

Theories and Models for Human-Computer Interaction

Michel Beaudouin-Lafon - mbl@lisn.fr

Laboratoire Interdisciplinaire des Sciences du Numérique

Ex Situ - <http://ex-situ.lri.fr>

Outline

What is a theory? a model?

Perception, action

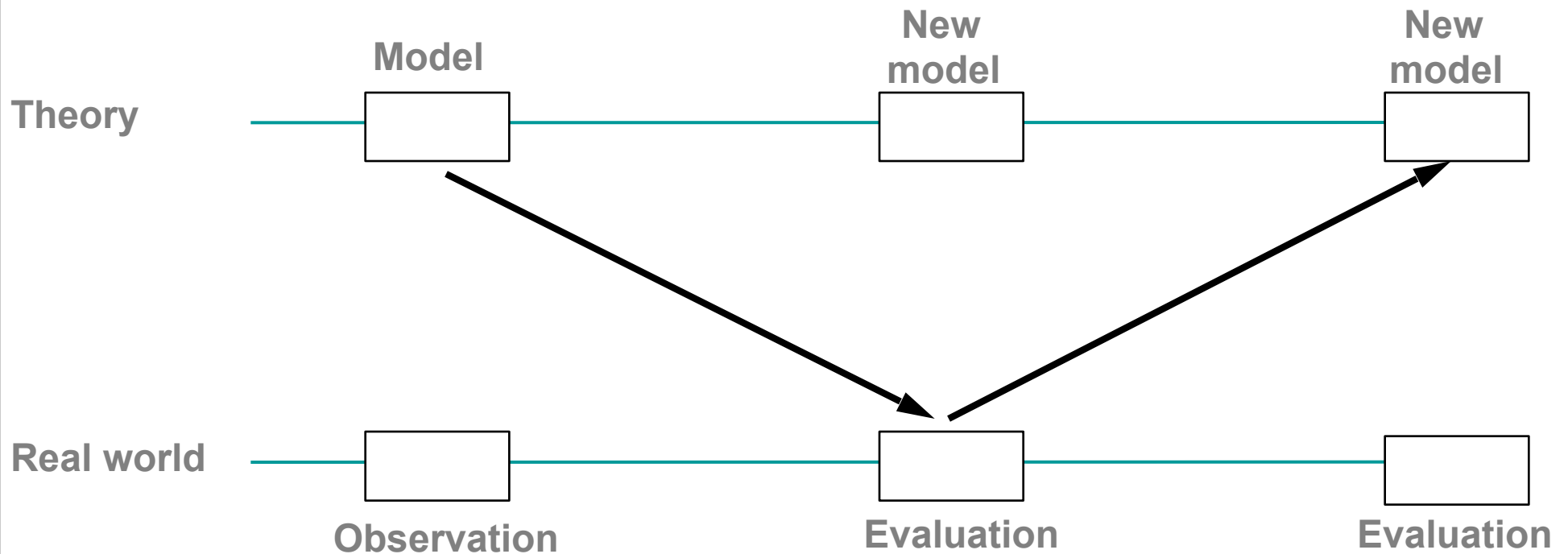
Cognition, behavior

Interaction

Software architectures

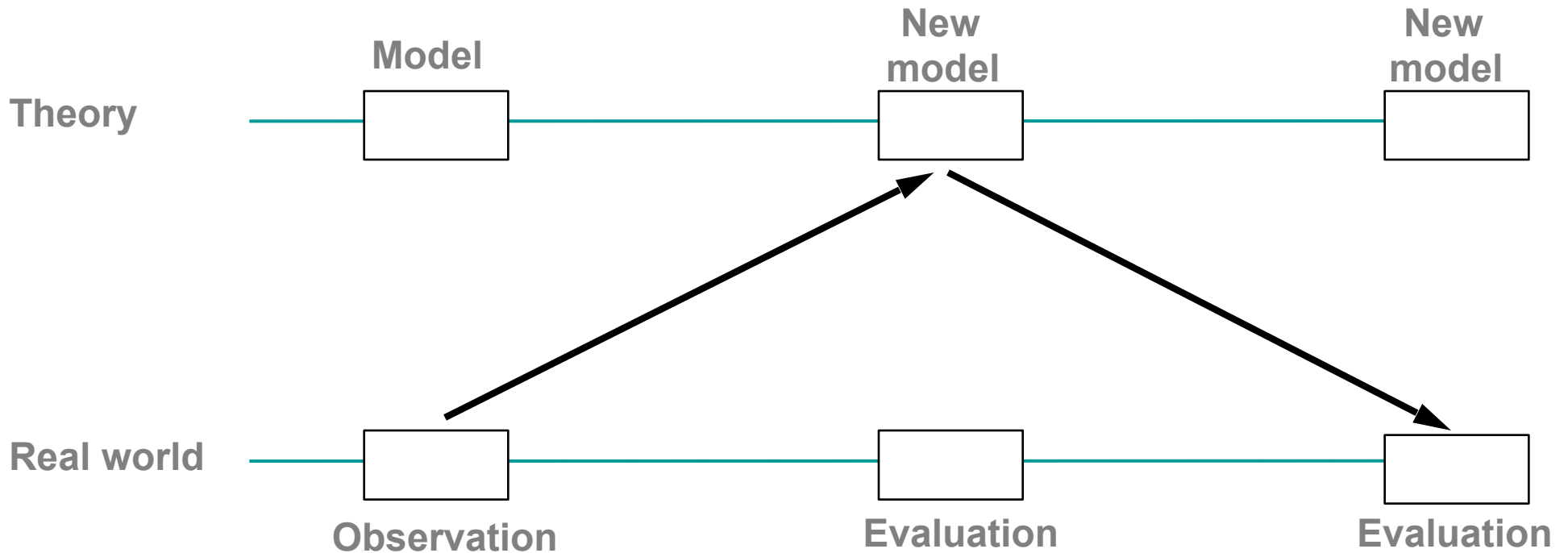
HCI as a scientific discipline

Natural sciences



HCI as a scientific discipline

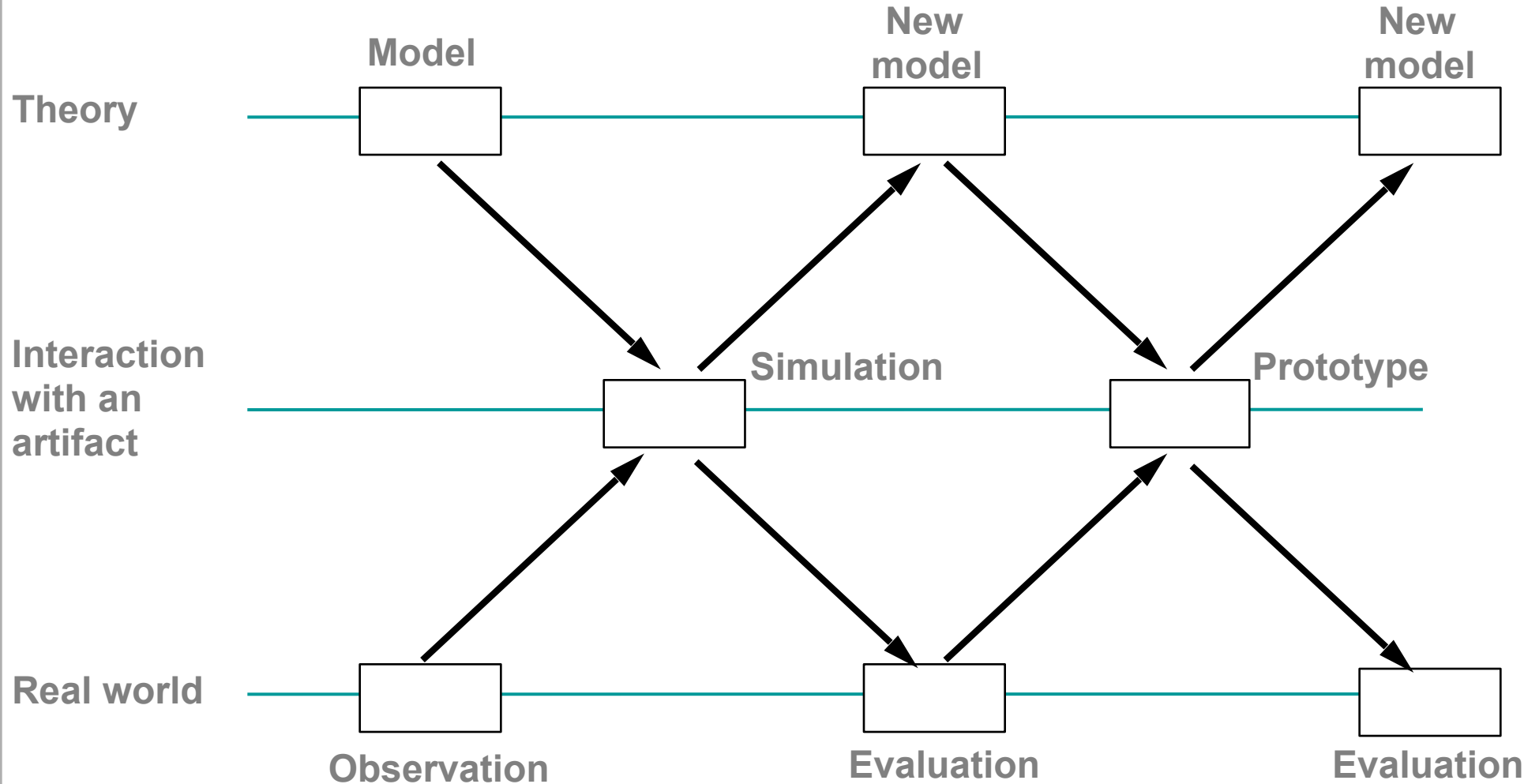
Natural sciences



HCI as a scientific discipline

Mackay & Fayard

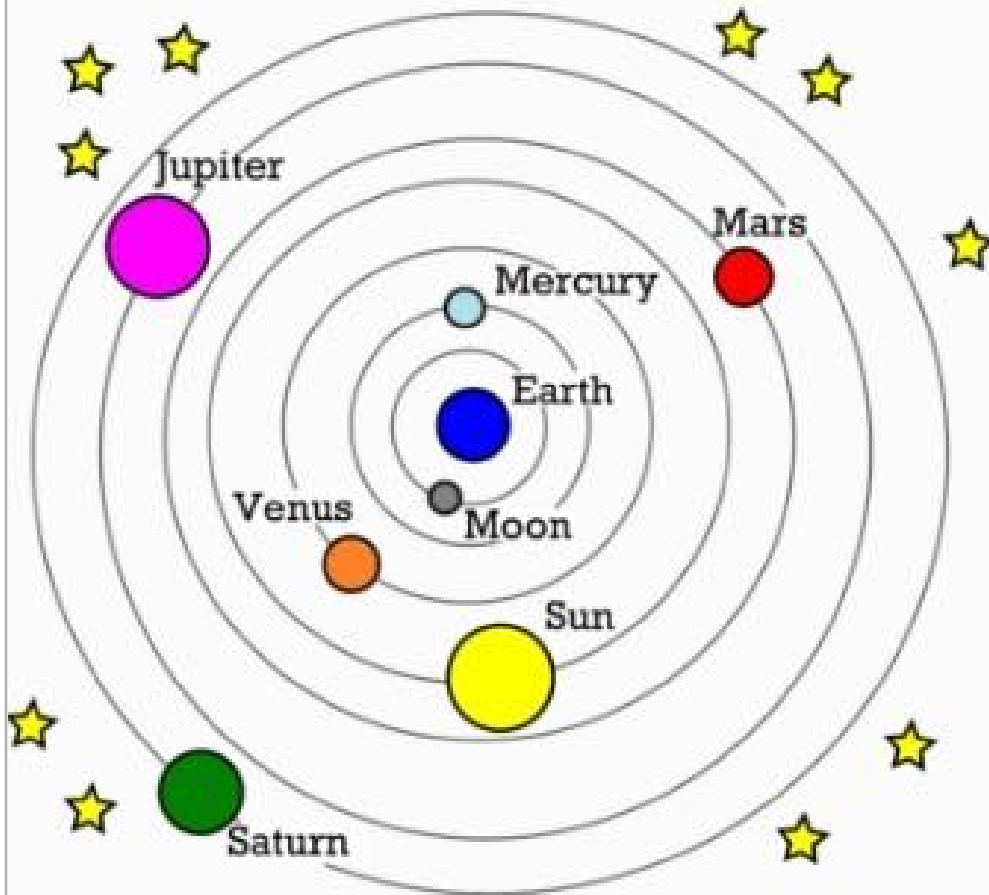
Sciences of the artificial



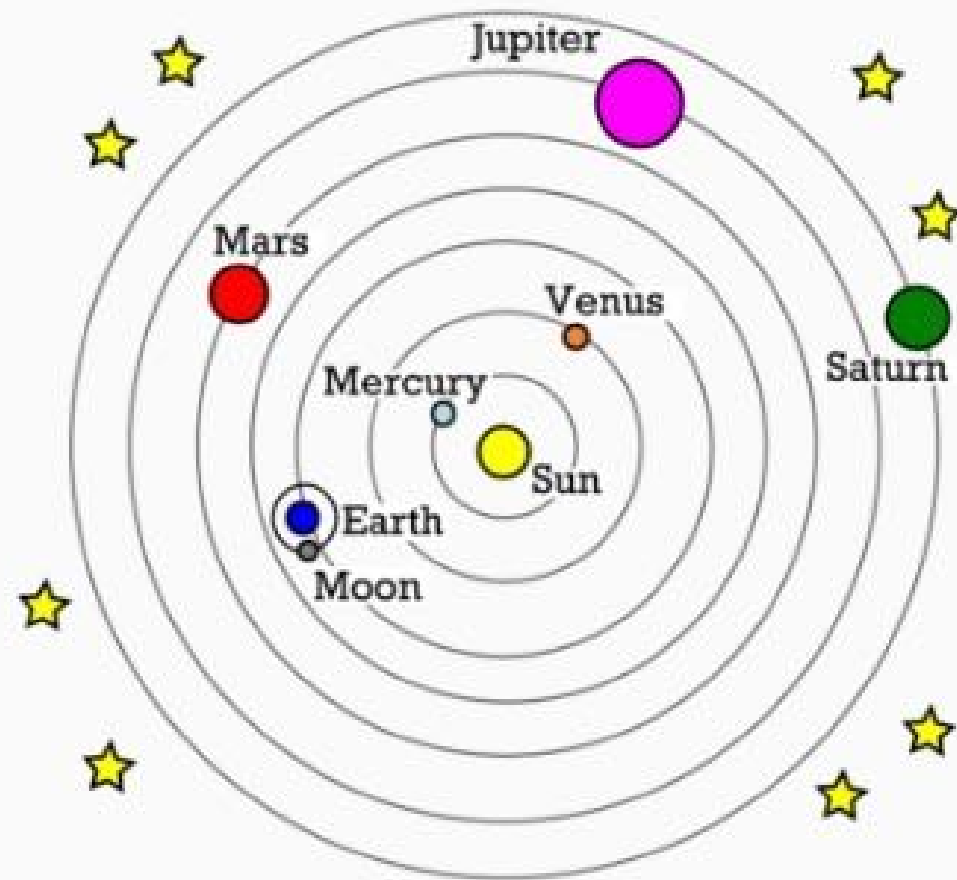
Models and theories



Models and theories

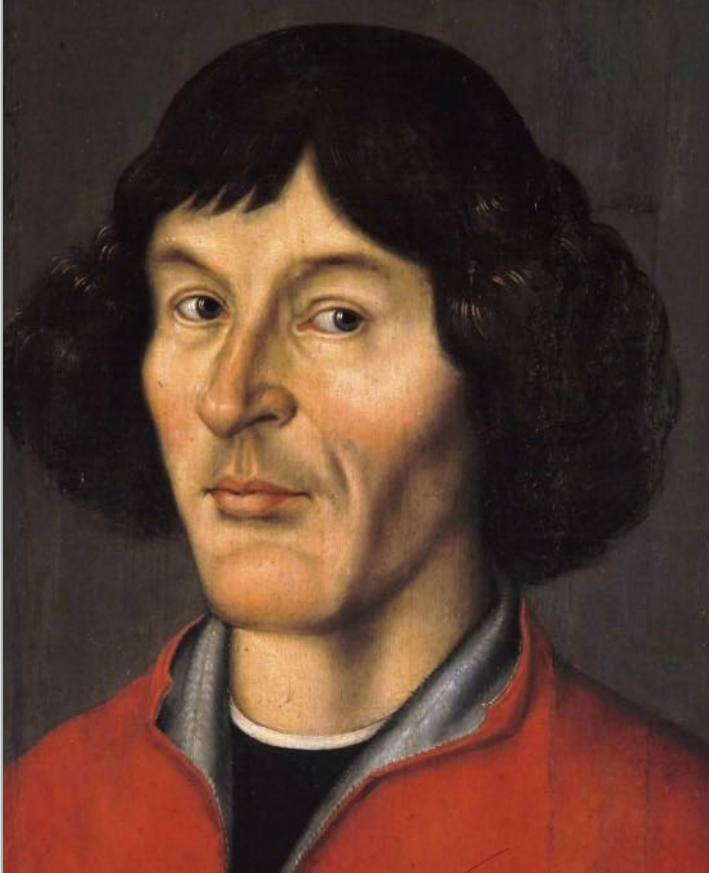


Earth at the Center



Sun at the Center

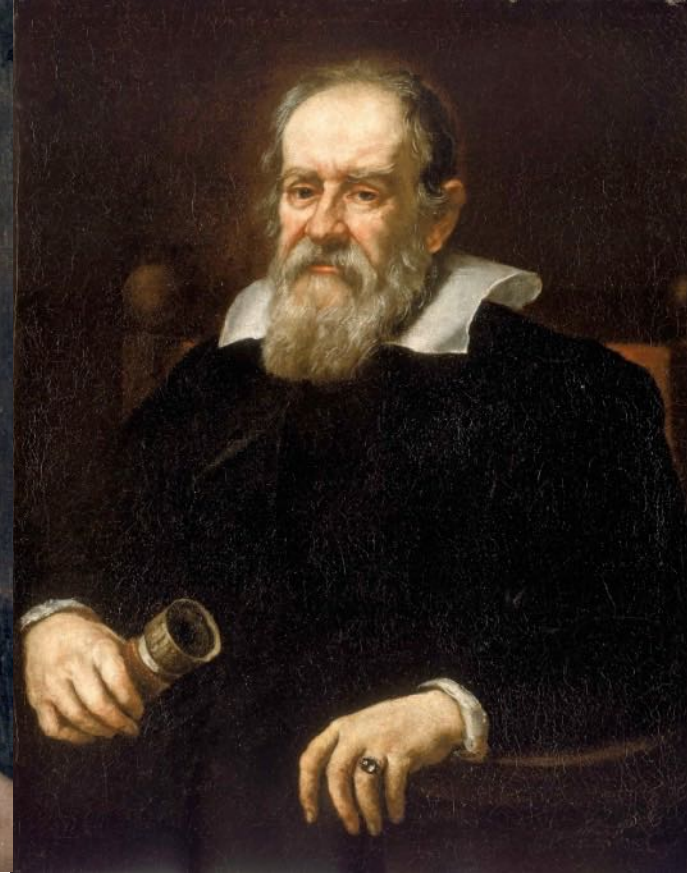
Models and theories



Copernicus (1473-1543)

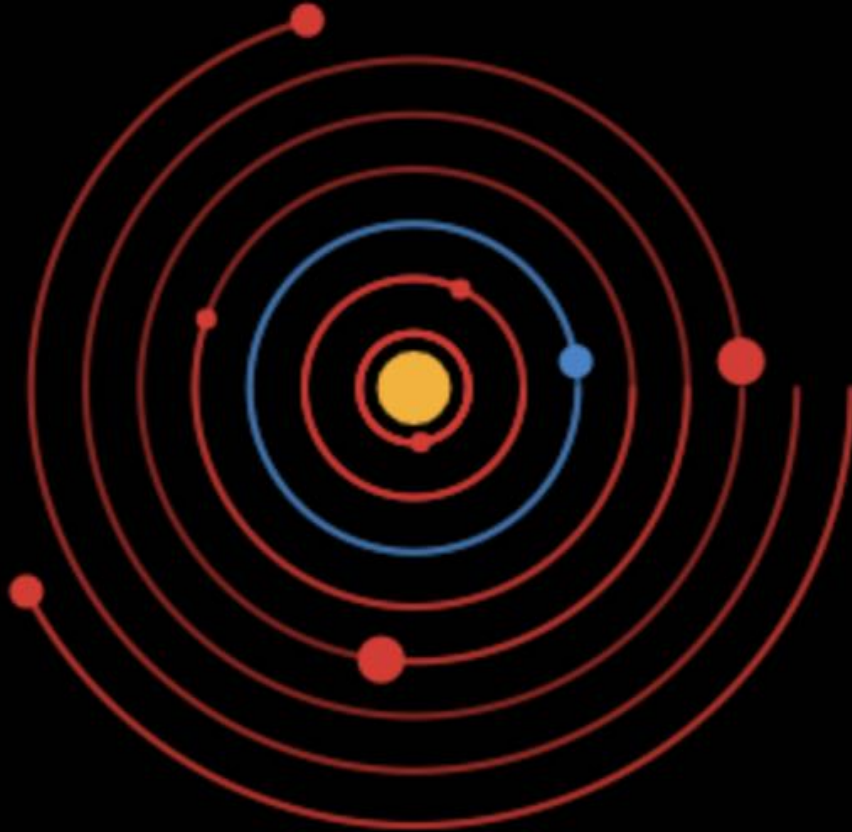


Kepler (1571-1630)

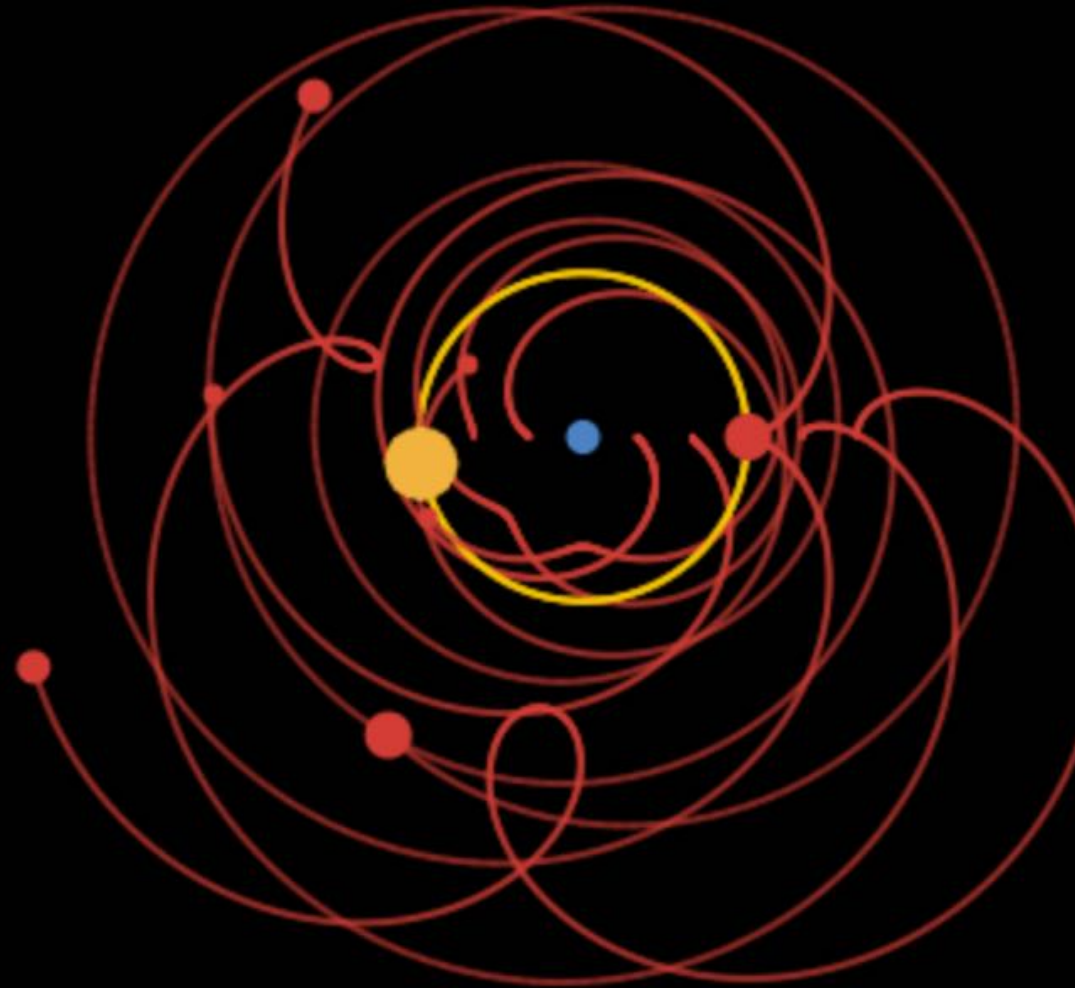


Galileo (1564-1642)

Models and theories



Heliocentrism



Geocentrism

What is a model?

Model = simplification of reality

- Goal: to be useful!
- Abstraction of reality: omit non-relevant details
- Conflict between precision and generality:
choose the level of abstraction

Power of a model

- Descriptive: ability to represent (aspects of) a phenomenon
- Predictive: ability to anticipate behavior
- Generative : ability to imagine new solutions to a problem

Notation = description language

- Not a model
- Informal, incomplete, inconsistent
- Example : UAN (User Action Notation)

What is a theory?

Theory = (attempt to) explain reality

- Often based on a model
- Validity not only of the predictions of the model, but also of the model itself

Falsifiability [Popper]

- A scientific theory must be disprovable through experiments
- A falsified theory can be refined into a “better” theory
 - Example : Newton -> Einstein
Relativity refines (and includes) classical mechanics

Empirical law = observation of a regularity, without explanation

- Not a theory

Perception and action

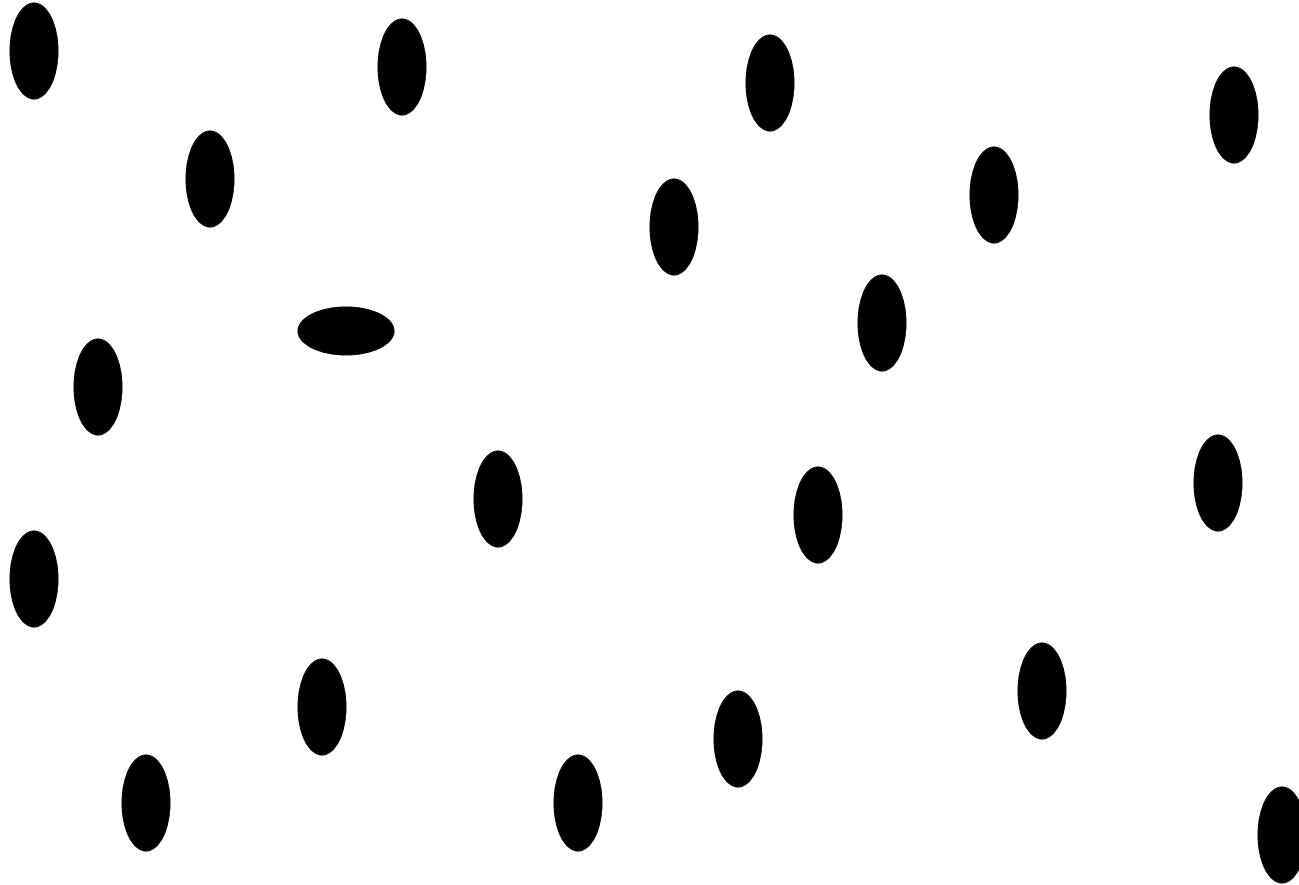
Pre-attentive perception [Triesman]

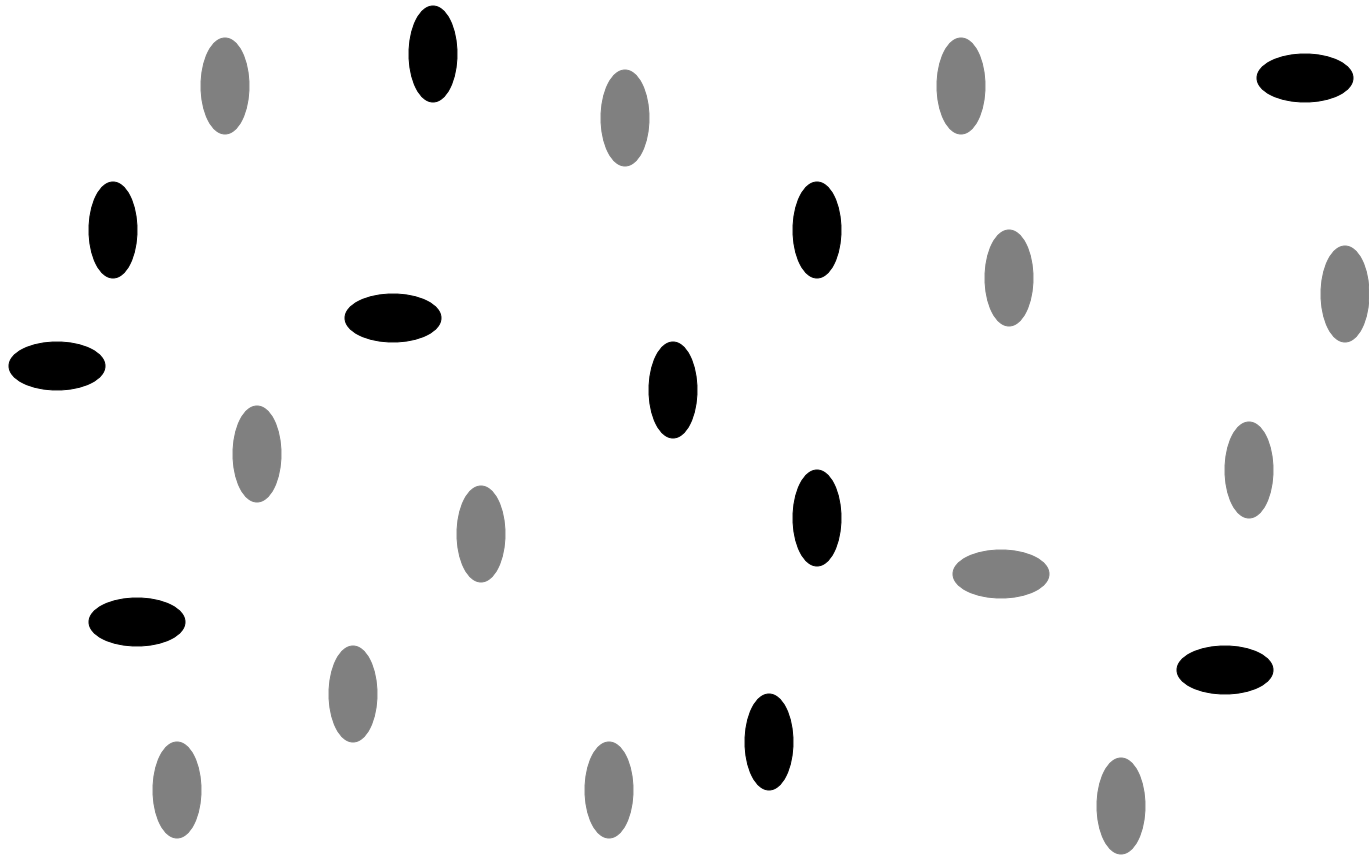
Ecological theory of perception [Gibson]

Hick's law

Fitts' law

Kinematic chain theory [Guiard]





Pre-attentive perception

Observation :

- Humans can recognize some visual features very rapidly:
- Line orientation, blobs, length, thickness, size, curvature, cardinality, endings, intersections, inclusion, hue, blinking, movement direction, depth, direction of light source...

Pre-attentive attributes

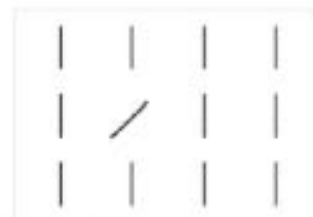
Form



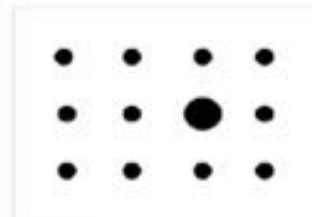
Length



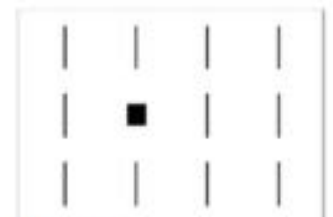
Width



Orientation



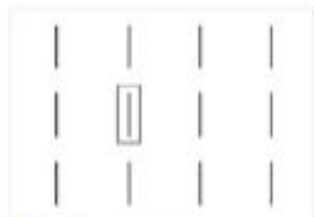
Size



Shape



Curvature



Enclosure



Blur

Color



Hue



Intensity

Position



2-D position



Spatial Grouping

Motion

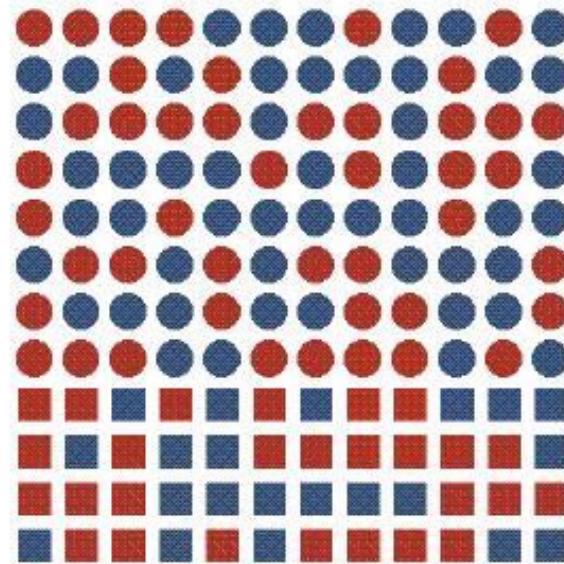
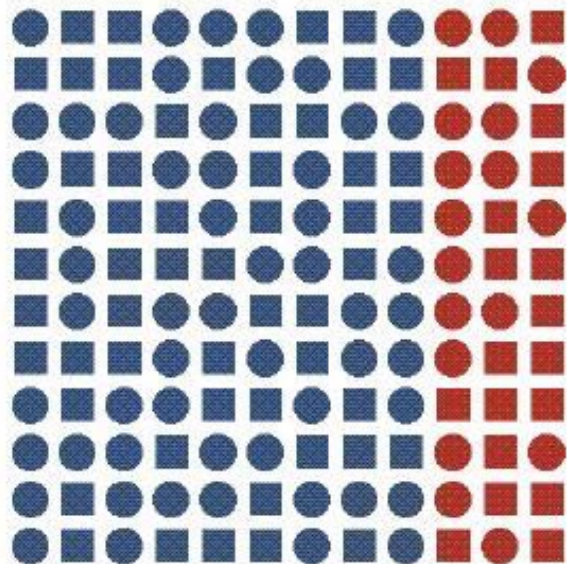


Direction of Motion

Pre-attentive perception

Observation :

- Humans can recognize some visual features very rapidly:
- Line orientation, blobs, length, thickness, size, curvature, cardinality, endings, intersections, inclusion, hue, blinking, movement direction, depth, direction of light source...
- There are interferences when combining several such changes



Pre-attentive perception

Observation :

- Humans can recognize some visual features very rapidly:
- Line orientation, blobs, length, thickness, size, curvature, cardinality, endings, intersections, inclusion, hue, blinking, movement direction, depth, direction of light source...
- There are interferences when combining several such changes

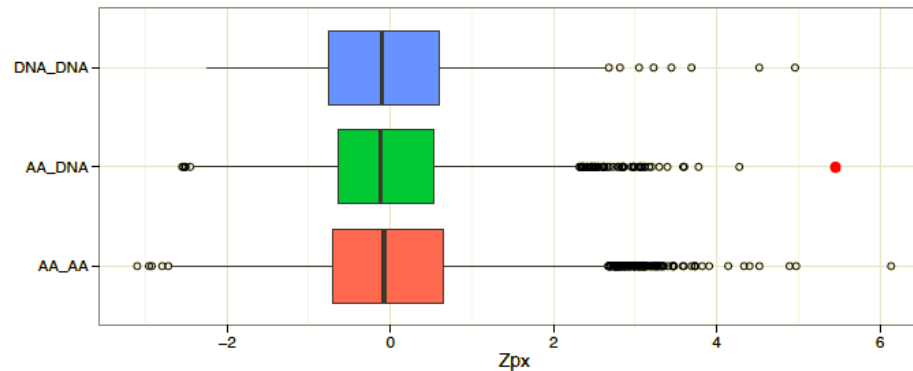
Theory : pre-attentive perception (Triesman, 1985)

- Parallel handling at the level of visual perception
- Information that is not perceived pre-attentively must be handled sequentially
- Links with Gestalt theory

Pre-attentive perception: application

Information visualization:

Use pre-attentive attributes to emphasize outliers

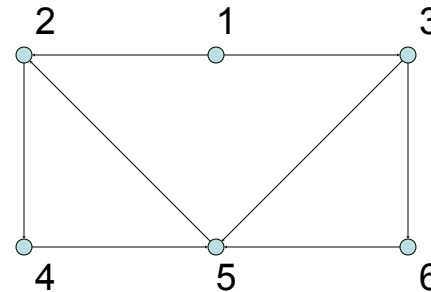
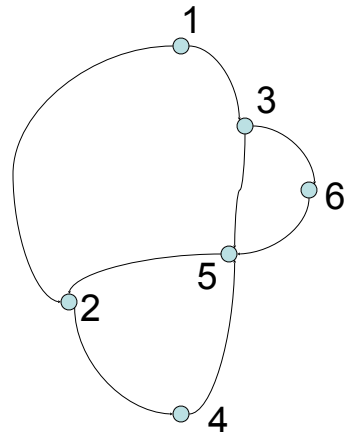
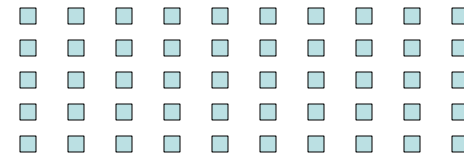
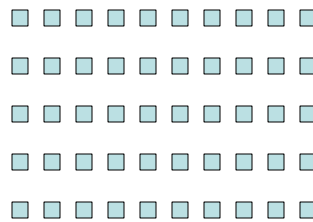
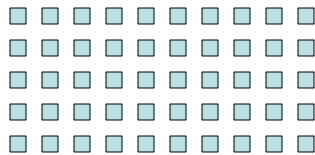
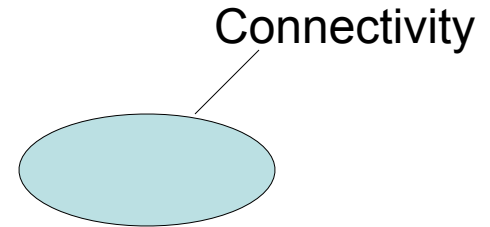
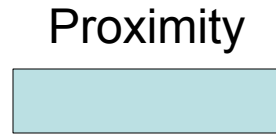
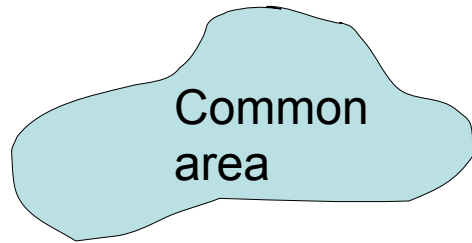


User Interface design:

Use color or other pre-attentive attribute to draw attention

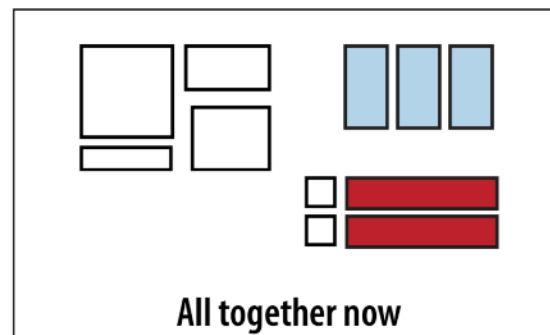
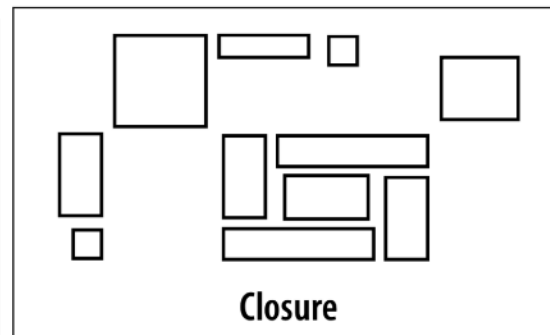
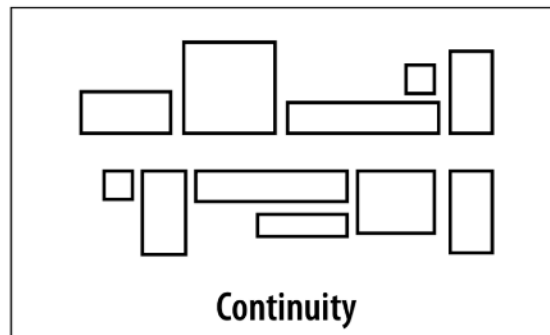
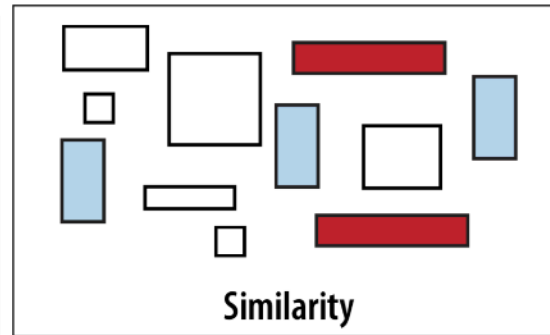
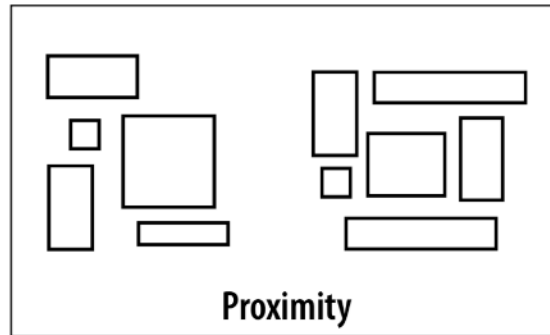
27107462144654207079014738109743897010971
 45929078059772098775972655665110049836645
 25747072354745666142018774072849875310665
 43907097349266847858715819048630901889074

Principles of Gestalt perception



Principles of Gestalt perception: application

Principles of Gestalt perception: application



Ecological Theory of Perception

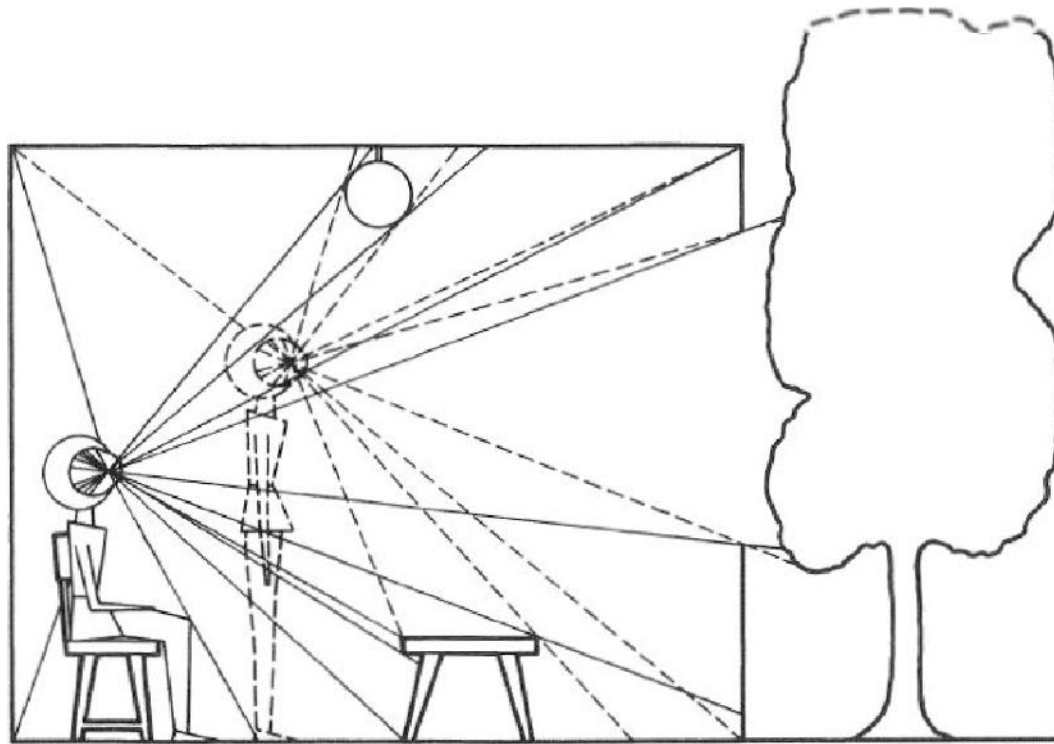
Fundamental hypotheses:

- Co-evolution between organism and its environment
- Behavioral pre-adaptation
- “Elegant” (and parsimonious) perceptual processes

Ecological Theory of Perception

Ecological optics

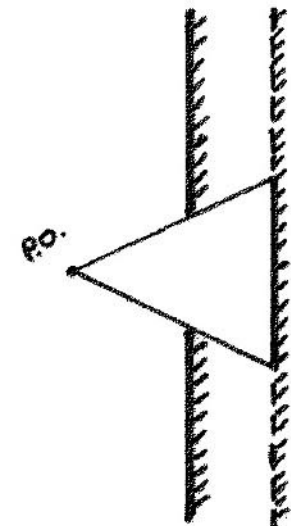
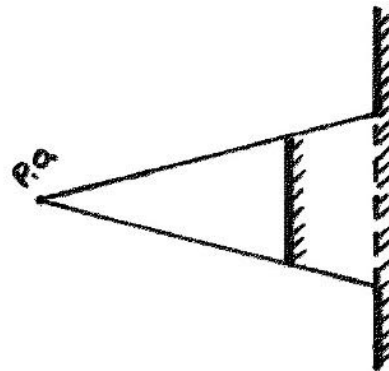
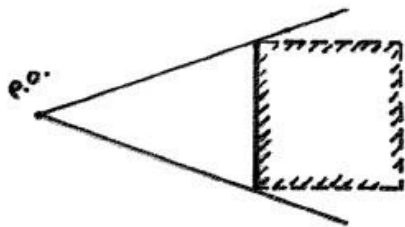
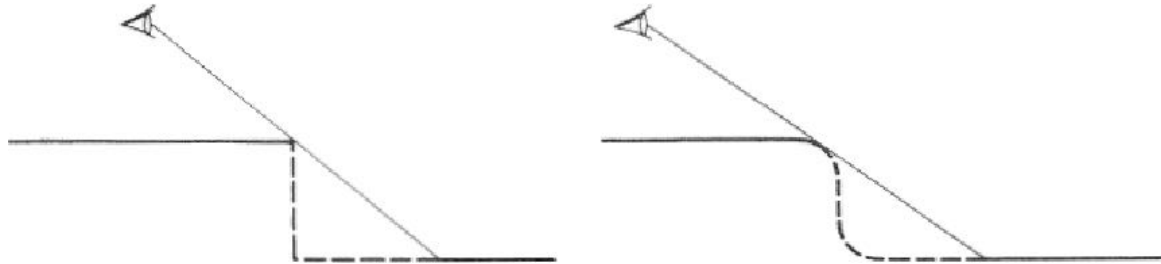
- Information is in the optical array



Ecological Theory of Perception

Ecological optics

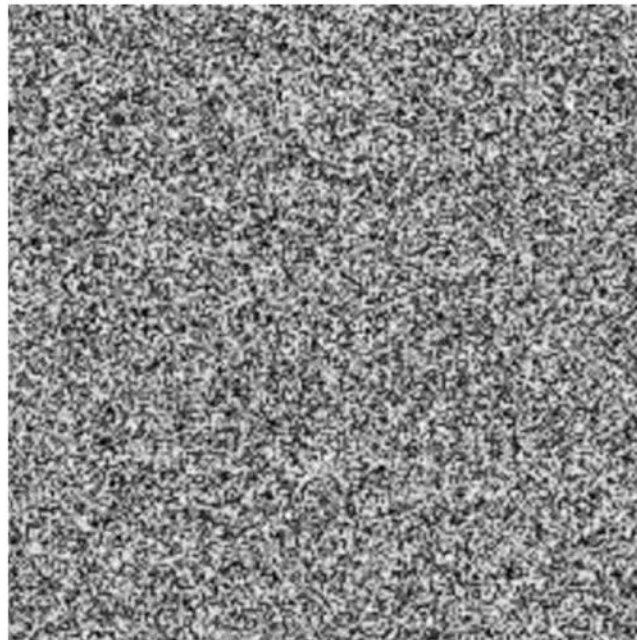
- Information is in the optical array
- An occluding edge reveals depth information



Ecological Theory of Perception

Ecological optics

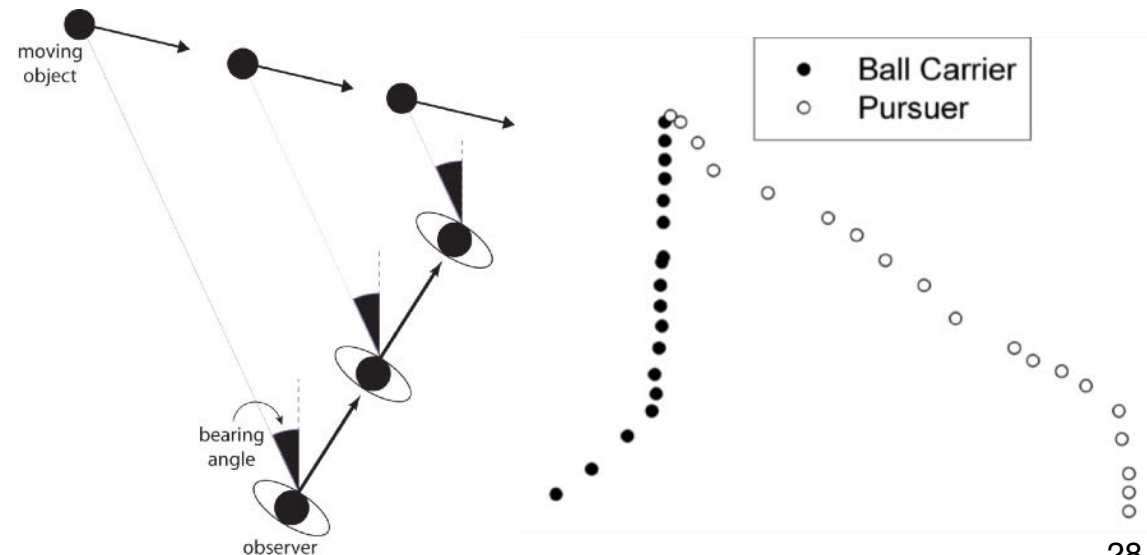
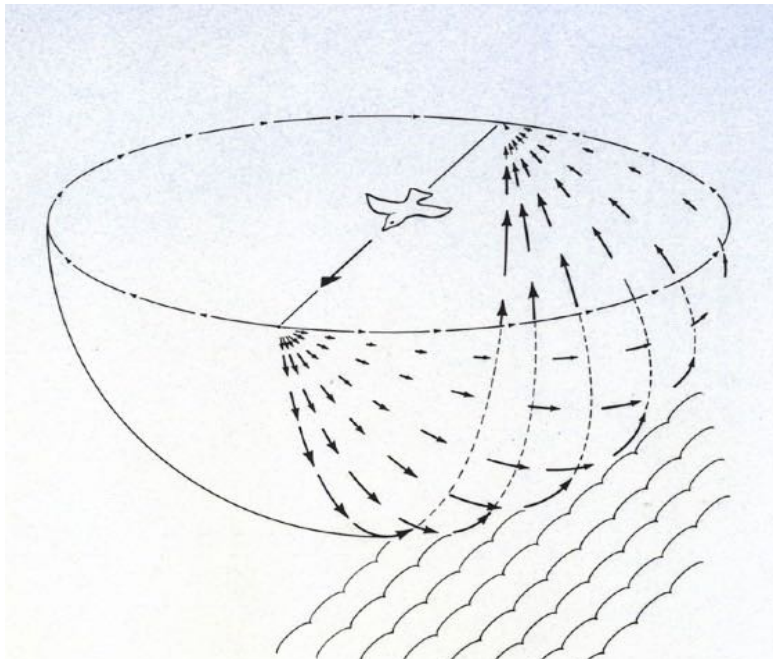
- Information is in the optical array
- An occluding edge reveals depth information
- Information is in the optical flow



Ecological Theory of Perception

Ecological optics

- Information is in the optical array
 - An occluding edge reveals depth information
 - Information is in the optical flow
 - The organism is equipped to extract invariants
- When moving, the only fixed point indicates the direction of motion
 - To catch a ball, keep a constant angle between the ball and your direction of motion



Ecological Theory of Perception

Ecological optics

- Information is in the optical array
- An occluding edge reveals depth information
- Information is in the optical flow
- The organism is equipped to extract invariants

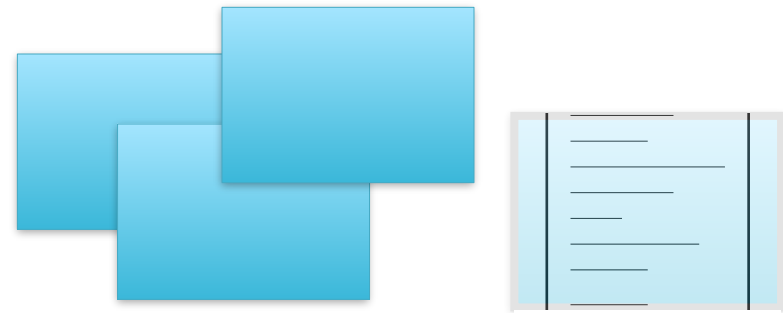
Relativity of the environment

- Action-perception coupling
 - One must move to perceive, and perceive to move
- Perception of affordances
 - Capabilities of the environment for actions

Ecological Theory of Perception: applications

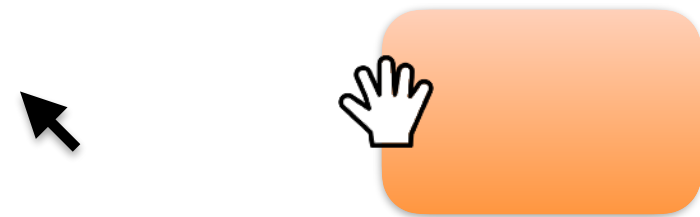
Ecological optics

- Overlapping windows
- Scrolling



Action-perception coupling

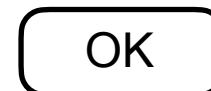
- Perceive to act: Feedback
- Act to perceive: Feedforward



Affordances

- Objects should reveal their capabilities for action

[Add comment](#)



Hick's law (1952) - also called Hick-Hyman law

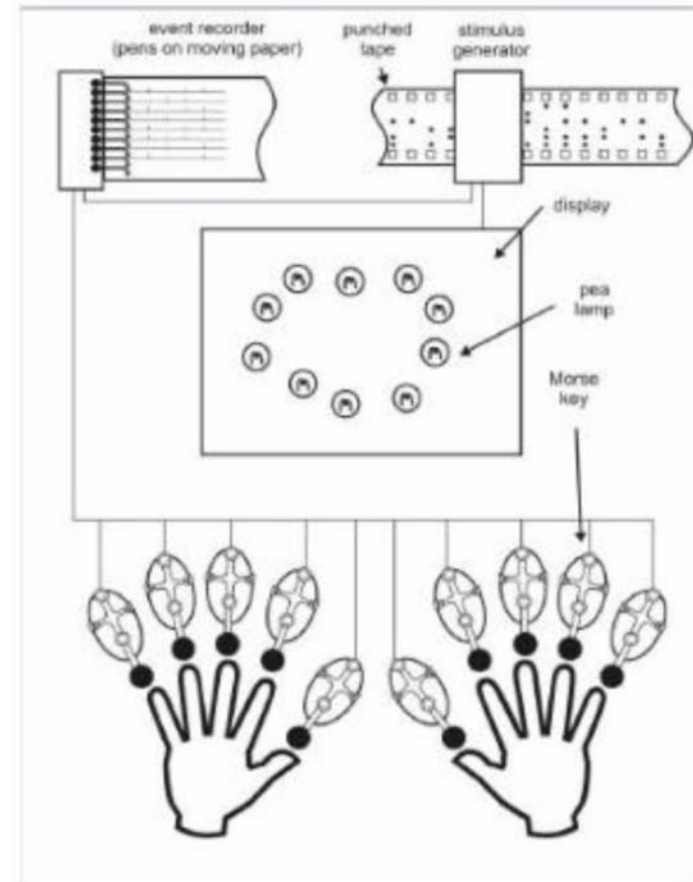
Choice reaction time: time it takes to identify an item in a set

- $RT = a + b H$
 a & b are constants
 H is the entropy of the set of choices
- $H = \log_2 (n)$ if all choices are equiprobable
 where n is the number of items
- $H = \sum p_i \log_2 (1/p_i)$
 where p_i is probability of choice i

Was performed with highly trained participants

(Hick trained himself over 8,000 trials!)

=> strong effect of learning

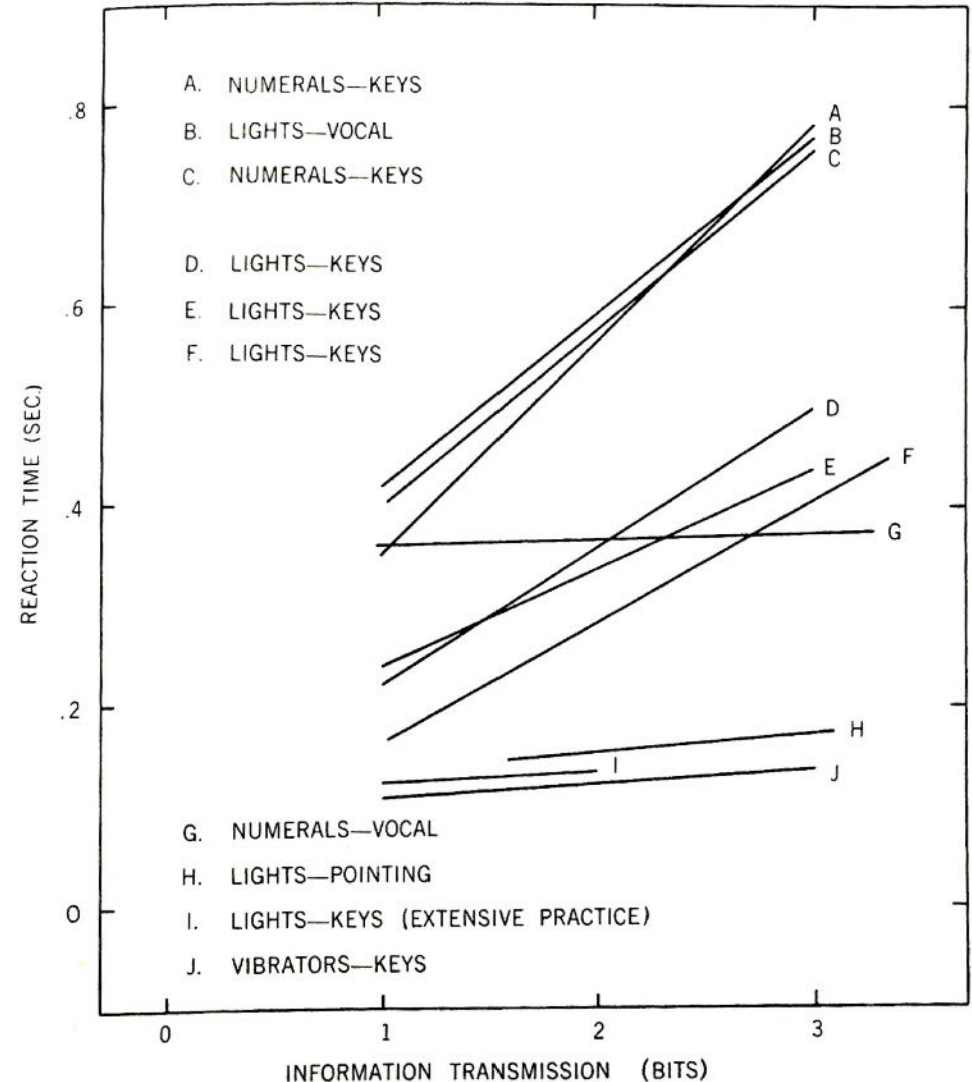


Hick's law (1952) - also called Hick-Hyman law

Effect of stimulus-response compatibility

Performance higher if, for example, the lights are aligned with the key to be pressed

Slope approaches 0 for very high S-R compatibility, e.g. spelling out a letter



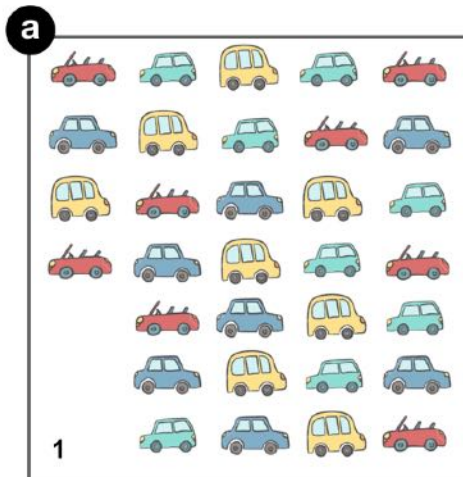
Hick's law (1952) - applications (or not!)

Often used to justify the design of menu systems

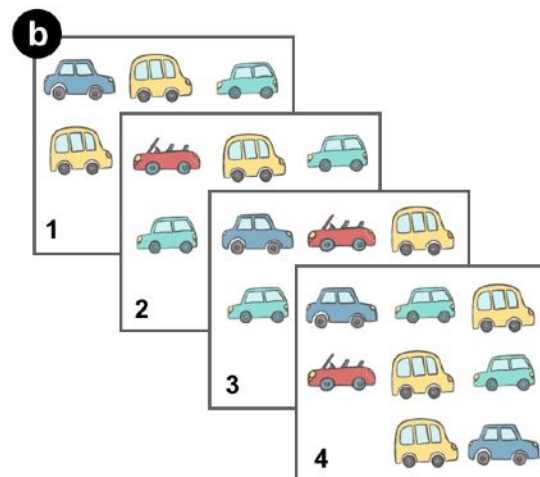
- Reduce number of choices in each level
- Hierarchical menu better than flat menu

BUT in general not true!

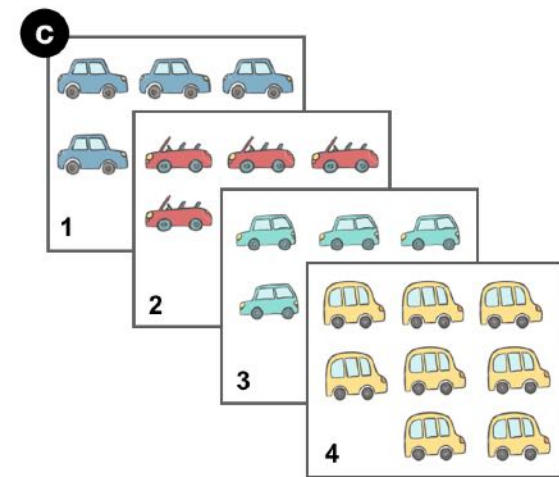
- High S-R compatibility and over-learned tasks
=> almost constant reaction time
- Other factors at play: Hick's law not relevant



$$RT = a + 5b$$



$$RT = 2.5a + 7.5b$$

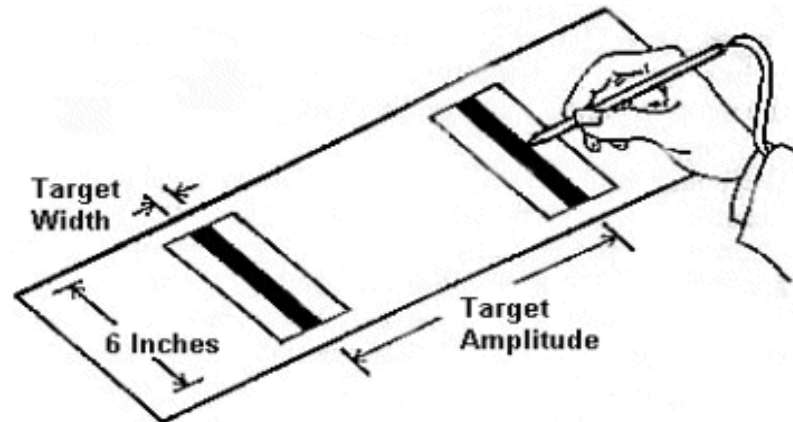


$$RT = 2a + 5b$$

Fitts' law (1954)

Fitts' law: time it takes to acquire a target

- $MT = a + b \log_2 (1 + D/W)$
 - a & b are constants
 - D = distance to target (amplitude)
 - W = pointing tolerance (width of the target)
- Information-based theory of perception



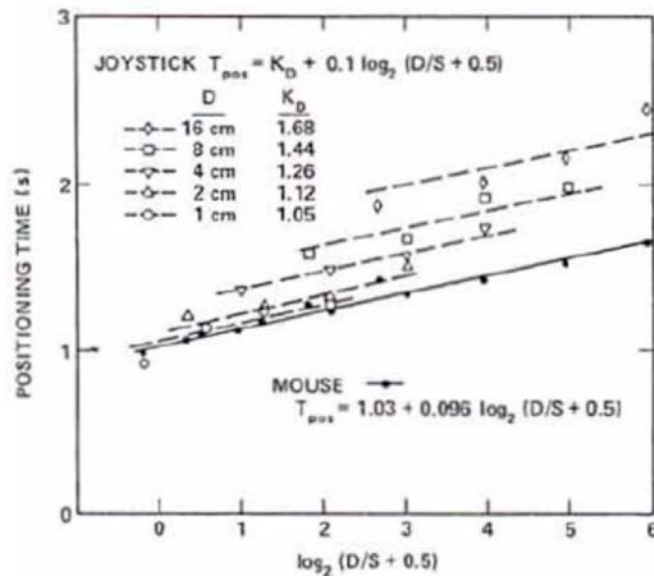
Empirical laws extracted from controlled observations

These laws are valid only in precise experimental settings

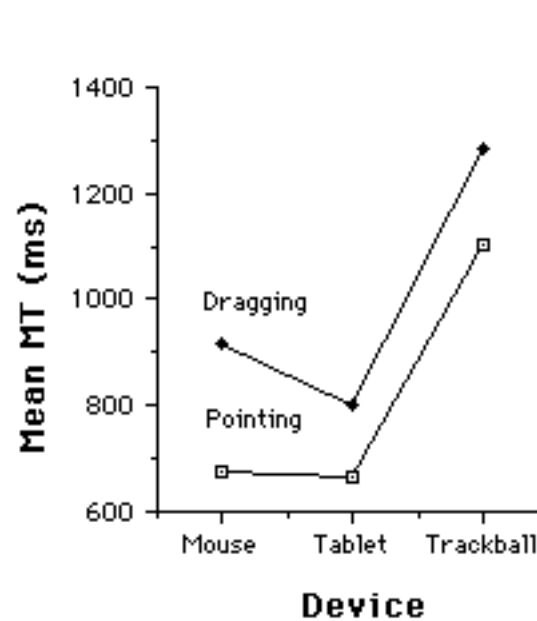
Fitts' law (1954) - applications

Assessing pointing devices

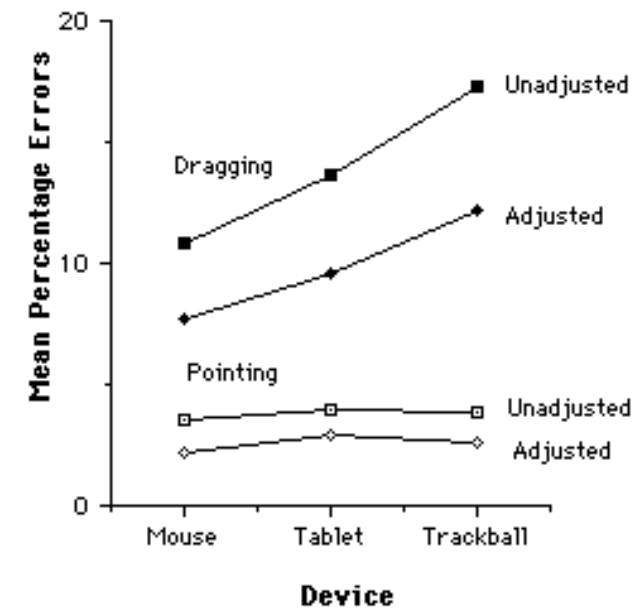
- Comparing a and b constants and error rates across devices



Card, English & Burr 1978



Mackenzie, Sellen & Buxton 1991



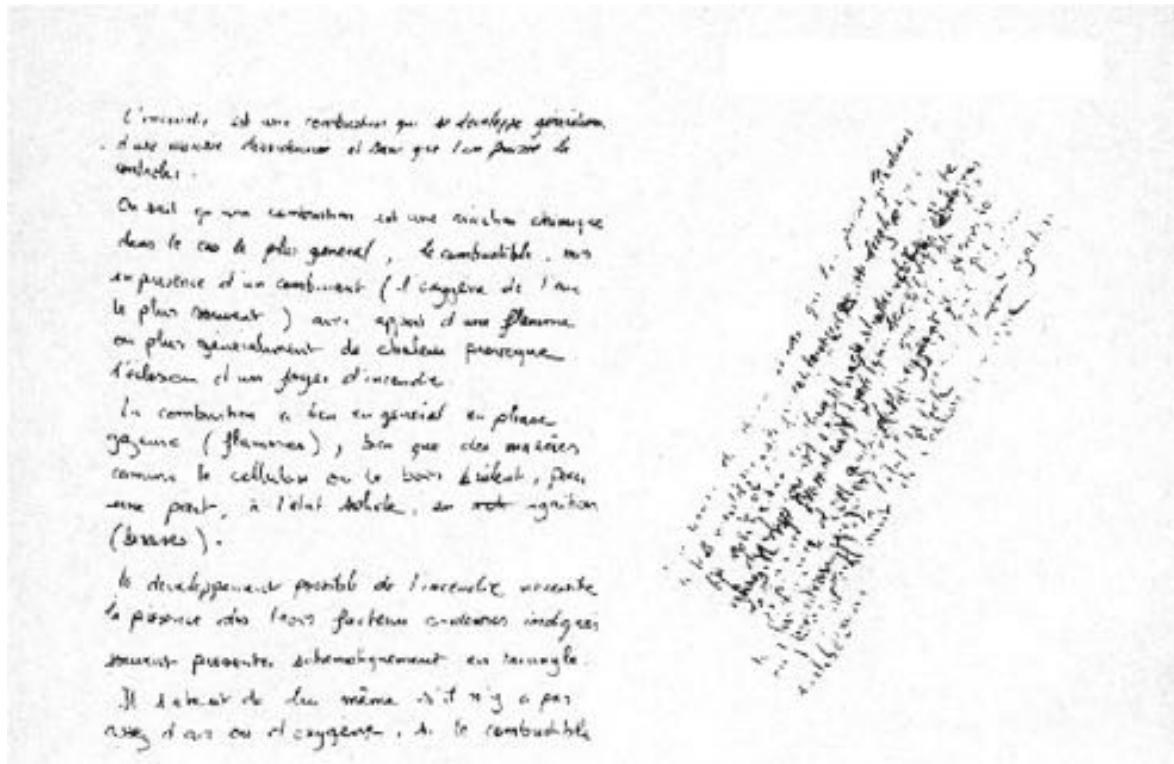
Creating and evaluating new pointing techniques

- Drag-n-pop, Semantic pointing, Bubble cursor, ...

Kinematic chain theory

Laterality of motor control

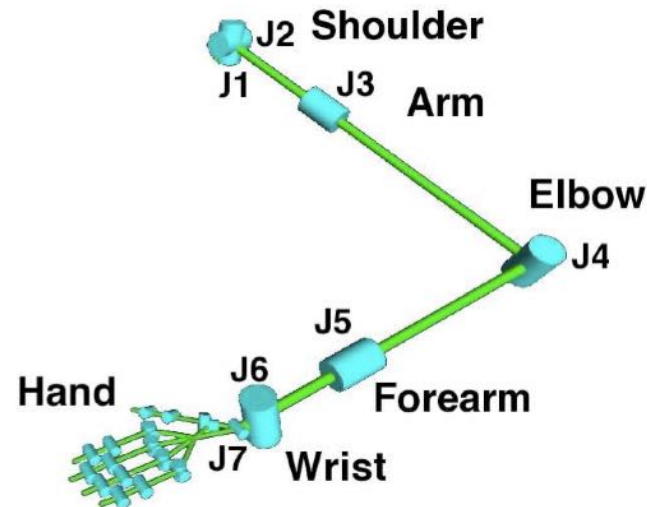
- Classical psychology:
“the left hand is a bad right hand”
- Observations of bimanual control:
the two hands have different *roles*



Kinematic chain theory

Kinematic chain:

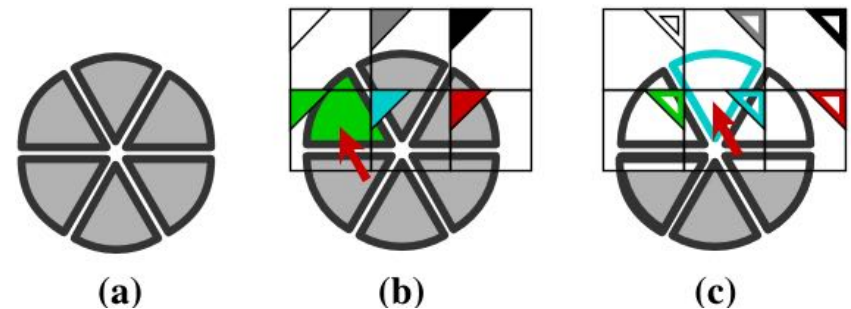
- Non-dominant hand: distal control
 - Acts first
 - Establishes the frame of reference (context) for the dominant hand
 - Movements do not need to be precise
- Dominant hand: proximal control
 - Acts after the non-dominant hand, within the frame of reference it establishes
 - Precise movements



Kinematic chain theory: applications

Bimanual interaction

- First study by Myers & Buxton (1986)
Selection and navigation task
- Bier et al. (1993)
Toolglasses and Magic Lenses
- Fitzmaurice, Ishii & Buxton (1995)
Graspable interfaces



Toolglasses and Magic Lenses,
Bier et al., SIGGRAPH 1993

Graspable interfaces



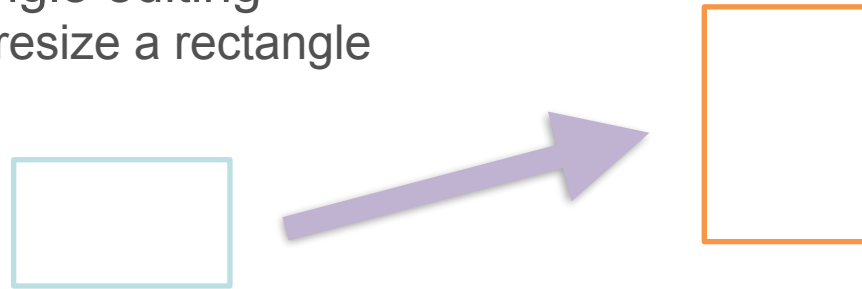
Kinematic chain theory

Falsification:

- Some tasks are more efficient when the hands have symmetric roles

Example: Rectangle editing

- Move and resize a rectangle



- Asymmetric assignment:
 - non-dominant hand moves the rectangle
 - dominant hand adjusts the size
- Symmetric assignment:
 - one hand moves one corner, the other hand the other corner
- Symmetric assignment more efficient

Cognition and behavior

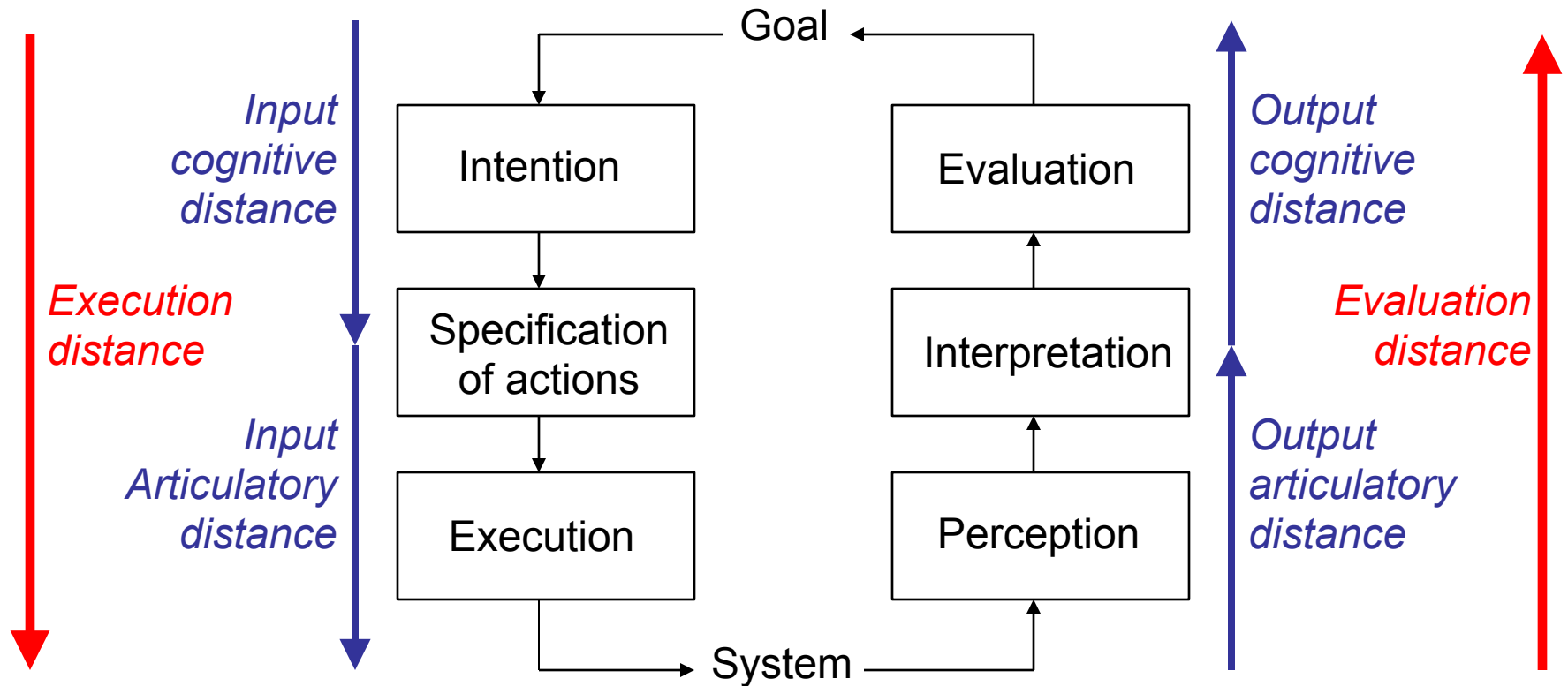
Seven stages of action [Norman]

Situated action [Suchman]

Activity theory [Vigotsky, Bødker]

Cognitive dimensions [Green]

Seven stages of action



Seven stages of action - applications

Analyze a design to assess the *gulf of execution*:

Input cognitive distance: what can I do?

=> make functionalities visible

Input articulatory distance: how can I do it?

=> feedforward, good mapping

Analyze a design to assess the *gulf of execution*:

Output articulatory distance: what happened?

=> feedback & response

Output cognitive distance: what does it mean?

=> good mapping, metaphor

Situated action

Classical cognitivist approach:

- Cartesian model where all actions are planned and human action is explained by cognitive processes
- Examples : action theory, task analysis, mental models

Ethnomethodological approach:

- Detailed analysis of work practices in order to determine the causal chains implied by the observed actions

Situated action:

- Human action takes place in a complex context that creates constraints and dependencies and affects the actions being undertaken
- If there is a plan, at best it is used as a guide
- Action adjusts to the context at hand and at the same time modifies it

Situated action: Ethnography and the PARC copier



Situated action - applications

Study your users!

Design for actual use in context,
not theoretical use

Observe breakdowns: When things don't go as expected

=> redesign to reduce causes of errors

Observe workarounds: How users find alternative ways

=> flexibility provides more opportunities

Observe user innovations: How they appropriate technology

=> unusual use becomes a feature

Activity theory

Vigotsky: analysis of human activity

- Subject-object relationship is mediated by tools (technical instruments) or signs (psychological instruments)

Leontiev : emphasis on the role of the community

- Rules and rituals, division of labor

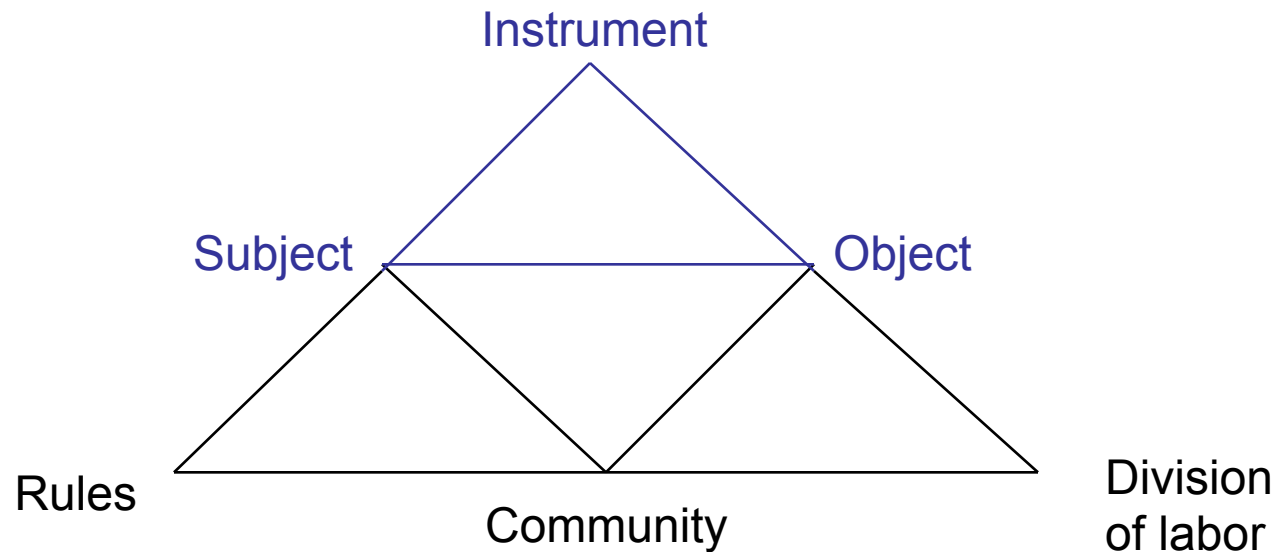
3 levels of activity:

- Activity: responds to a need (materialistic or intellectual) – Why
- Actions: executed consciously to reach an explicit goal set by the subject – What
- Operations: executed unconsciously or semi-consciously to execute actions – How

Activity theory

Levels of activity:

- Action -> operation: automation / internalisation
- Operation -> action: conceptualisation (e.g., in case of failure)
- Activity -> action: according to the context



Activity theory - applications

Activity-centric computing

Instead of a task-oriented analysis of practice, analyze the *activities* people engage in and the *tools* they use to conduct them

Analyze the division of labor







Support collaborative activities with different roles

Analyze artifact ecologies

Support combinations of devices (digital and physical)

Apple Continuity

“All your devices, one seamless experience”

-  - Use iPhone as camera
-  - Use mouse/keyboard for tablet
-  - Use tablet for desktop
-  - AirPlay between devices
-  - Get messages across devices
-  - Unlock screen with watch
-  - Answer call on desktop
-  - Airdrop files across devices
-  - Copy-paste across devices
-  - Start editing on one device, finish on another
-  - Use phone as hotspot
- ...



Cognitive dimensions

Notation :

- Tool to help interaction designers
- Evaluating a system according to certain criteria
- Scientific foundation: importance of representation to solve a problem

6 types of activity:

- Incrementation : add data
- Transcription : copy from another source
- Modification : change content, adapt to a new problem
- Exploration : trial and error to find a solution
- Search: look for an object that may not exist
- Comprehension : discover an unknown aspect of the system

Cognitive dimensions

Dimensions : aspects of the informational structure that can be analyzed according to the activity being studied.

Some examples :

- Viscosity: resistance to change
- Visibility: ability to see components easily
- Premature commitment: constraints on the order of actions
- Hidden dependencies: important but hidden links between entities
- Role expressiveness: the role of an entity is easy to infer
- Abstraction : types and availability of abstraction mechanisms
- Consistency : similar semantics are expressed with similar syntax
- etc.

Cognitive dimensions - applications

Walkthrough a design to assess each dimension

Example of viscosity

Changing the travel date when looking for a train

Example of premature commitment

Giving the name of a file when saving it

Example of hidden dependency

Deleting a file when there are aliases pointing to it

Example of abstraction

The notion of style to represent a set of attributes

Interaction

Morphological analysis of input devices [Card et al.]

UAN [Hartson]

GOMS [Card-Moran-Newell]

Instrumental interaction [Beaudouin-Lafon]

Morphological analysis of input devices

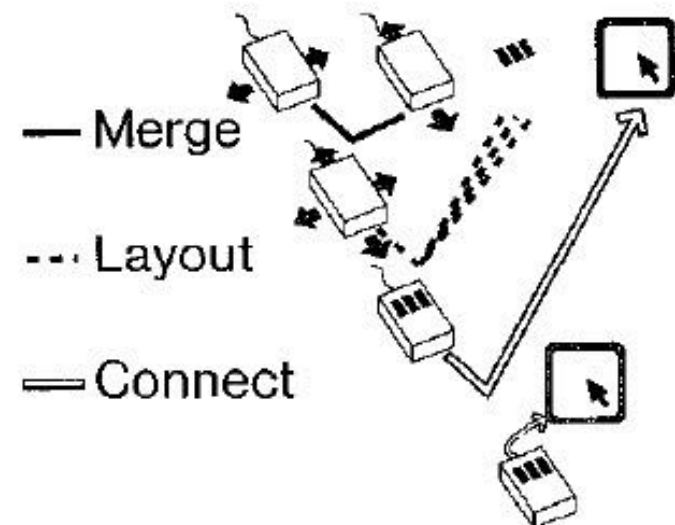
Description of the properties of an input device:

Transducer of physical properties into logical properties

- M = Manipulation operation
 - position/force, absolute/relative => P, F, dP, dF
 - linear/circular => X, Y, Z / rX, rY, rZ
- In = Input domain
- S = Current state of the device
- R = Resolution function: In -> Out
- Out = Output domain
- W = Other properties of interest

Composition of input devices:

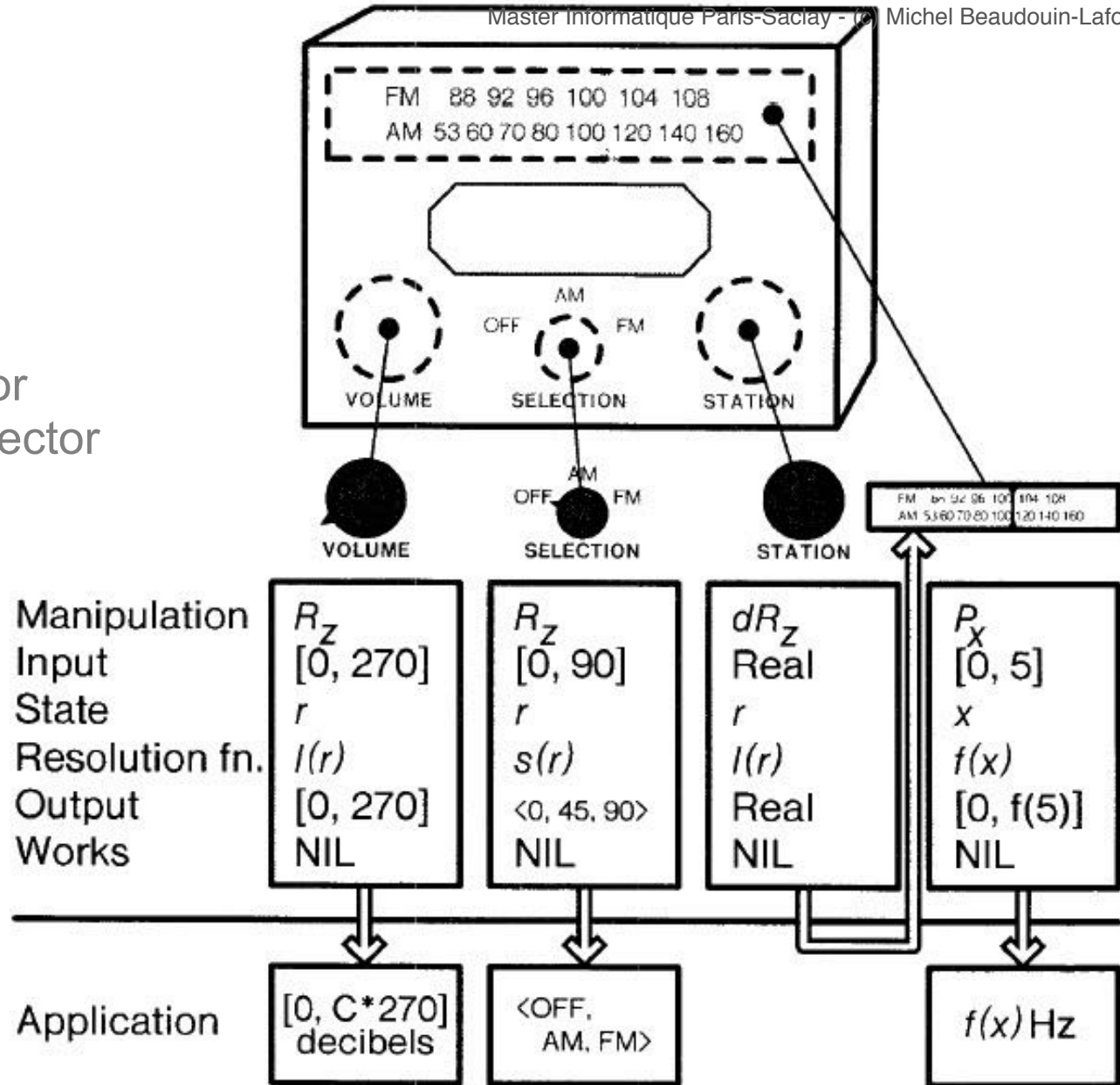
- Merge
- Layout
- Connect



Example

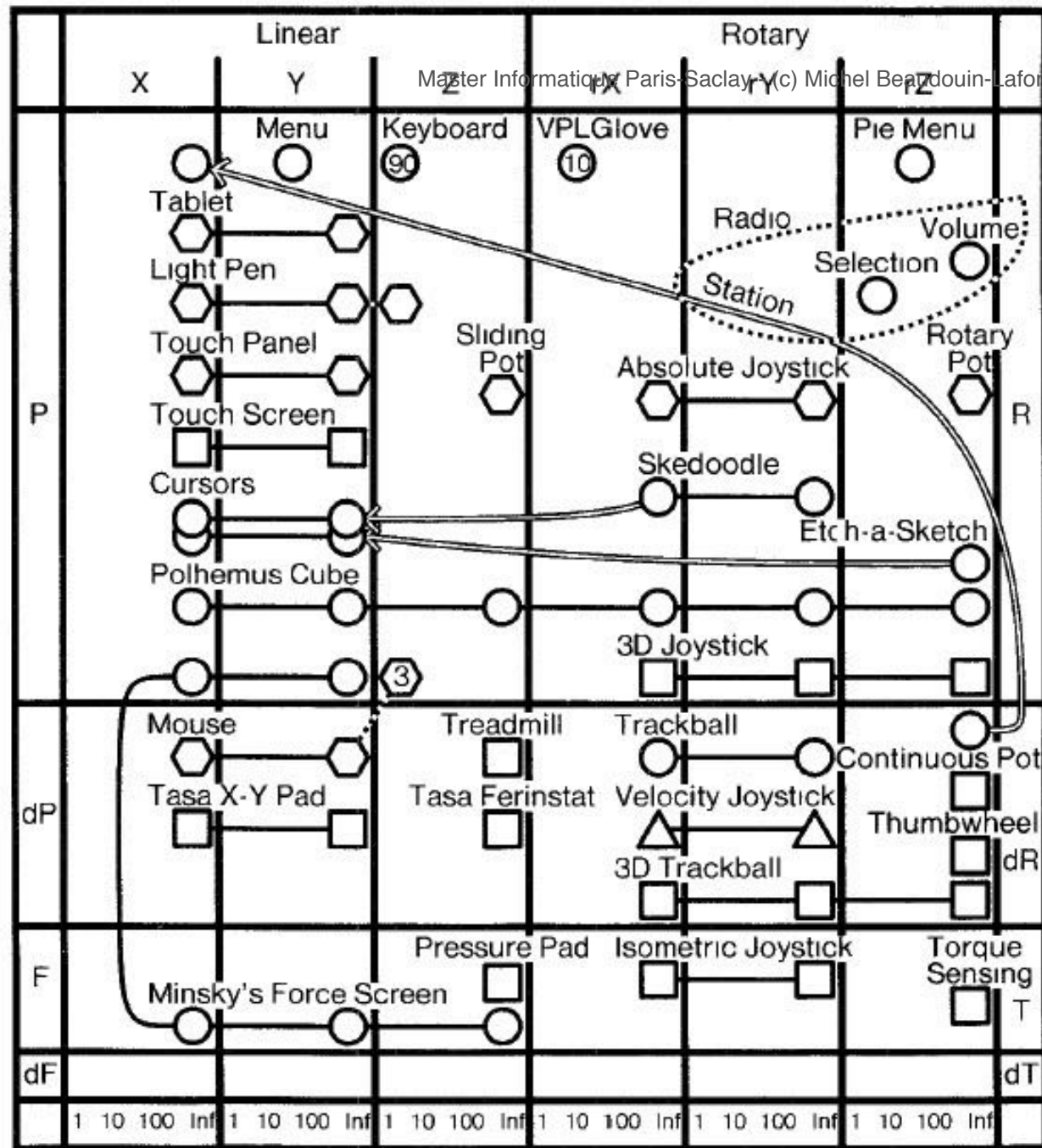
Radio :

- Volume dial
- AM/FM selector
- Frequency selector



Taxonomy

Comparison of input devices, including those studied by Foley and by Buxton



△ Only Foley □ Only Buxton ⬡ Both Foley & Buxton ○ Other

UAN : User Action Notation

Description of user actions and system responses

Example : selecting an icon

Action	Feedback
~[icon] Mv^	icon!

More accurate version:

~[icon] Mv	icon-! : icon! , all icon'! : icon'-!
M^	

Moving an icon:

~[file_icon] Mv	file_icon-! : file_icon! , all icon'! : icon'-!
~[x,y]* ~[x',y']	outline(file_icon) > ~
M^	@x',y' display(file_icon)

UAN

Action	Feedback	Interface state	Computation
~[file_icon] Mv	file_icon-! : file_icon! , all icon'! : icon'-!	selected = file	
~[x,y]* ~[x',y']	outline(file_icon) > ~		
M^	@x',y' display(file_icon)		pos(file_icon) = x',y'

Informal notation

- Usable with a standard keyboard
- Easy to remember
- Separates symbols from their meaning
- Can be extended if needed:
 - New symbols
 - New columns (e.g., cognitive load)

The GOMS family of models

GOMS = Goals, Operators, Methods, Selection rules

- **G**oals: what the user wants to do
- **O**perators: actions supported by the software application
- **M**ethods: learned sequences of subgoals and operators to reach a goal
- **S**election rules: users' personal rules to choose one of several methods

GOMS is both:

- A method to describe user tasks
- A set of descriptive (and sometimes predictive) models, used at several levels of abstraction

Task analysis techniques based on models of information processing

Example : move a sentence in a text

Initial goal: edit text

Sub-goal: select text to move

Operators:

- a. move the mouse
- b. clic mouse button
- c. enter key on keyboard

Methods:

- For editing:
 - 1. Delete sentence and type again
 - 2. Cut-paste using keyboard shortcuts
 - 3. Cut-paste using menu items
- For selection :
 - 4. Click and drag text
 - 5. Double-click first word, shift-click last word

Selection rules:

- For editing: method 1 if the text is short, method 2 if the user knows the shortcuts, method 3 otherwise.
- For selection: method 4 if the text to be moved is not a set of complete words, method 5 otherwise.

KLM : Keystroke-Level Model

Operators in the original version:

- K – hit key or button (0.08s - 1.20s, mean 0.40s)
- P – pointing a target with the mouse (1.10s)
- H – Homing = moving hand between mouse and keyboard (1s)
- D – Drawing a line segment ($0.9n + 0.16l$, n segs de long. l)
- M – Mental activity to prepare for next action (1.35s)

“Magical” rules for placing operator M

Example : Method 5 then 3

- Selection: M PK PK
 - Copy command: M PK PK
 - Select destination: M PK
 - Paste command: M PK PK
- total = 14.9s

CMN-GOMS : Card-Moran-Newell GOMS

Evolution of the Keystroke-level model

- Some additional operators
- Computer support
 - Automatic evaluation of predicted times
 - Automatic evaluation of selection rules

Predictive model (as is KLM)

- Helps compare various methods for a single task
- Example : shows that the selection rule for moving the cursor with the mouse vs. the keyboard tends to choose the optimal method.

Problem: tendency to overestimate execution times

- Operators have a fixed duration
- Learning is not taken into account

CPM-GOMS : Critical-Path Method

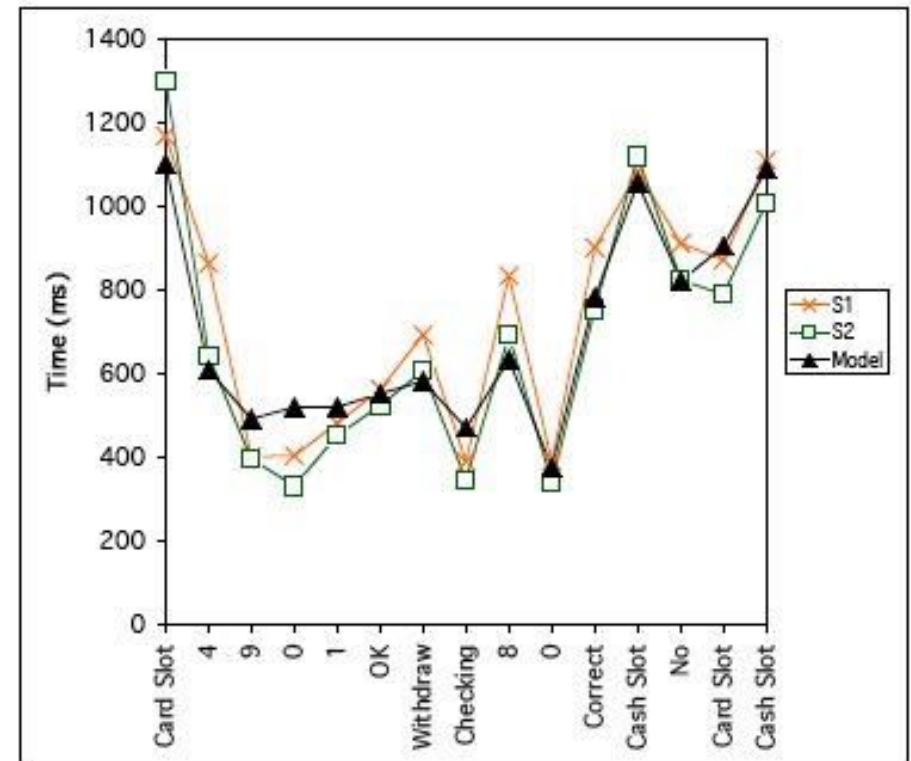
Based on the Model Human Processor (MHP)

- Parallel processing of perceptual, cognitive and motor activities
- PERT diagram created from the CMN-GOMS description of the task using templates of MHP operators for elementary tasks

Predictive power:

- Performance prediction is more accurate than KLM
- Qualitative analysis using the critical path in the PERT diagram

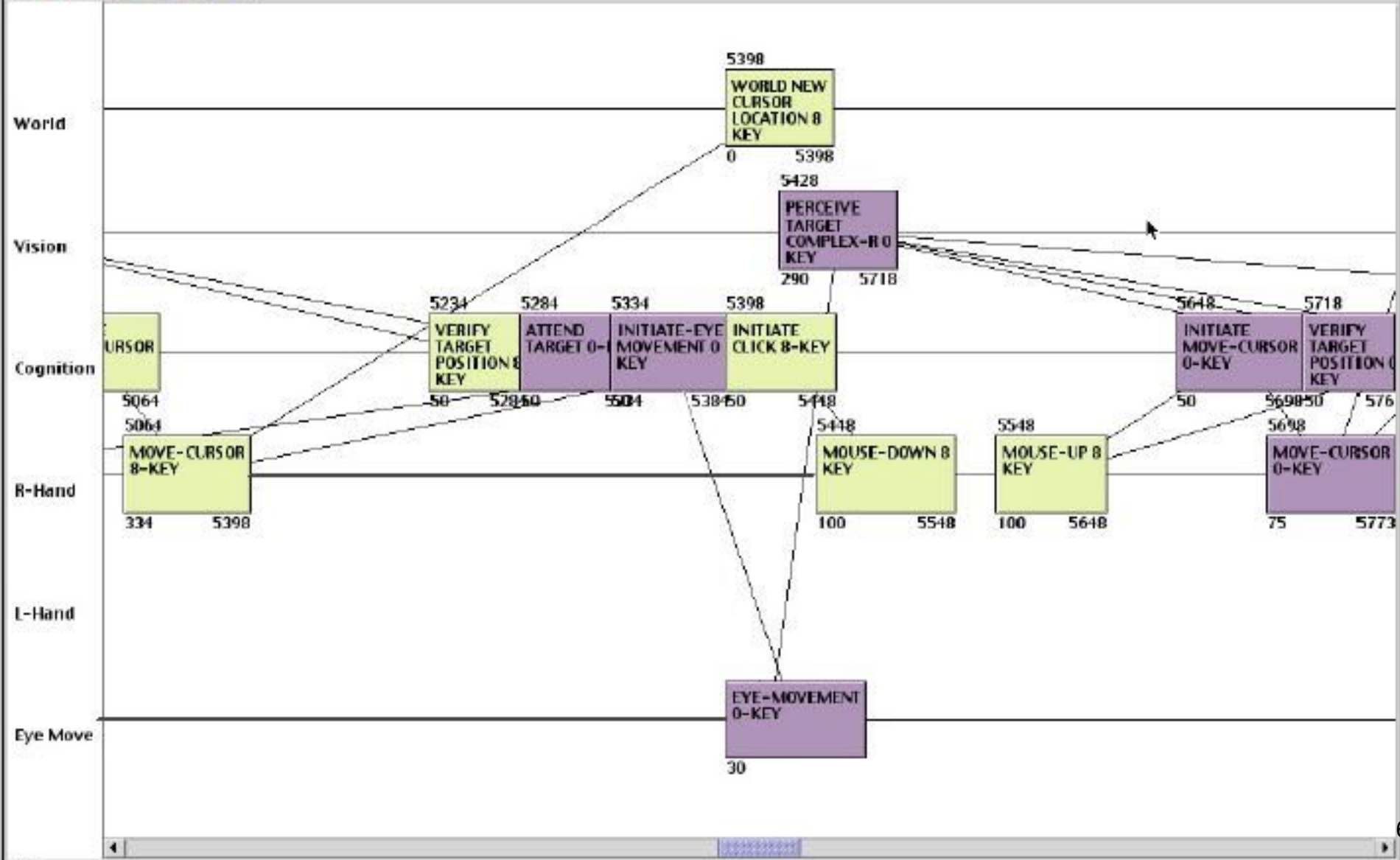
APEX : tool that automates the creation of diagrams



File APEX Options Help

Connect Sim World: ATM-CPM-DURS

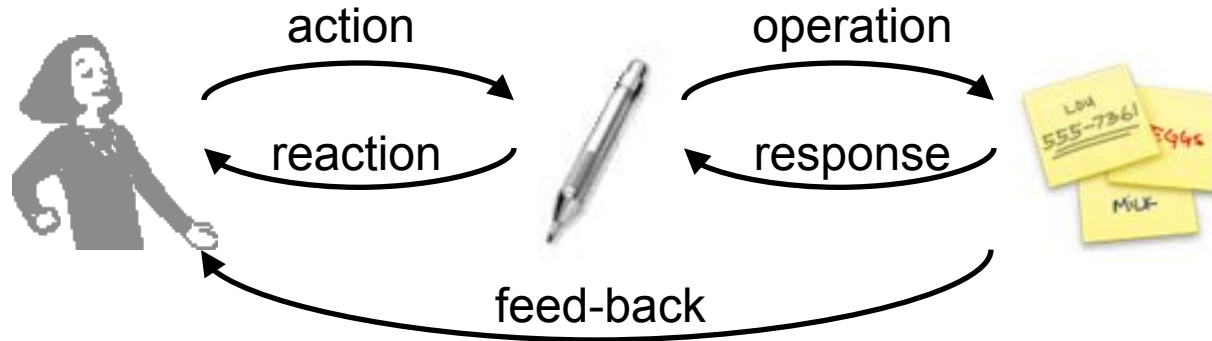
SimView Part ATM-CPM-DURS [10678]



Instrumental Interaction

Interaction model

- Describes an interface in terms of *domain objects* and *instruments*



Descriptive aspect

- Covers a large set of existing techniques (GUI, tangible, AR, ...)

Predictive aspect

- Properties for comparing instruments
 - Degree of indirection, degree of integration, degree of compatibility

Generative aspect

- Design principles: reification, polymorphism, reuse

Software architecture models

Seeheim

MVC - Model-View-Controller

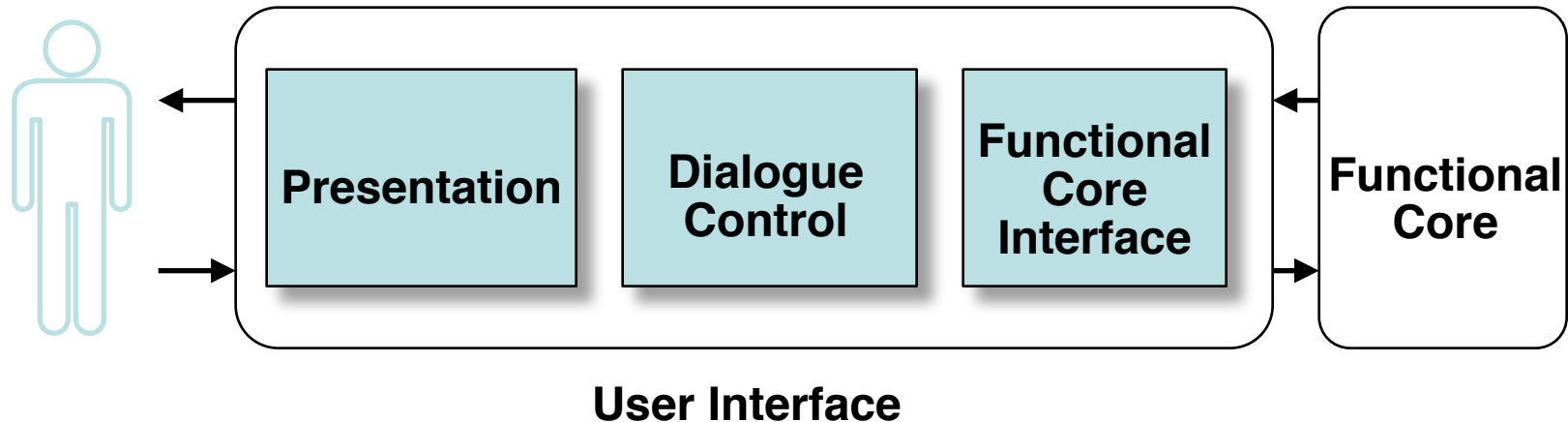
Arch

PAC - Presentation-Abstraction-Contrôle [Coutaz]

MVVM

State machines [Newman]

Seeheim



Presentation

- Manages input and display at a low level

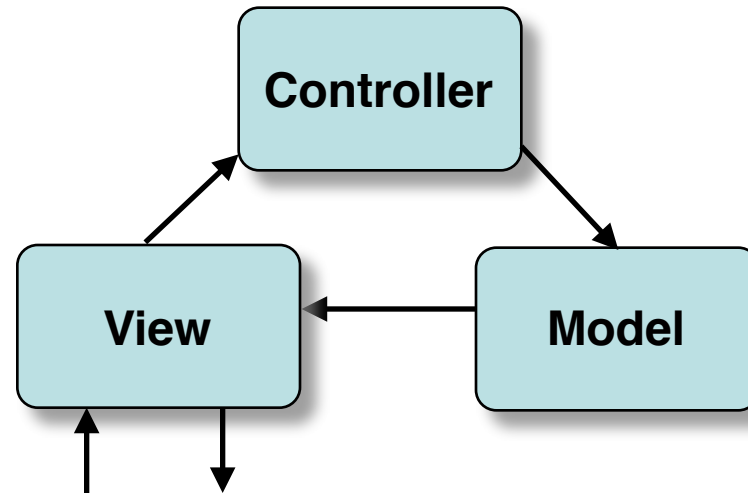
Dialogue control

- Validates input and transforms it into commands
- Transforms responses from the Functional Core into graphical entities

Functional core interface

- Adapts the functional core to the needs of the interface

MVC - Model-View-Controller

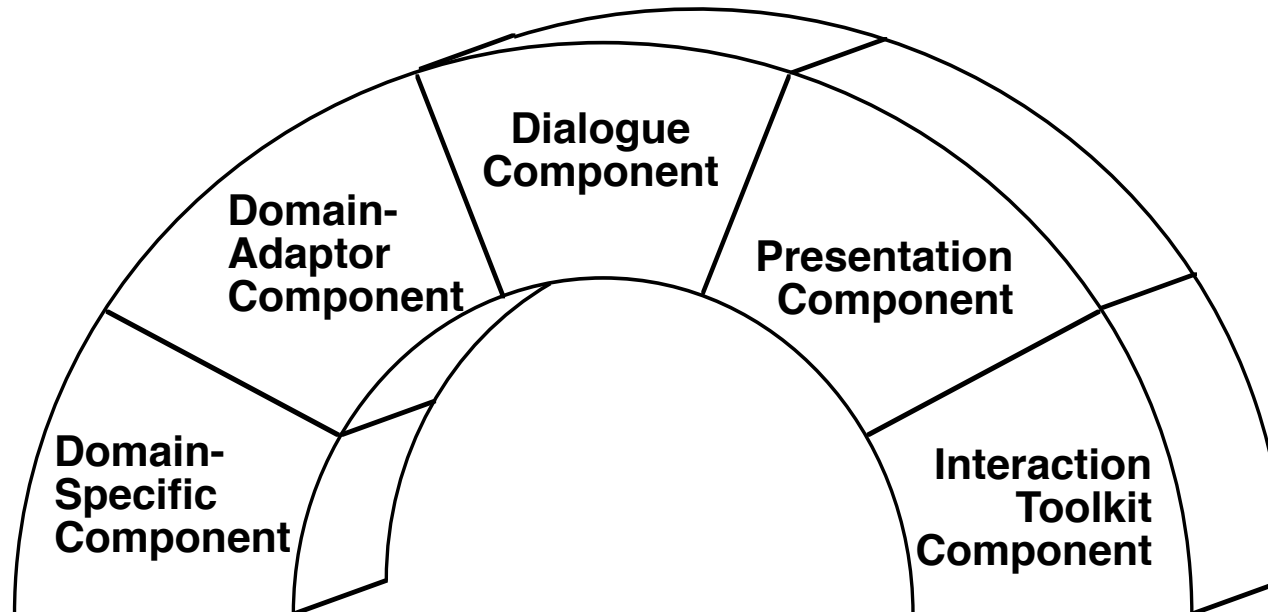


Interface = hierarchical composition of MVC triplets

- Model: abstract representation of the interactive object
- View: graphical representation and input management
- Controller: updates the model when the view is edited

Implemented originally in the Smalltalk system

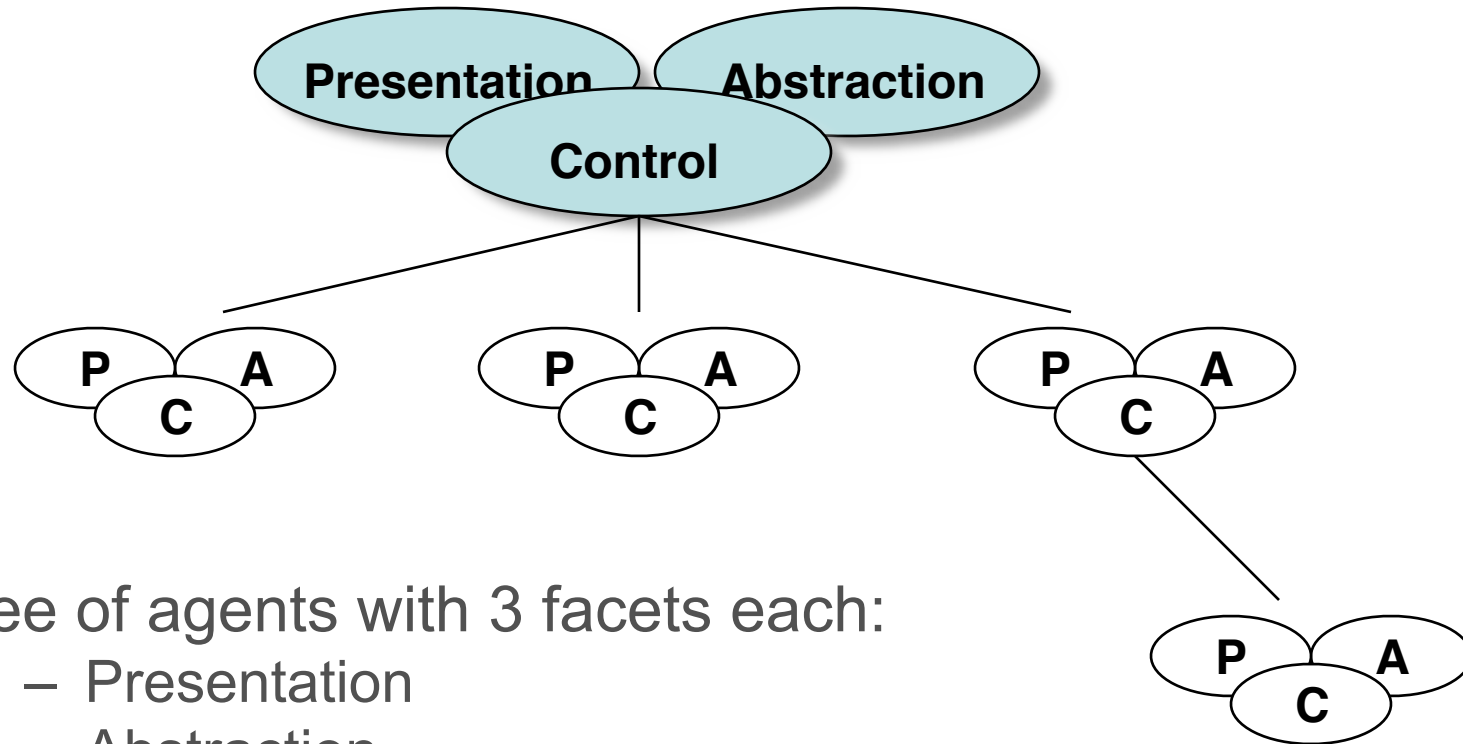
Arch



Modern version of Seeheim

- Acknowledges the existence of user interface toolkits
- Adaptators
 - On the presentation side
 - On the functional core side
- Components can be of different sizes, or even non-existent

PAC - Presentation-Abstraction-Control



Tree of agents with 3 facets each:

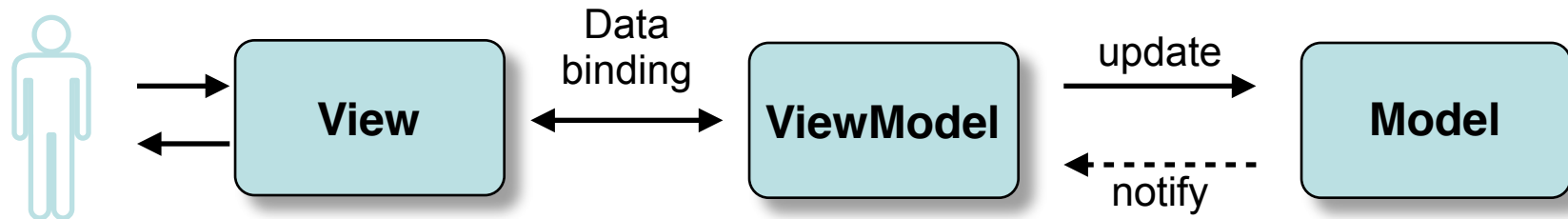
- Presentation
- Abstraction
- Control

Heuristics for the structure of the tree (e.g., multiple views)

Abstract model: no software platform (unlike Smalltalk for MVC)

Numerous evolutions: PAC-Amodeus, PAC*, CoPAC, etc.

MVVM - Model-View-ViewModel



Modern variant of MVC

The View manages the visual representation and user input

The Model holds the application state

The ViewModel abstracts out the state of the View

Data binding automates the synchronization
between View and ViewModel

Software architecture models - applications

User interface toolkits / frameworks

Java Swing is based on MVC

JavaFX is based on MVVM

Web frameworks:

React, Angular, Vue are based on MVVM

The View is provided by HTML/CSS

The Model is provided by the application

The ViewModel is provided by the framework
and uses data binding to automate updates

State machines

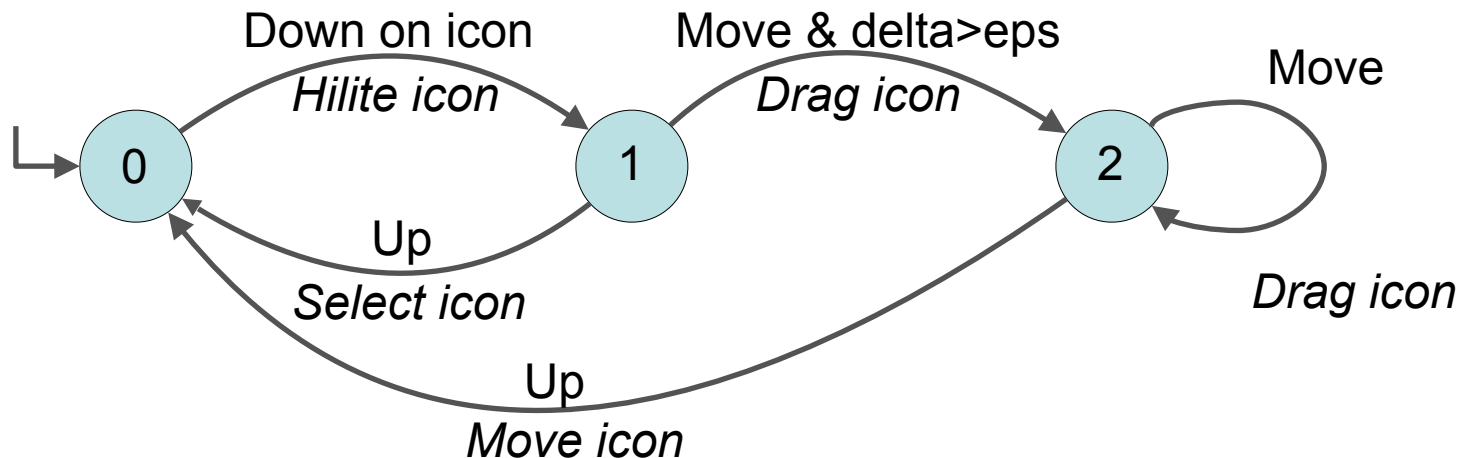
Formal description of the behavior of the interface

Extend finite state automata or transition networks:

- ATN (augmented transition networks)
- RTN (recursive transition networks)
- Statecharts (Harel)
- Petri nets

Proof and validation of properties is possible

Direct link to implementation



State machines - application

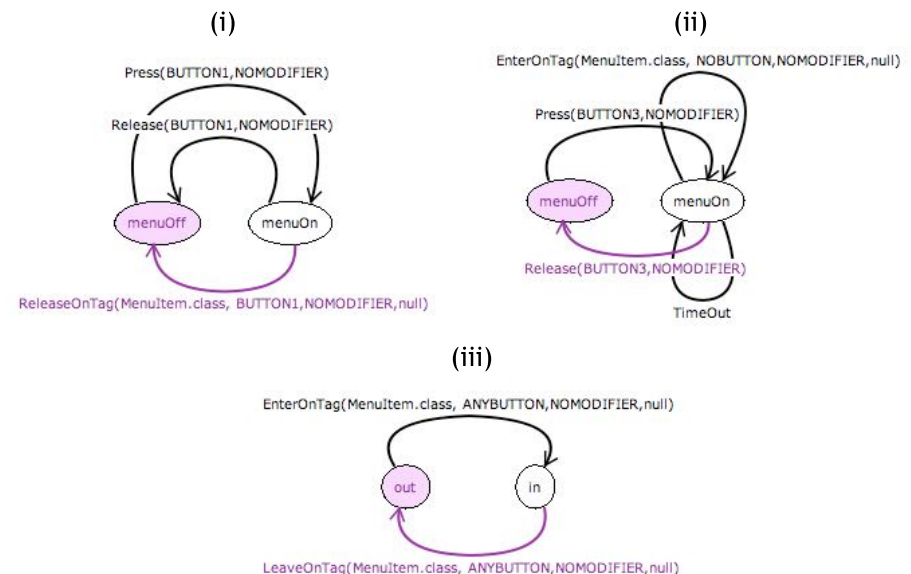
SwingStates library:

Describe interaction with state machines
rather than event listeners / callbacks

```

1  StateMachine sm = new StateMachine() {
2    SMSShape dragged = null;
3    public State start = new State() {
4      Transition dragOn =
5        new PressOnShape(BUTTON1, "drag") {
6          public void action() {
7            dragged = getShape();
8          }
9        };
10   };
11   public State drag = new State() {
12     Transition drag = new Drag(BUTTON1, "drag") {
13       public void action() {
14         move(dragged);
15       }
16     };
17     Transition dragOff =
18       new Release(BUTTON1, "start") { } ;
19   };
20 };

```



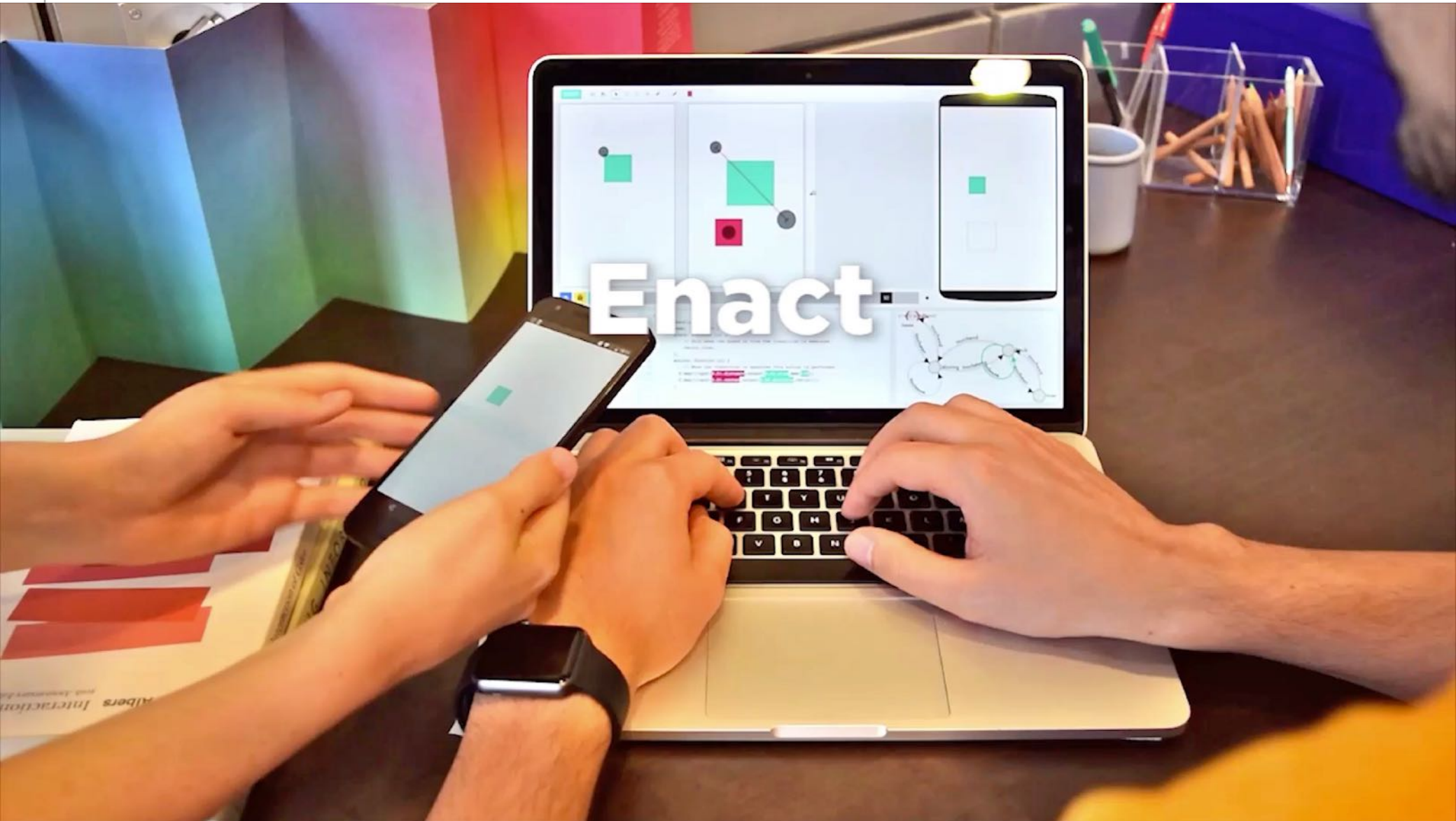
State machines - application

ENACT: support designer-developer collaboration
Create touch-based interaction interactively

The screenshot displays the ENACT software interface, which is used for creating touch-based interactions. The interface is divided into several panels:

- Top Panel:** A toolbar with various icons for editing and navigation.
- Design Canvas:** A large central area showing a design with three horizontal blue bands and a yellow band in the middle. A vertical red line with a circular handle at the bottom is positioned over the yellow band. This design is shown in three different states: (a) the initial state, (b) the state after a touch event (indicated by a plus sign), and (c) the state after a pinch gesture. A smartphone mockup on the right shows the design as it would appear on a mobile device.
- Bottom Panel:** A code editor and a state machine diagram.
 - Code Editor:** Contains JavaScript code for defining actions. The code includes comments and function calls for mapping input events to output actions. A red box highlights a specific part of the code, and a label (d) points to it.
 - State Machine Diagram:** A diagram showing three states: Idle, Moving, and Pinch. Transitions between states are labeled with touch events: touchstart, touchmove, and touchend. A red circle highlights the Pinch state, and a label (c) points to it.

ENACT



Conclusion

Models and theories in human-computer interaction

- Borrowed from Psychology
 - Action/Perception, Cognition
- Borrowed from Sociology
 - Ethnomethodology
- Borrowed from Computer Science
 - Automata
- Specific to HCI
 - GOMS, Instrumental Interaction

Models and theories in HCI are more often descriptive than predictive, and they are rarely generative

Bibliography

HCI Models, Theories, and Frameworks
John. M. Carroll, ed.
Morgan Kaufmann, 2003