Human-in-the-loop? or Human-Computer Partnerships?

Wendy E. Mackay

Inria, Université Paris-Saclay
22 April 2018
Unified principles of interaction

Two complementary perspectives:
System: How to build it?
Instrumental Interaction and Substrates

with Michel Beaudouin-Lafon
Unified principles of interaction

Two complementary perspectives:

System: How to build it?
   Instrumental Interaction and Substrates

User: How to interact with it?
   Co-adaptive Systems
   Human-computer partnerships
Our relationship to a computer

Computer as **tool**
Empower users

Computer as **servant**
Delegate tasks

Computer as **medium**
Communicate

Human-Computer Interaction

Artificial Intelligence

Mediated Communication
Different perspectives:

Machine learning:

*Human-in-the-loop*

Use human input to *improve the algorithm*
Different perspectives:

Machine learning:  
*Human-in-the-loop*  
Use human input to *improve the algorithm*

User types  
**Google** suggests  
User chooses
Too often, this is the ‘human-in-the-loop’!
Different perspective

If *Human-in-the-loop* uses human input to *improve the algorithm*
How do we interact with computers?

If Human-in-the-loop uses human input to *improve the algorithm*, shouldn’t we also have ‘Computer-in-the-loop’ to *empower the user*?
Human-Computer Partnerships

Instead of just creating models of users to inform the system, we should create models of the system to inform the user?

Together, they create effective human-computer partnerships.
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
Multiple ways to interact

Discover

Appropriate

Express
Human-Computer Partnerships

To share control, users need:

- Discoverability
- Appropriability
- Expressivity
Why can’t we learn to ‘play’ software tools without relearning the interface after every software upgrade?
Take a smartphone ...
Smartphone interfaces are simple

- Tap
- Drag
- Swipe
- Pinch
Smartphone interfaces are simple

Why not powerful, expressive and simple?
Major design trade-off

How to balance:
- power of expression
- simplicity of execution?
Solution: Shift the curve

Simple things should be simple, complex things should be possible
Human-computer partnerships

People can

adapt to technology

adapt the technology

they learn it

they appropriate it
Human-computer partnerships

People can

adapt to technology they learn it
adapt the technology they appropriate it

Computers can

adapt to people they learn (AI)
adapt people’s behavior they teach (CAI)

Reciprocal co-adaptation
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
Octopocus

Progressive feedforward

What gestures are available?

Progressive feedback

What did the system recognize?

Bau & Mackay, UIST'09
Inking the 'Help' command
How can I create my own gesture commands?
Fieldward

Create your own gesture commands
Must be:
   easy for you to remember
Create your own gesture commands
Must be:
easy for you to remember
easy for the system to recognize
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!
Fieldward:

Fieldward

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

How can I generate expressive text?
Human expression

Handwriting is expressive

I will be thankful for what I have
1. I will be thankful for what I have
2. I will be thankful for what I have
3. I will be thankful for what I have
4. I will be thankful for what I have
5. I will be thankful for what I have
6. I will be thankful for what I have
7. I will be thankful for what I have
8. I will be thankful for what I have
9. I will be thankful for what I have
10. I will be thankful for what I have
11. I will be thankful for what I have
12. I will be thankful for what I have
13. I will be thankful for what I have
14. I will be thankful for what I have
15. I will be thankful for what I have
16. I will be thankful for what I have
17. I will be thankful for what I have
18. I will be thankful for what I have
Handwriting is expressive ... SMS messages -- not so much
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 . . .
Human expression vs. Machine classification

Four ways to input the word “great”

All produce the identical result: great
Expressive Keyboard vs. Machine classification

Machine learning
Guess the correct word (classify)
Throw away human variation

Human-centered approach
Create expressive output
Transform human variation

Alvina, Malloch & Mackay CHI’16
Expressive Keyboard

Expressive Keyboards produce accurate words, but let users control the output properties.
Expressive Keyboard

Map gesture variation to output properties

Users control:
- text color
- font style
- emojis
Expressive Keyboard

Machine learning guesses the correct word ...

Expressive Keyboards produce accurate words, but let users control the output properties.
Expressive Keyboard – measure variation

Can users control their gestures deliberately to produce rich output, such as colored text?
Expressive Keyboard – Expressive emojis

What else can we do with Expressive Keyboards?
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
What about socio-technical principles?

Social scientists conduct extensive field studies and provide deep insights in the form of socio-technical principles about how people interact with technology in context.

But it is difficult to translate these principles into specific designs.
Generative Deconstruction

Apply socio-technical principles systematically to generate grounded designs

Socio-technical principles

**OBSERVE**
use-technology-context
Specific anecdotes, breakdowns, surprises

**DECONSTRUCT**
design problem
from: abstractions
to: patterns

**RECONSTRUCT**
design solution
Revise design space
Explore design options
Examples of Socio-technical Principles

<table>
<thead>
<tr>
<th>Principle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Situated Action</td>
<td>Go beyond planned activities; Users decide how to act in unforeseen circumstances</td>
</tr>
<tr>
<td>Rhythms &amp; routines</td>
<td>Build upon routine activities and spatial patterns; Users integrate systems into their daily lives</td>
</tr>
<tr>
<td>Peripheral awareness</td>
<td>Design for both focus and periphery; Users vary degree of engagement</td>
</tr>
<tr>
<td>Co-adaptation</td>
<td>Expect users to re-interpret and customize; Enable capture and sharing of customizations</td>
</tr>
<tr>
<td>Distributed cognition</td>
<td>Let objects and people reduce cognitive load for memory or communication tasks</td>
</tr>
</tbody>
</table>
So …

How do we incorporate socio-technical principles into the design process?
Generative Deconstruction & Reconstruction

Observe users either:
- to understand what to design or
- to evaluate what has been designed

First **deconstruct** what is going on:
- Who is the user?
- What is the technology?
- What is the user’s context?
- What is the interaction like?

Then **reconstruct** the design
- to design a new technology or
- to fix an existing one
Generative Walkthroughs: Creative redesign

Structured walkthroughs
Systematic critique of design artifacts, such as scenarios & storyboards
Generative Walkthroughs: Creative redesign

Structured walkthroughs
Systematic critique of design artifacts, such as scenarios & storyboards

plus

Focused brainstorming
Generation of novel ideas, based on socio-technical principles
Exercise: Generative Walkthroughs

Step-by-step: deconstruct and reconstruct a scenario in light of socio-technical principles

- scenario or storyboard
- situated action
- rhythms & routines
- peripheral awareness
- co-adaptive systems
- distributed cognition
How do we
- deconstruct today’s interactive systems
- reconstruct to generate new forms of interaction?
Discoverability

How can I learn new gestures that execute commands?
Gesture typing: Typing with gestures

Instead of tapping...

draw through each letter to type a word

Zhai, S. & Kristensson, P.-O. CHI’03
Transform gestures into commands ...

draw: 

\textit{alice}

Alvina, Griggio, Bi & Mackay UIST'17
CommandBoard

Transform gestures into commands ...

draw:
alice
choose contact: Alice Brooke

Alvina, Griggio, Bi & Mackay UIST’17
CommandBoard

Transform gestures into commands ...

write: alice
choose contact: Alice Brooke
choose comm app: Alice Brooke

Alvina, Griggio, Bi & Mackay UIST’17
CommandBoard

Command-line interaction from a soft keyboard

CommandBoard offers a simple, efficient and incrementally learnable technique for issuing gesture commands from a soft keyboard.
CommandBoard

Execute any command from a gesture keyboard

<table>
<thead>
<tr>
<th>Screen Space</th>
<th>Contact</th>
<th>User Action</th>
<th>User Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>command-gesture input space</td>
<td>WhatsApp</td>
<td>draw gesture</td>
<td>execute command, add gesture shortcut</td>
</tr>
<tr>
<td>command bar</td>
<td>Alice Brooke</td>
<td>cross command option</td>
<td>choose command option</td>
</tr>
<tr>
<td>suggestion bar</td>
<td>Alice Waltz</td>
<td>tap word</td>
<td>choose word or command</td>
</tr>
<tr>
<td>text input space</td>
<td>slice</td>
<td>tap key, cross key, dwell on key</td>
<td>enter text/emoticon, change layout, specify command</td>
</tr>
</tbody>
</table>

Alvina, Griggio, Bi & Mackay UIST’17
CommandBoard

Execute any command from a gesture keyboard

Alvina, Griggio, Bi & Mackay UIST’17
CommandBoard

Draw *color*, then pick from a color wheel

Draw table, then insert a table