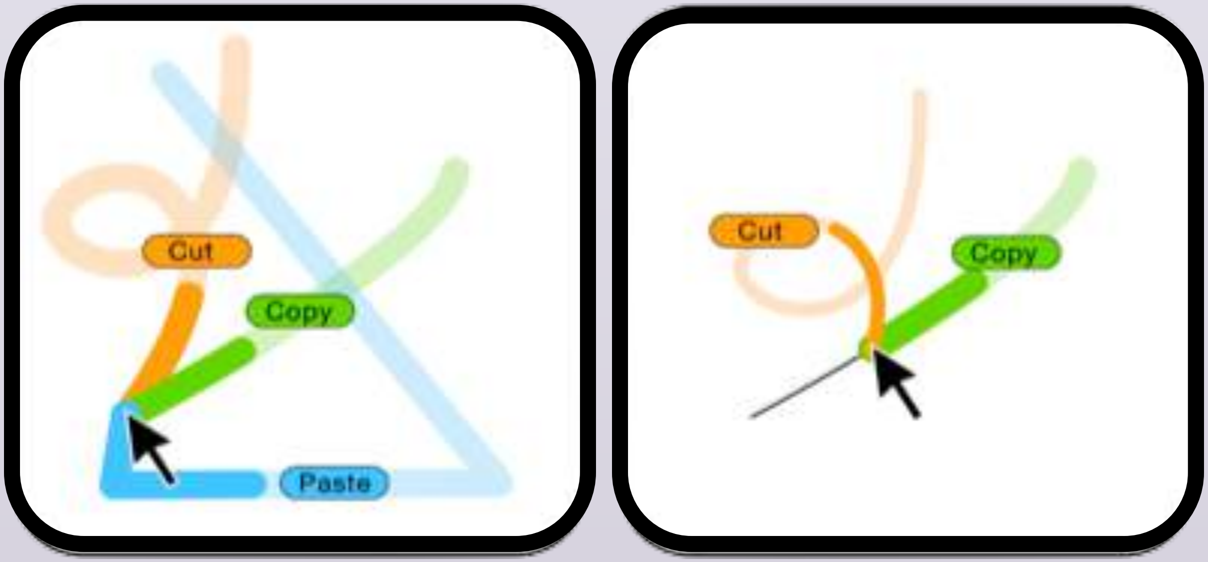
Octopocus

Progressive feedforward

What gestures are available ?

Progressive feedback

What did the system recognize ?



Bau & Mackay, UIST'09

Inking the '*Help*' command

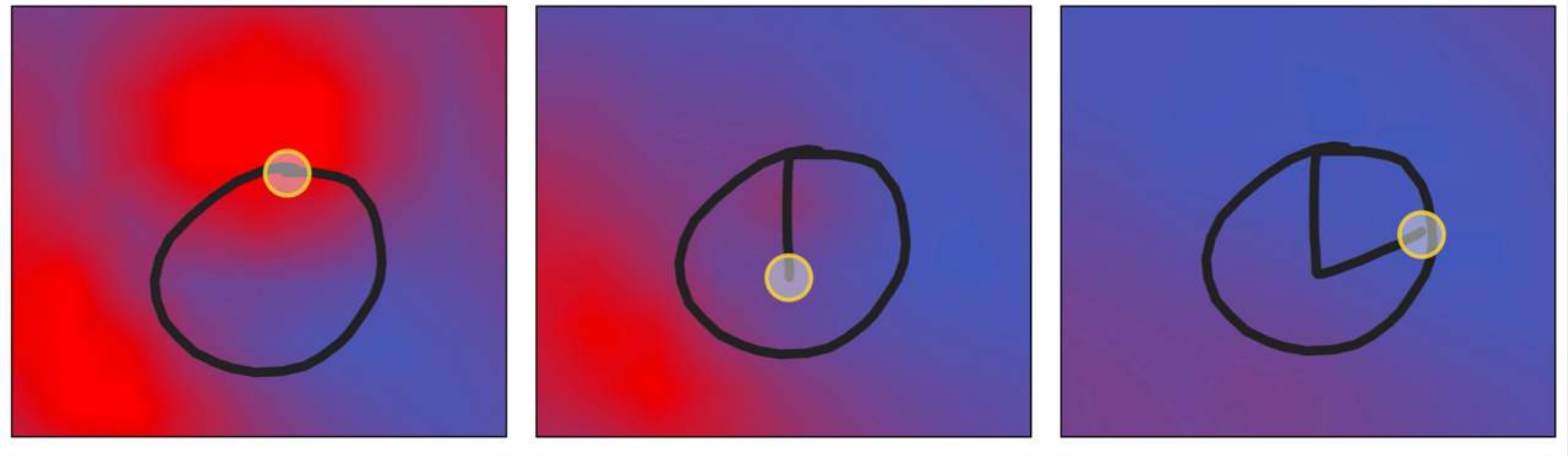
Appropriability

How can I
create my own
gesture commands?

Fieldward

To create your own gesture commands, they must be:

easy for you to remember



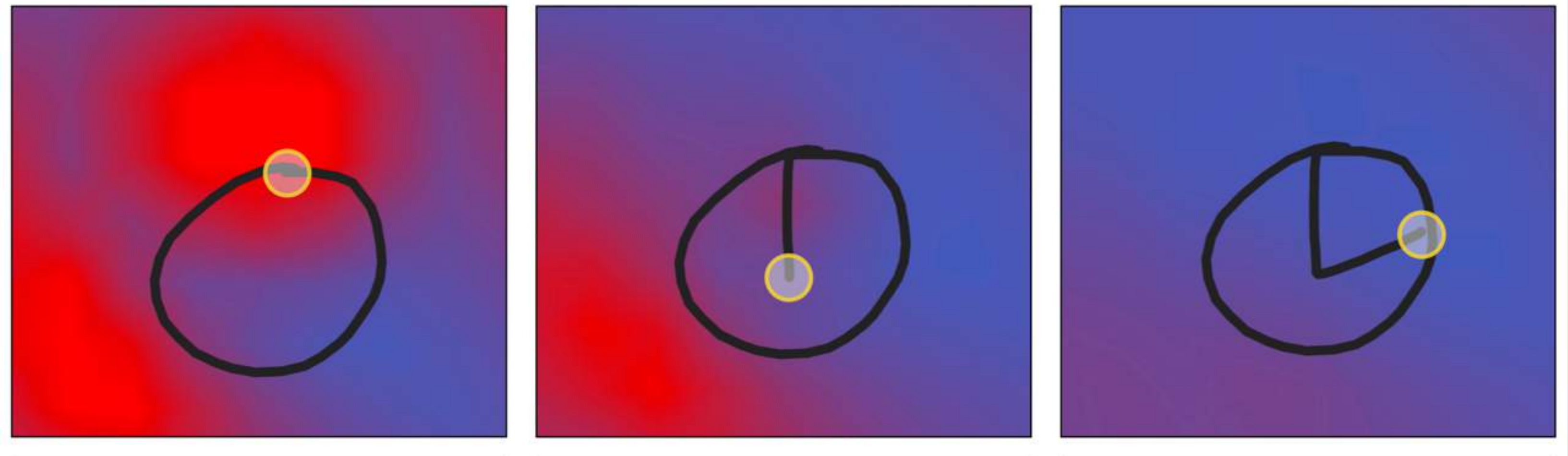
Malloch, Griggio, McGrenere & Mackay CHI'17

Fieldward

To create your own gesture commands, they must be:

easy for you to remember

easy for the system to recognize



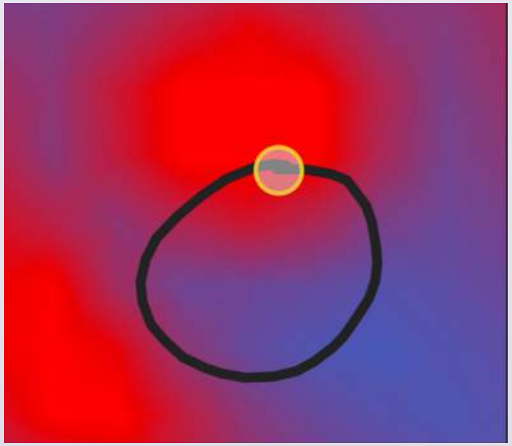
Malloch, Griggio, McGrenere & Mackay CHI'17

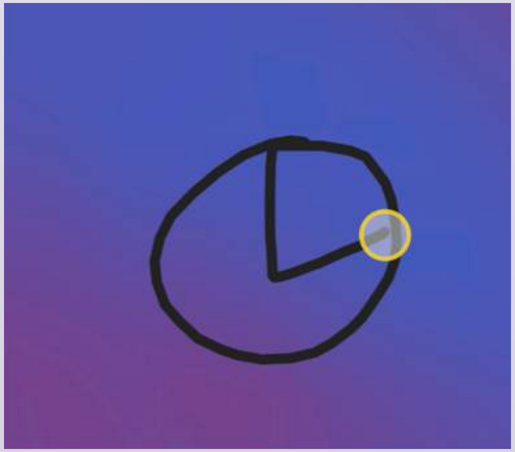
Fieldward

Draw a gesture

If it ends in a red zone
    the gesture already exists

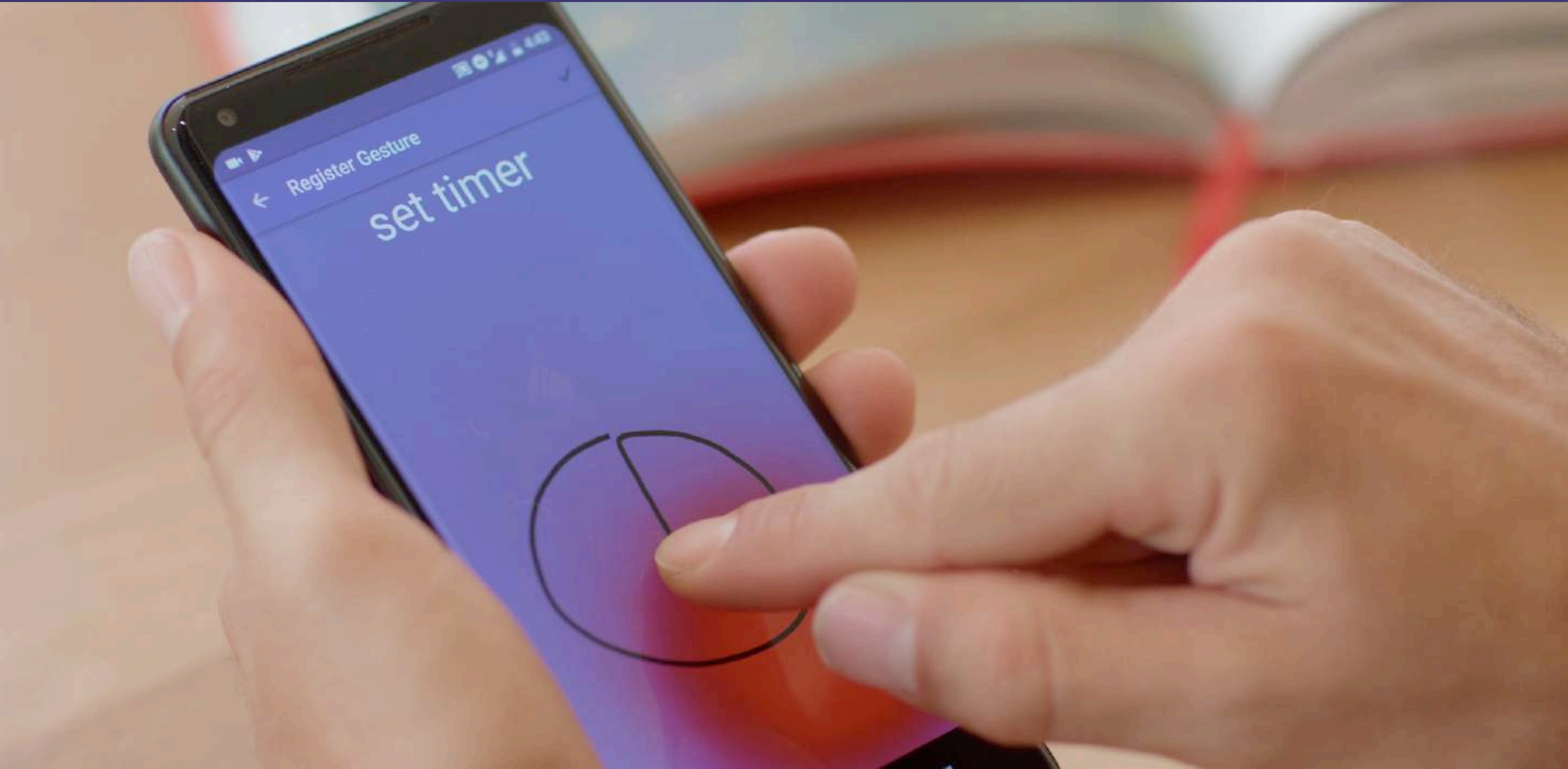
If it ends in a blue zone
    you have a new gesture !

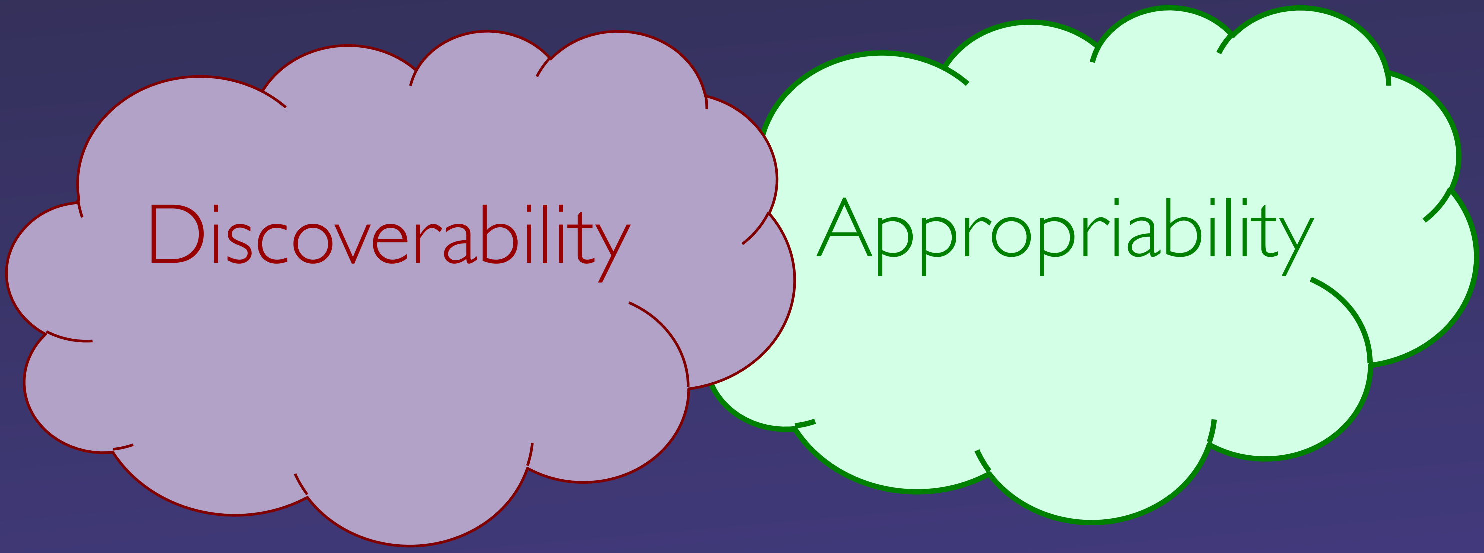




Malloch, Griggio, McGrenere & Mackay CHI'17

Fieldward (set timer)





Discoverability

Appropriability

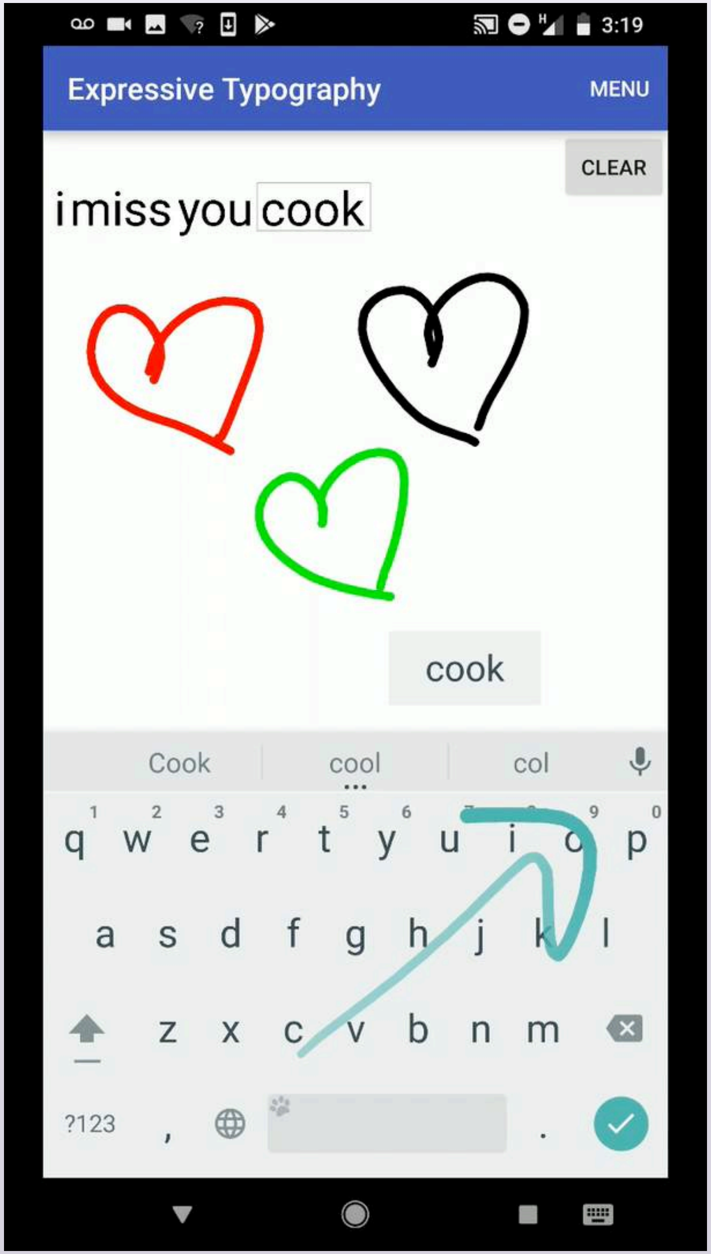
How can I access
the phone’s power
. . . simply ?

CommandBoard

Transform the space
above a soft keyboard
into a command input space

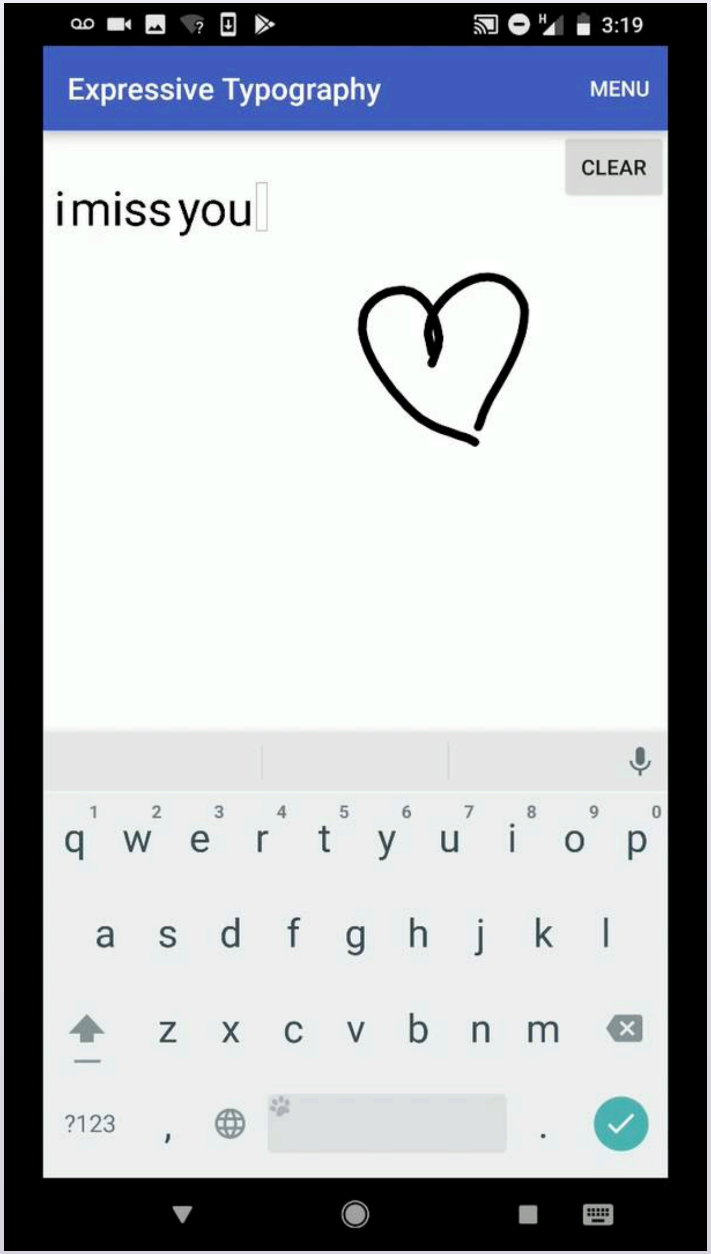
Offers the power of a
command-line interface
on a mobile phone

Alvina, Griggio, Bi & Mackay UIST'17



CommandBoard

Type 'doodle'
then 'execute' gesture ^
Launches 'doodle'



Alvina, Griggio, Bi & Mackay UIST'17

CommandBoard

Type 'doodle'

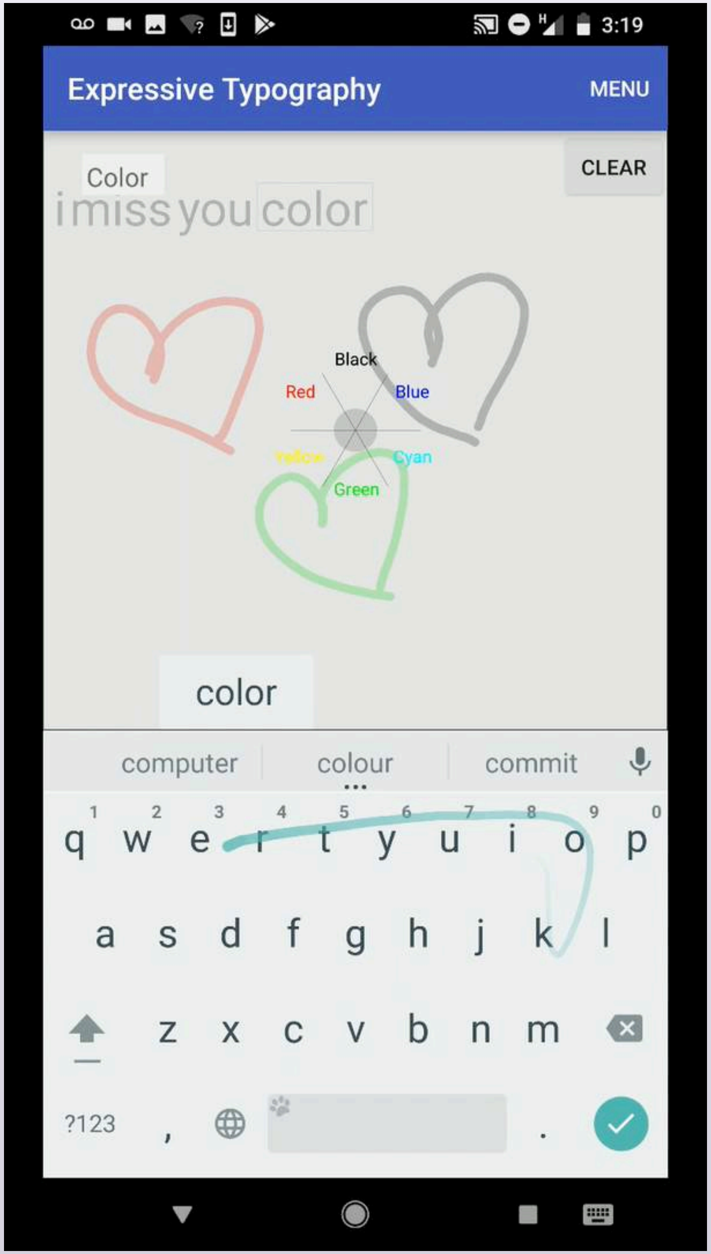
then 'execute' gesture ^

Launches 'doodle'

Type 'color'

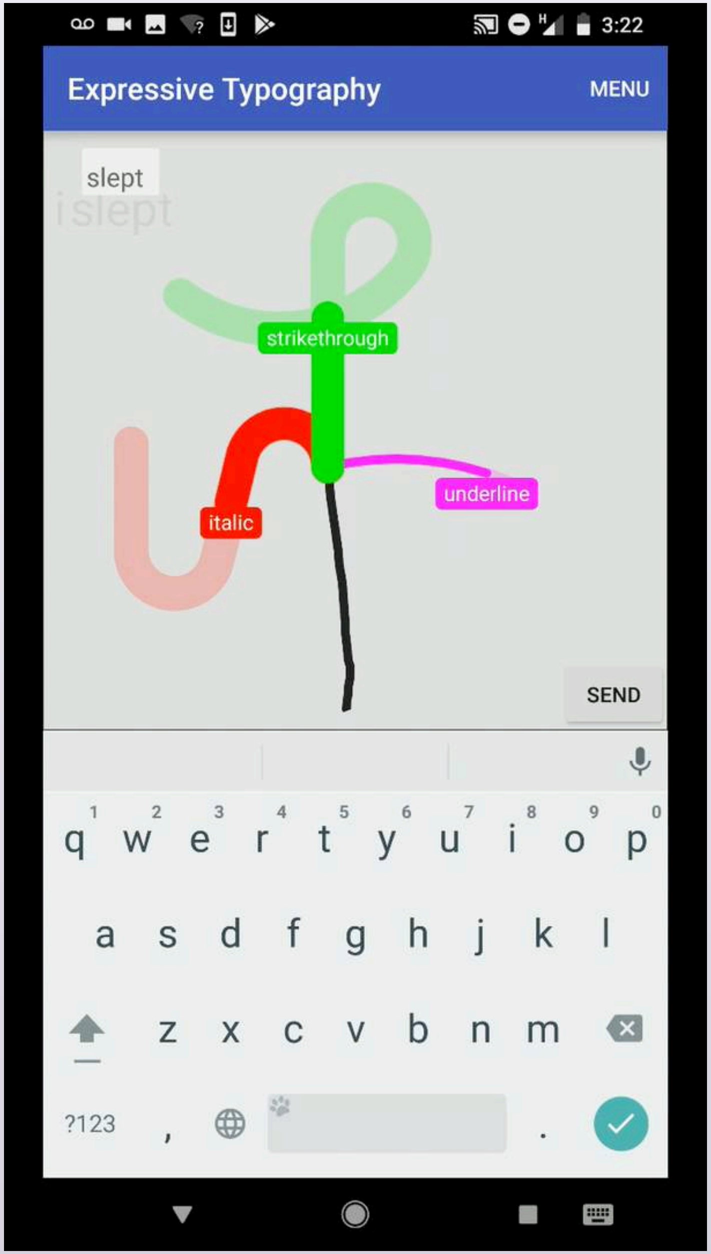
then select a color

Alvina, Griggio, Bi & Mackay UIST'17



Commandboard

Use progressive feedforward to **discover** strike-through command



Alvina, Griggio, Bi & Mackay UIST'17

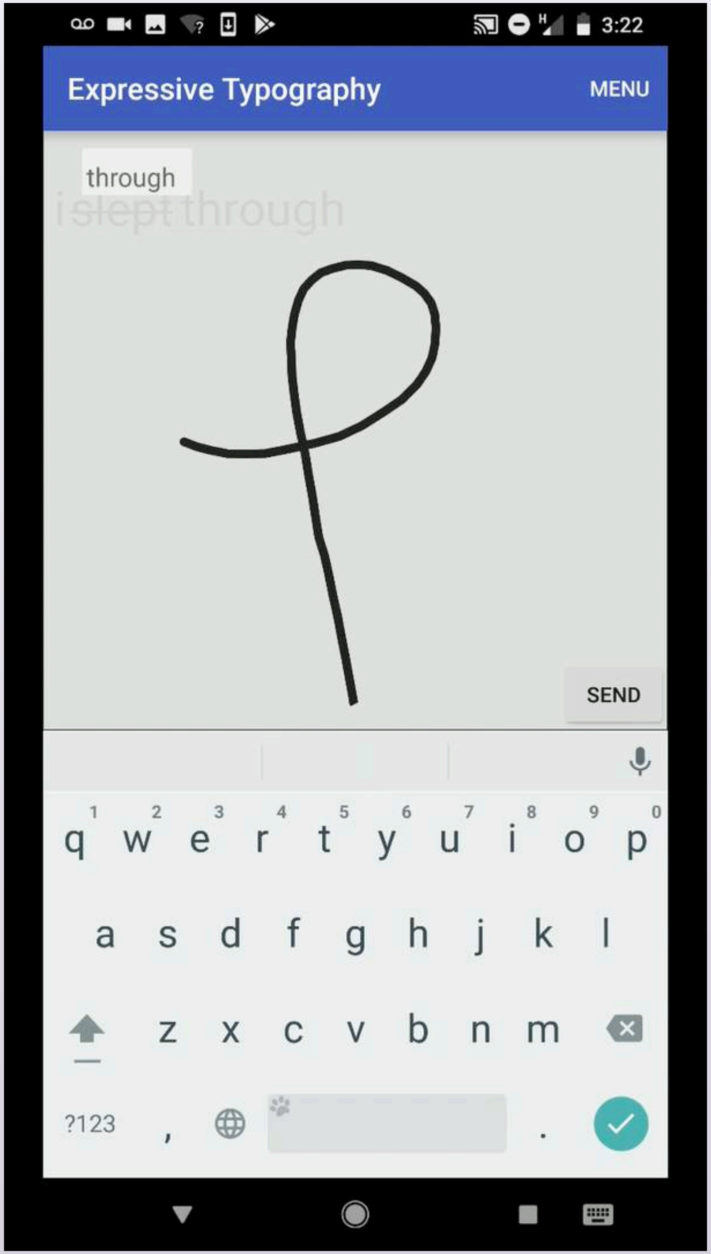
Commandboard

Use progressive feedforward to
discover strike-through command

When you know the gesture
you just draw it

I ~~slept through~~ loved the lecture

Alvina, Griggio, Bi & Mackay UIST'17



Expressivity

How can I
generate
expressive output?

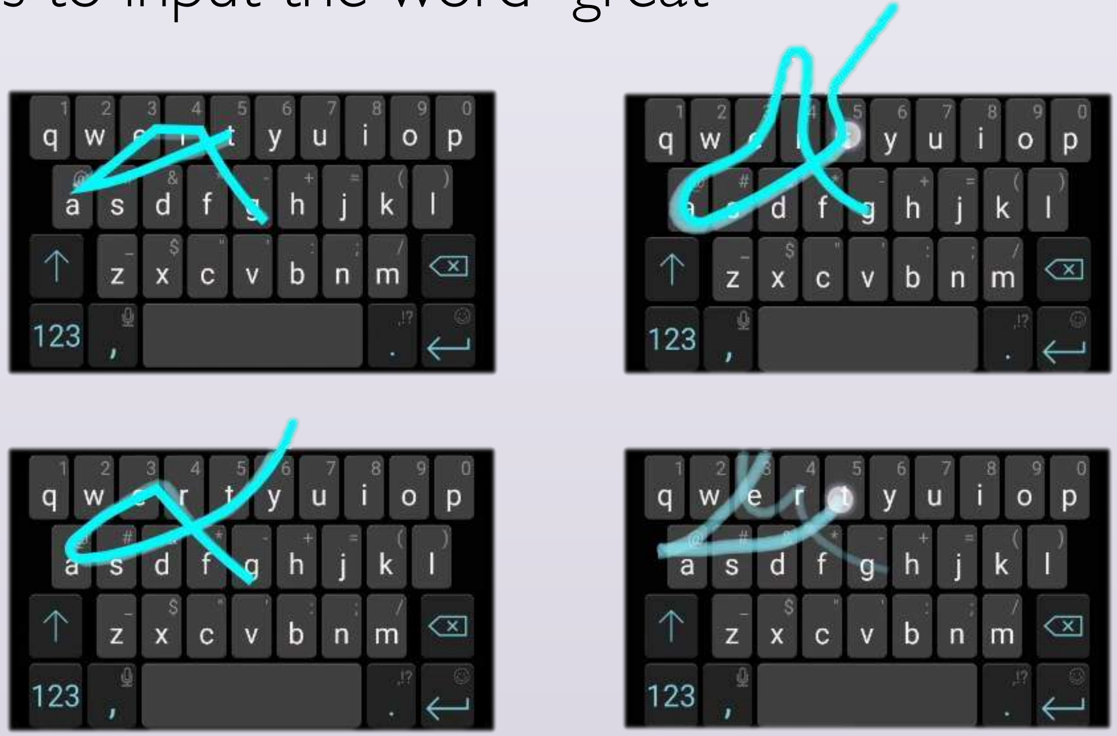
Human expression vs. Machine classification

Machine learning algorithms:

- Goal is to classify the correct word
- Human variation is treated as noise

Gesture typing algorithms are great . . .

Four ways to input the word "great"

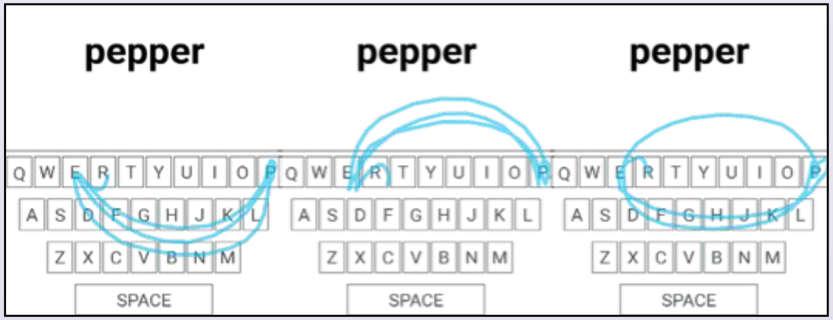


All produce the identical result: **great**

Expressive Keyboard vs. Machine classification

Machine learning approach

- Classify the correct word
- ***Remove*** human variation

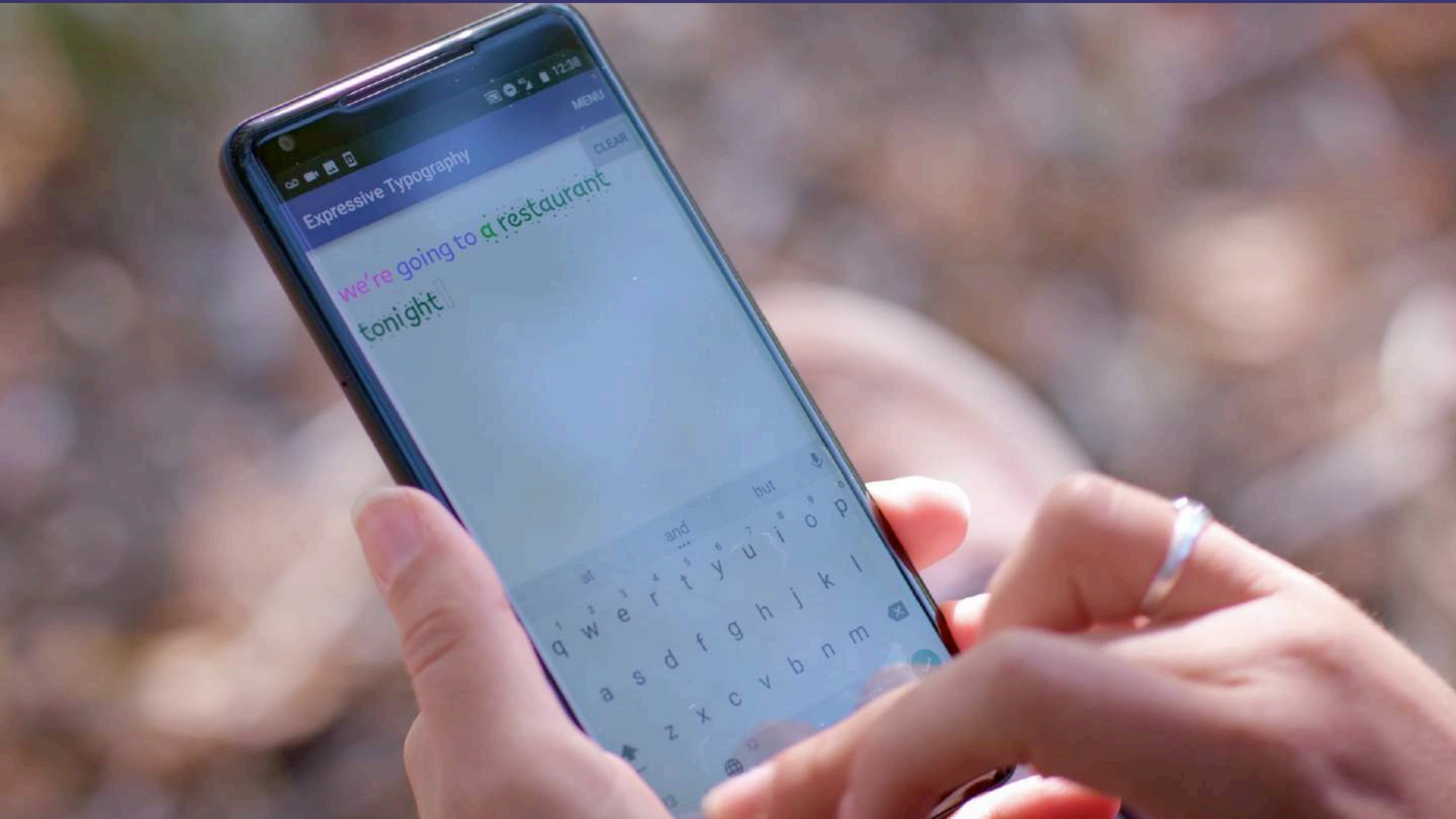


Our approach

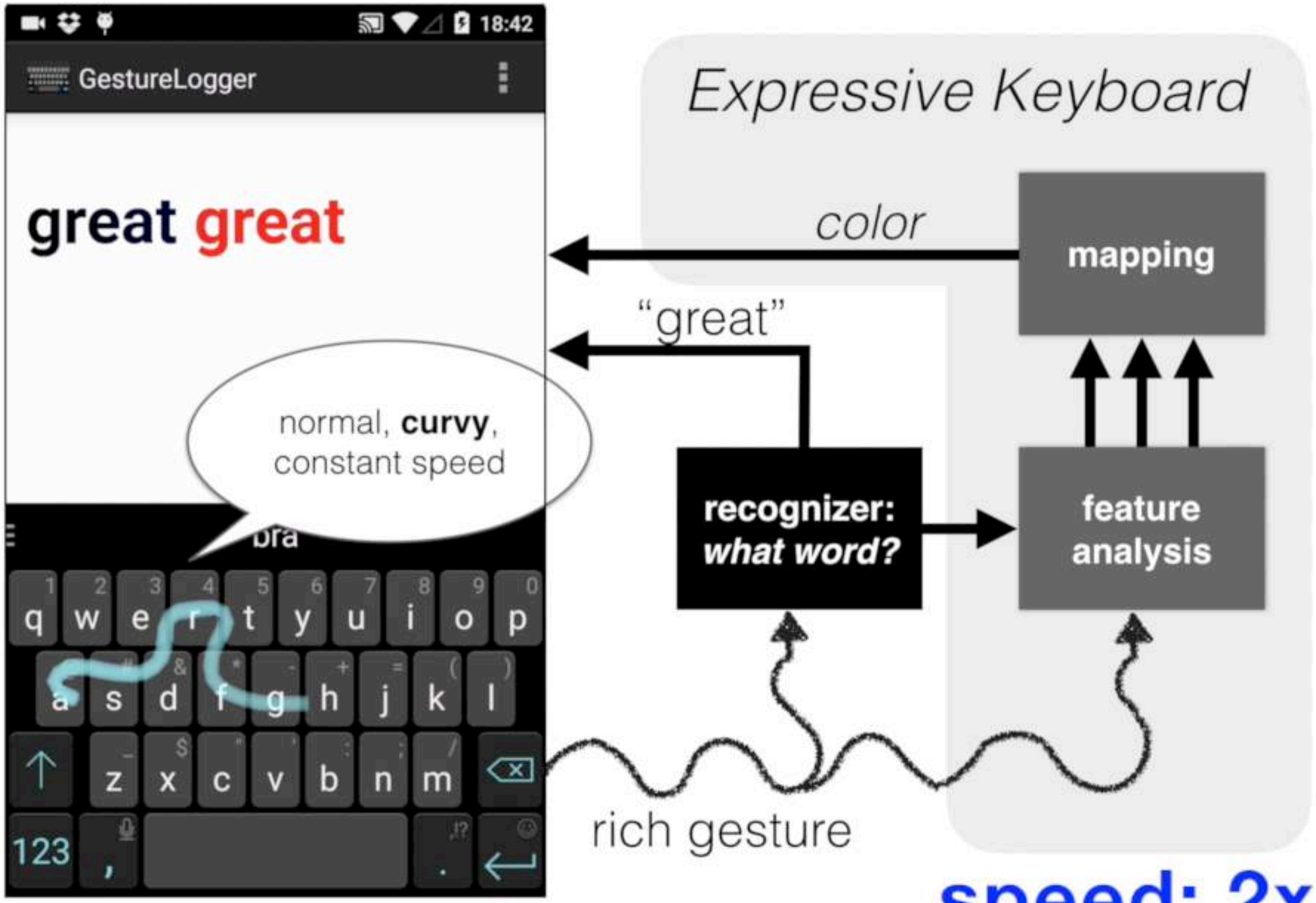
- ***Transform*** human variation into expressive output
  - color, emojis, typography ...

Alvina, Malloch & Mackay CHI'16

Expressive Keyboard



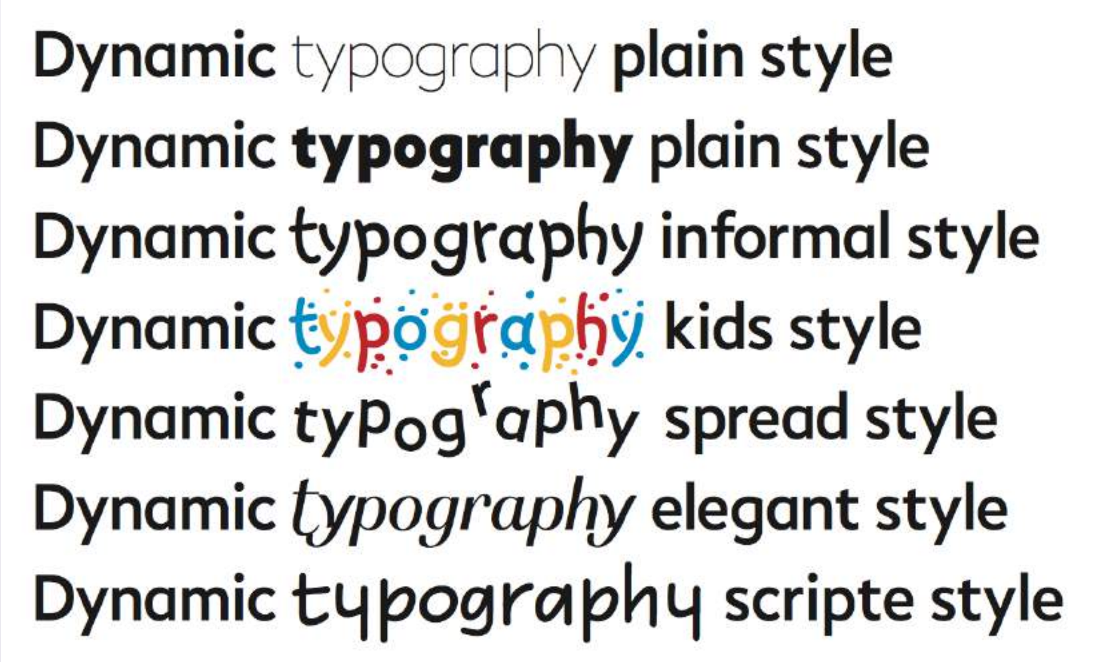
Expressive Keyboard – measure variation



speed: 2x

Expressive Typography

Vary fonts for different
audiences, goals and contexts



Dynamic typography plain style

Dynamic typography plain style

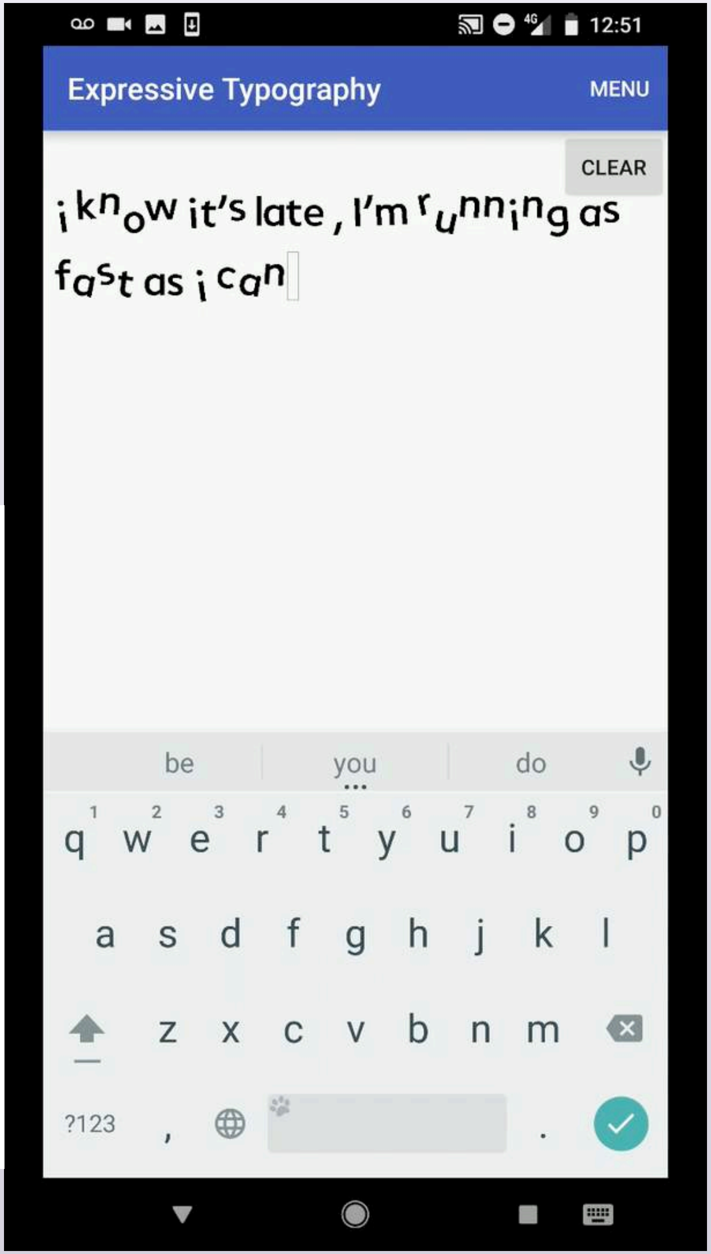
Dynamic typography informal style

Dynamic typography kids style

Dynamic typography spread style

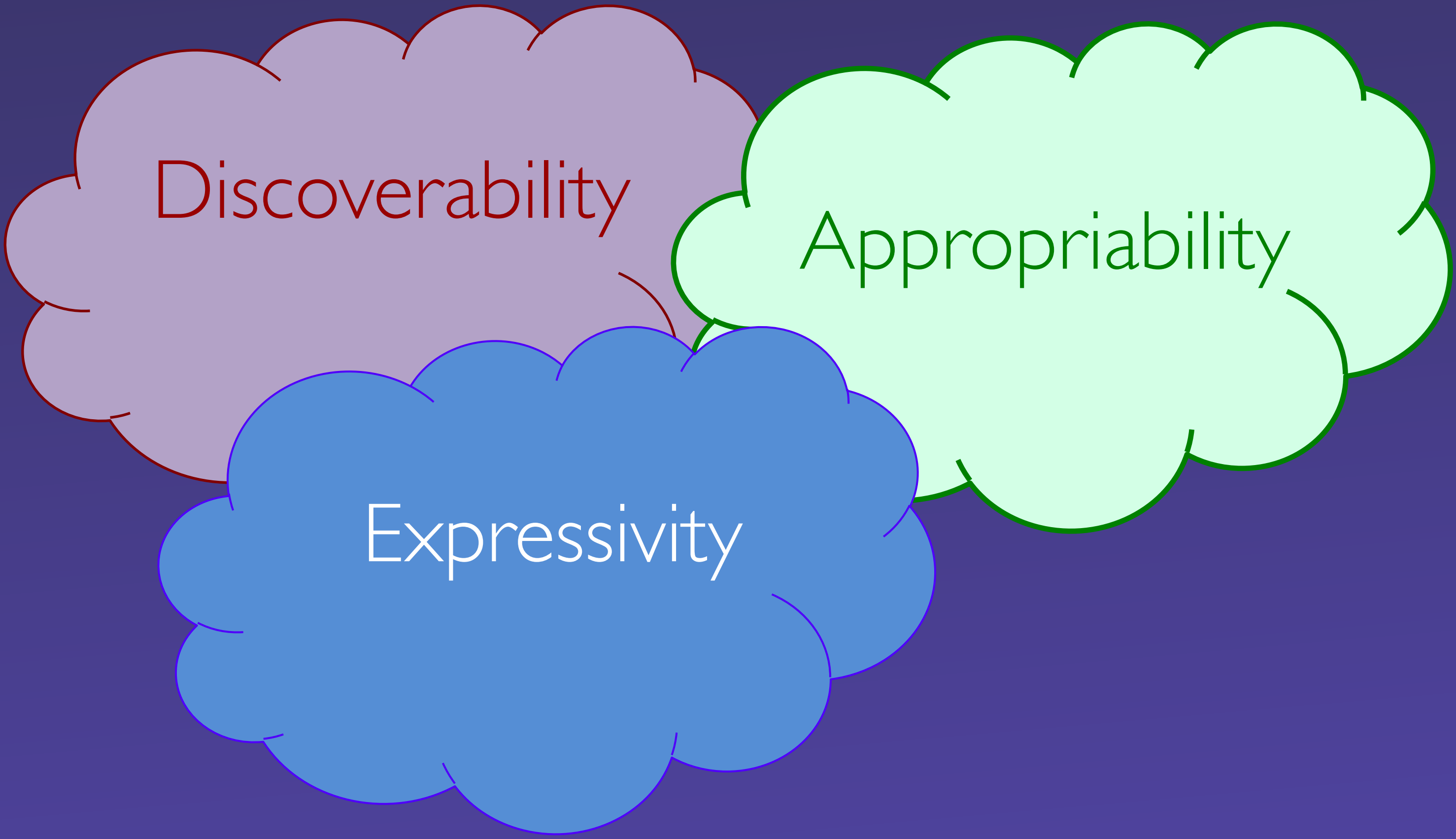
Dynamic typography elegant style

Dynamic typography scripte style



i know it's late, I'm running as fast as i can

Human-Computer Partnerships



Discoverability

Appropriability

Expressivity

Human-computer partnerships like these?

Discover

Appropriate

Express

Thank you !

Wendy E. Mackay

Inria, Université Paris-Saclay

11 October 2018