Theories and Models for Human-Computer Interaction

Michel Beaudouin-Lafon - mbl@lri.fr
Laboratoire de Recherche en Informatique
In Situ - http://insitu.lri.fr
Outline

What is a theory? a model?

Perception, action

Cognition, behavior

Interaction

Software architectures
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
- 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
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...
  - 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
Principles of Gestalt perception

Common area

Proximity

Connectivity
Ecological Theory of Perception

Fundamental hypotheses:
- Co-evolution between organism and its environment
- Behavioral pre-adaptation
- “Elegant” (and parcimonious) perceptual processes

Ecological optics
- Information is in the optical array and the optical flow
- The organism is equipped to extract invariants
  Example: when moving, the only fixed point indicates the direction of motion

Relativity of the environment
- Action-perception coupling
- “Affordances”
Hick’s law, Fitts’ law

Empirical laws extracted from controlled observations

Hick’s law: time it takes to select an item in a set
- $RT = a + b \log_2(n)$
  - $a$ & $b$ are constants, $n$ is the number of items

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

This laws are valid only in precise experimental settings
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:
- 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

Falsification:
- Some tasks are more efficient when the hands have symmetric roles
Cognition and behavior

Action theory [Norman]

Situated action [Suchman]

Activity theory [Vigotsky, Bødker]

Cognitive dimensions [Green]
Action theory

- Goal
- Intention
- Specification of actions
- Execution
- Perception
- Interpretation
- Evaluation
- Output articulatory distance
- Output cognitive distance
- Execution distance
- Articulatory input distance
- Articulatory cognitive distance

System
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
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)
- Actions: executed consciously to reach an explicit goal set by the subject
- Operations: executed unconsciously or semi-consciously to execute actions
Activity theory

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

Morphological analysis of input devices [Card et al.]

UAN [Hartson]

State machines [Newman]

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
- \( \text{In} \) = Input domain
- \( S \) = Current state of the device
- \( R \) = Resolution function: In -> Out
- \( \text{Out} \) = Output domain
- \( W \) = Other properties of interest

Composition of input devices:
- Merge
- Layout
- Connect

Card, Mackinlay & Robertson
Example

Radio:
- Volume dial
- AM/FM selector
- Frequency selector
Taxonomy

Comparison of input devices, including those studied by Foley and by Buxton
UAN : User Action Notation

Description of user actions and system responses
Example : selecting an icon

<table>
<thead>
<tr>
<th>Action</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>~[icon] Mv^</td>
<td>icon!</td>
</tr>
</tbody>
</table>

More accurate version:

<table>
<thead>
<tr>
<th>Action</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>~[icon] Mv</td>
<td>icon-! : icon! , all icon’! : icon’-!</td>
</tr>
<tr>
<td>M^</td>
<td></td>
</tr>
</tbody>
</table>

Moving an icon:

<table>
<thead>
<tr>
<th>Action</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>~[file_icon] Mv</td>
<td>file_icon-! : file_icon! , all icon’! : icon’-!</td>
</tr>
<tr>
<td>~[x,y]^ ~[x’,y’]</td>
<td>outline(file_icon) &gt; ~</td>
</tr>
<tr>
<td>M^</td>
<td>@x’,y’ display(file_icon)</td>
</tr>
</tbody>
</table>
## UAN

<table>
<thead>
<tr>
<th>Action</th>
<th>Feedback</th>
<th>Interface state</th>
<th>Computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>~[file_icon] Mv</td>
<td>file_icon-! : file_icon! , all icon'! : icon’-!</td>
<td>selected = file</td>
<td></td>
</tr>
<tr>
<td>~[x,y]* ~[x’,y’]</td>
<td>outline(file_icon) &gt; ~</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M^</td>
<td>@x’,y’ display(file_icon)</td>
<td>pos(file_icon) = x’,y’</td>
<td></td>
</tr>
</tbody>
</table>

**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)
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
The GOMS family of models

GOMS = Goals, Operators, Methods, Selection rules

- **Goals**: what the user wants to do
- **Operators**: actions supported by the software application
- **Methods**: learned sequences of subgoals and operators to reach a goal
- **Selection 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

GOMS models are 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

Six 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 (1.00s)
- 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
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
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]
Seeheim

User Interface

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
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-existant
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.
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