# 4M21 Software Engineering and Design
# Human-Computer Interaction

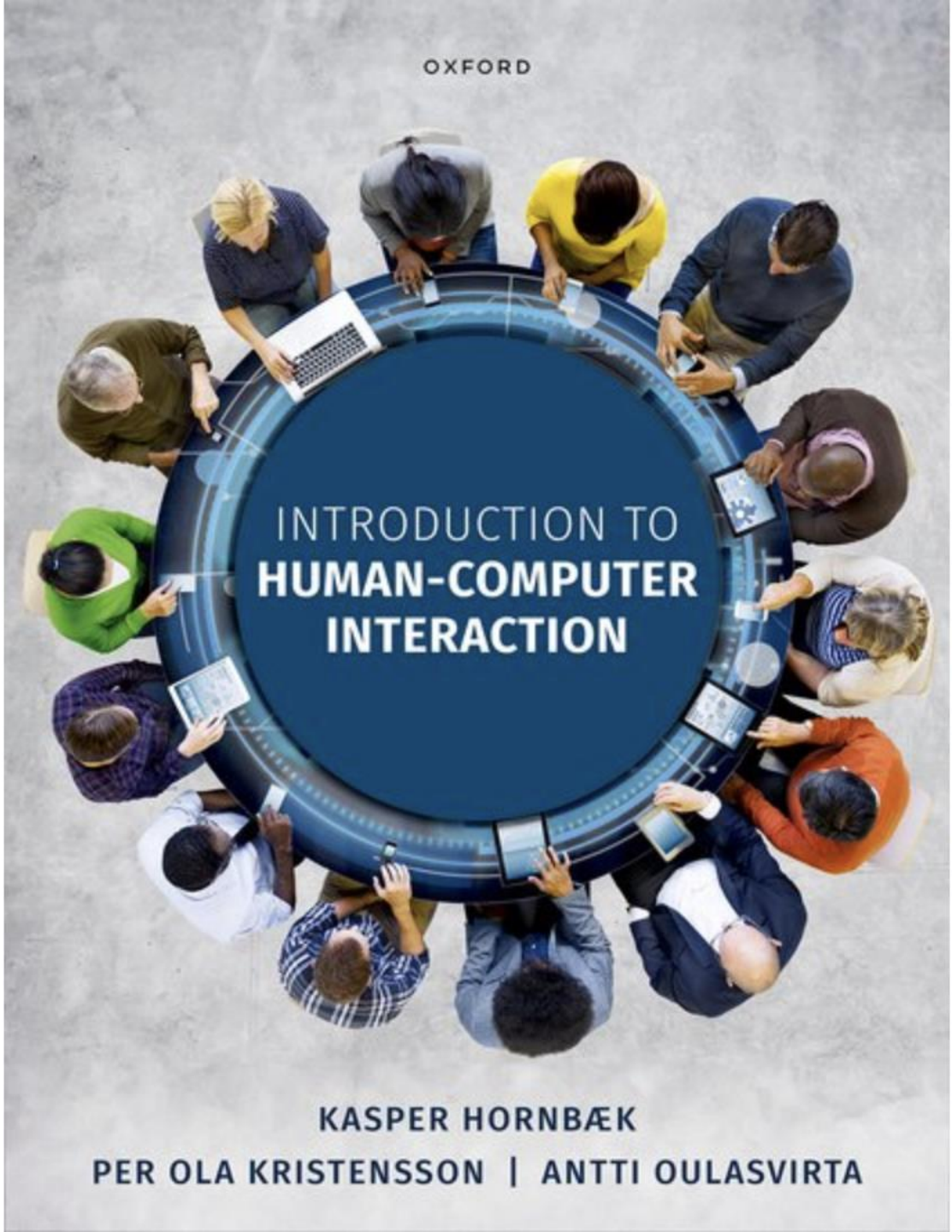
## Lecture 4/8

Professor Per Ola Kristensson

Department of Engineering

University of Cambridge

https://global.oup.com/academic/product/introduction-to-human-computer-interaction-9780192864543?cc=gb&lang=en&



# Interaction

# What makes a good theory of interaction?

- Theories should **explain**
  - The **explanatory power** of a theory refers to the empirical accuracy and coverage of the explanations that a theory offers
  - The more accurate and the broader the coverage, the higher the explanatory power
- Theories should **predict**
  - Theories link predictions to factors that we can measure or affect, such as characteristics of the task, the user interface, or users
  - Theories are instantiated through **models** that allow us to make predictions
- Theories should help us **evaluate**
- Theories should guide **measurements**
- Theories should **inform design**
  - *Abduction*: explaining observations through theory
  - *Deduction*: theory suggest ways to design
  - *Counter-factual reasoning*: if the design was *X* then the interaction would be *Y*

# Information and Control

# Information, entropy, and perplexity

- **Self-information** of a message *m* in bits:

$$I(m) = \log_2\left(\frac{1}{P(m)}\right) = -\log_2\left(P(m)\right)$$

- The **binary entropy function** of a first message with probability *p* and a second message with probability $q = 1 - p$ is:

$$H_2(p) = -p\log_2(p) - q\log_2(q)$$

  - Note: binary entropy is maximized when $p = q = 0.5$

- The average self-information in a message space $\mathbb{M}$ is known as **entropy**:

$$H(\mathbb{M}) = \sum_{m\in\mathbb{M}} P(m)I(m) = -\sum_{m\in\mathbb{M}} P(m)\log_2(P(m))$$

  - Entropy is the average bits required to communicate a message when using an optimal coding scheme
  - **Redundancy** is the difference between the average actual bits used for communication compared to the average bits required for communication using an optimal coding scheme

- **Perplexity** is the weighted average number of choices a random variable has to make:

$$PP = 2^H$$

# Mutual information

- The **mutual information** of two random variables $\mathbb{A}$ and $\mathbb{B}$ is:

$$I(\mathbb{A};\mathbb{B}) = \sum_{a\in\mathbb{A}}\sum_{b\in\mathbb{B}} P(a,b)\log_2\left(\frac{P(a,b)}{P(a)P(b)}\right)$$

- Mutual information expresses the number of bits of information about a random variable that can be obtained by observing another random variable
- Intuitively, it captures how much knowing one random variable reduces the uncertainty of another random variable
- In the extreme case when both random variables are independent the mutual information is zero

# Rate

- Assume a random variable $\mathbb{I}$ is a distribution over the set of words a user is intending to write and the random variable $\mathbb{O}$ is a distribution over the set of words the user is writing
- Using the concept of mutual information, we can then arrive at a **rate** *R* (in bits/s):

$$R = \frac{I(\mathbb{I};\mathbb{O})}{t}$$

  - where $I(\mathbb{I};\mathbb{O})$ is the mutual information of $\mathbb{I}$ and $\mathbb{O}$, and $t$ is the average time duration in seconds
- Since $I(\mathbb{I};\mathbb{O}) = H(\mathbb{I}) - H(\mathbb{I}|\mathbb{O})$, where $H(\mathbb{I}|\mathbb{O})$ is the **conditional entropy**, the rate can be rewritten as:

$$R = \frac{H(\mathbb{I}) - H(\mathbb{I}|\mathbb{O})}{t}$$

- If the probability of error is zero, the term for the conditional entropy $H(\mathbb{I}|\mathbb{O})$ will vanish and the rate *R* can be expressed as:

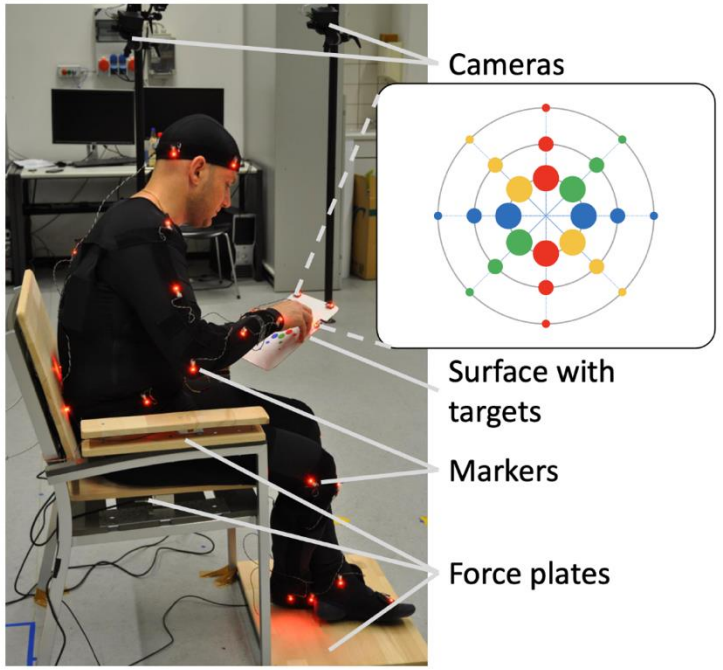
$$R = \frac{H(\mathbb{I})}{t}$$

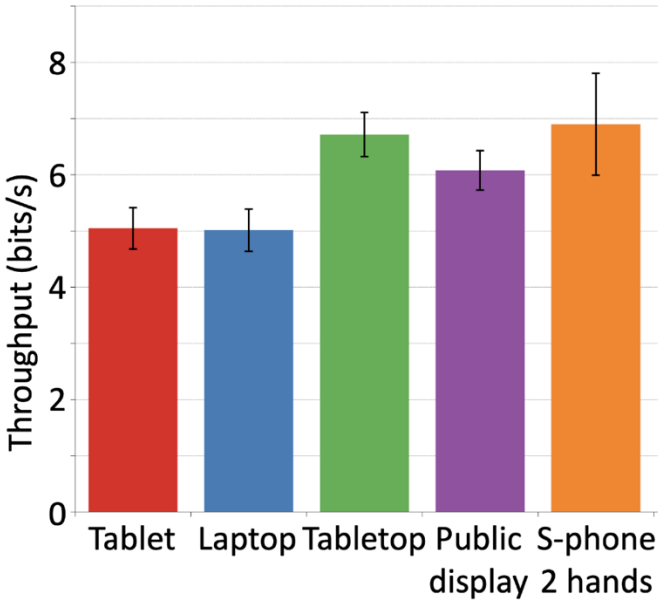
# Throughput

- An abstraction is needed in order to measure performance intrinsic to an input device
  - Instead of comparing raw performance scores, we should look at how many bits per second a user is able to express
- This requires the following steps:
  - Users are asked to point as quickly and as accurately as possible in varying *ID* conditions
  - Movement times and accuracies are recorded and averaged per *ID* condition
  - The Fitts' law equation $MT = a + bID$ (linear regression) is fit on the data
  - Throughput (*TP*) is computed
- Throughput definition 1: $TP = \frac{ID_{avg}}{MT_{avg}}$, where $ID_{avg}$ is the average *index of difficulty* and $MT_{avg}$ is the average movement time
  - Downside: Reliant on an average index of difficulty, which is arbitrary
- Throughput definition 2: $TP = \frac{1}{b}$, where $b$ is the slope in Fitts' law
  - Downside: Ignores any effect of the intercept parameter $a$ in Fitts' law

# Example: measuring throughput

- Investigating the throughput of devices using motion capture and biomechanical simulations allows explanations of different throughputs
- Muscle groups and movement ranges matter
- Using larger muscles means lower performance
- Surprisingly, when given the chance to decide their own posture, laptop users mainly used shoulder and arm muscles for pointing
- While their posture was suitable for long-term use, performance in pointing was poor

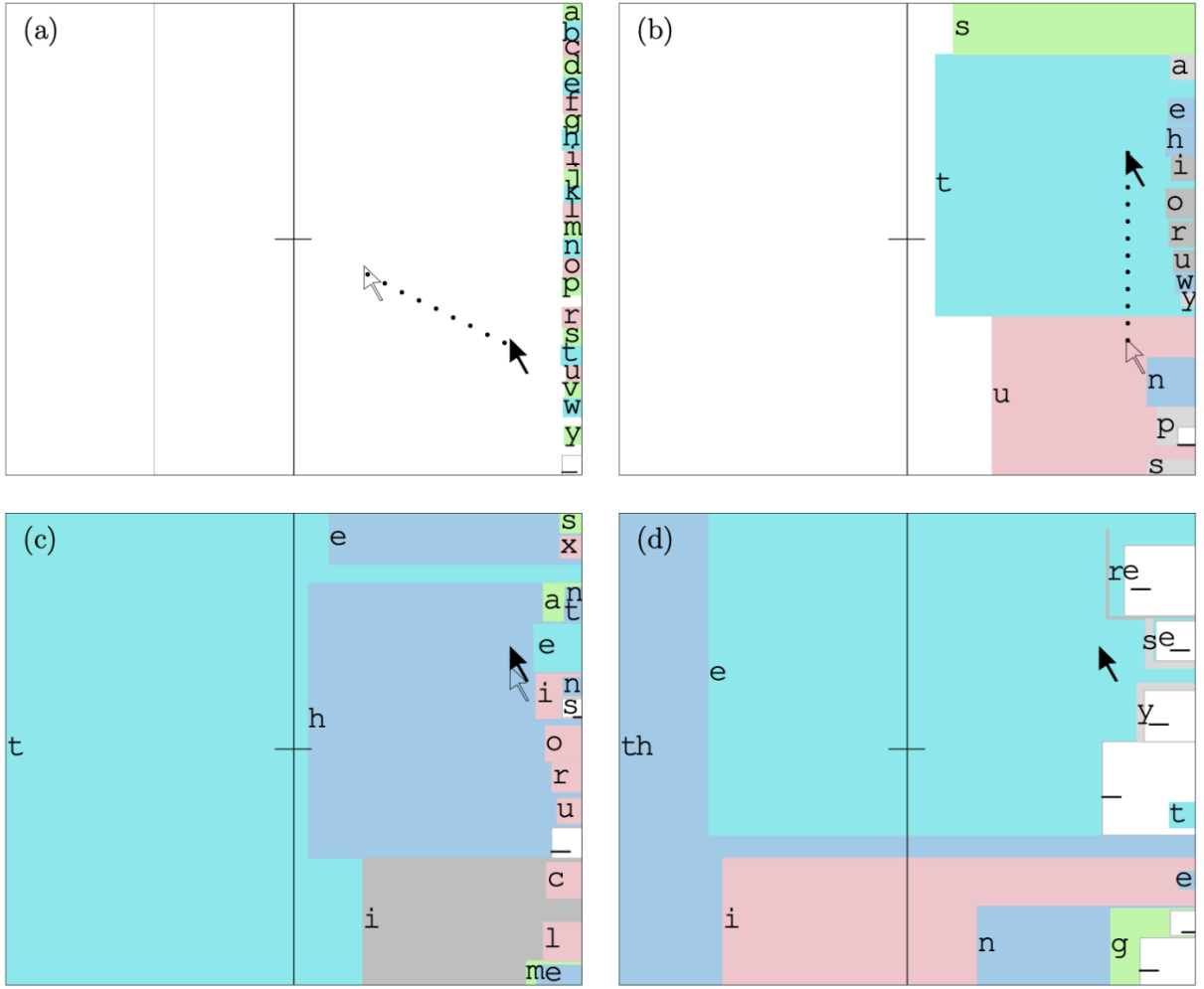



M. Bachynskyi, G. Palmas, A. Oulasvirta, J. Steimle, and T. Weinkauf. Performance and ergonomics of touch surfaces: a comparative study using biomechanical simulation. In Proceedings of the Conference on Human Factors in Computing Systems, pages 1817–1826, 2015.

# Example: Dasher

- *Dasher* is motivated by information theory: our keyboards are highly inefficient
  - To enter text using a regular keyboard, a user needs to  type character-by-character one key from a set of 80 keys (Qwerty)
  - Every key press has an information capacity of $\log_2(80) = 6.3$ bits
  - However, the English language has entropy of about 1 bit per character
  - This means that the keyboard is inefficient by a factor of six
- In Dasher, the user navigates to the desired letter or word, which causes the display to zoom into that area
- Dasher is information-efficient but relies on closed-loop visual feedback, which is slow



# Example: analyzing Dasher

- A user writes using Dasher by continuously navigating to a desired sequence of letters in a graphical user interface laid out according to a language model
  - If the user writes at a rate $R_D$ (in bits) then Dasher attempts to zoom in to the region containing the user's intended text by factor of $2^{R_D}$
  - If the language model generates text at an exchange rate of $R_{LM}$ then the user will be able to reach an entry rate of $R_D/R_{LM}$ characters per second
- Dasher can also be controlled by discrete button presses in which timing is disregarded
  - Assuming the button pressing is precise and therefore without errors, the capacity $C$ of the communication channel between the user and the computer is:

$$C = \frac{\log_2(n)}{t}$$

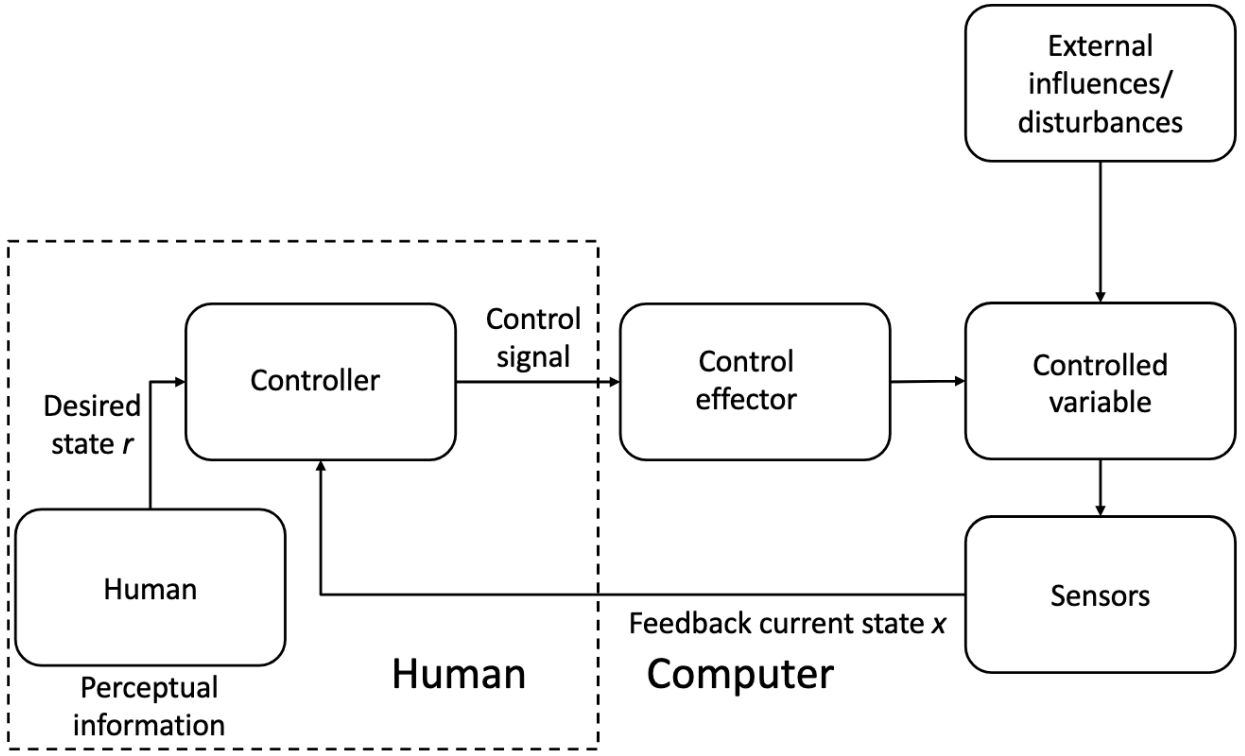
  - where $n$ is the number of buttons and $t$ is the average time to switch button
- If the process is noisy, such that the wrong button is pressed a fraction $f$ of the time, then the capacity is:

$$C = \frac{\log_2(n)}{t} H_2(f)$$

  - where $H_2$ is the binary entropy function
  - Even at a rather low error rate of $f = 1/10$ , the channel capacity $C$ is scaled by approximately 0.47 and thus reduced by approximately a factor of two

# Closed-loop human computer systems

- In addition to viewing human-computer interaction is transmitting bits across a noisy channel, we can also view human-computer interaction as a control system
- The figure shows a model of a model of a close-loop human-computer system



# Example: control as target acquisition 1/ 2

- Identify the reference variable $r$
  - The reference variable $r$ corresponds to the distance (amplitude) $A$ of the target in Fitts' law
- Now assume a change from the home position to the target position is a step change in the reference variable $r$
- Set the controller to a first-order controller with a gain $k$ and integrator $\dot{x} = Bu$, where $u$ is the control signal $u = r - x$, and $B = k$
- A step change in $r$ from the initial state $x = 0$ results in a response of the first-order lag, which is exponential:

$$x(t) = r\left(1 - e^{-kt}\right)$$

# Example: control as target acquisition 2/ 2

- Consider a target size $w$, corresponding to a target size $W$ in Fitts' law
- The time it would take to get within $\frac{1}{2}w$ of $r$ for a target size $w$ centered on $r$ is:

$$x(t) = r\left(1 - e^{-kt}\right) = r - \frac{1}{2}w$$

$$\mathrm{e}^{-kt} = \frac{w}{2r}$$

$$-kt = \ln\frac{w}{2r}$$

$$t = \frac{1}{k}\ln\frac{2r}{w}$$

- Changing the base of the logarithm to base 2 we obtain the following expression:

$$x(t) = \frac{\ln 2}{k}\log_2\frac{2r}{w}$$

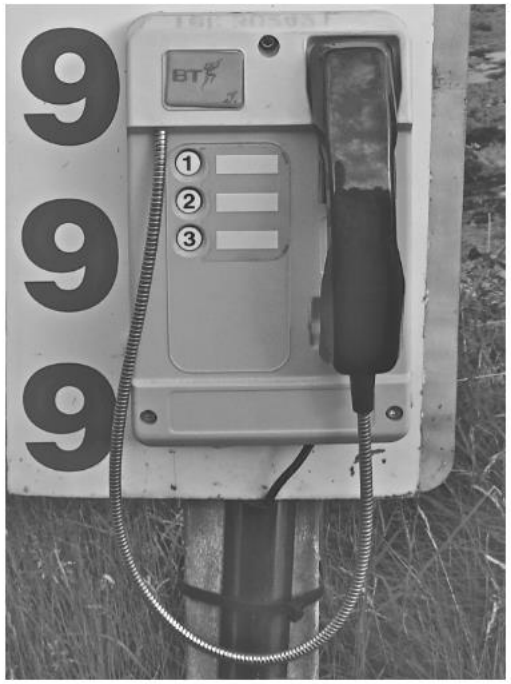
  - which is similar to the formulation of Index of Difficulty in Fitts' law:

$$ID = \log_2\left(\frac{2A}{W}\right)$$

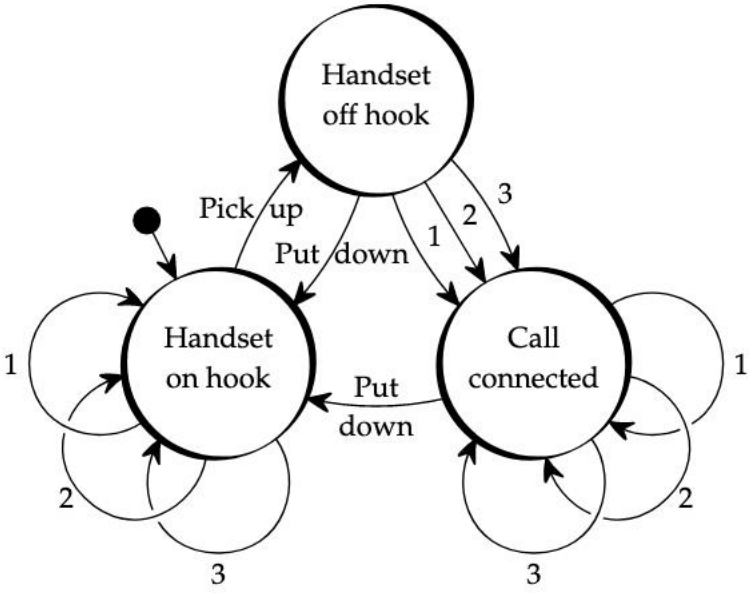
# Dialogue

# Dialogue as state-based interaction

- Dialogue is modeled as transitions or exchanges between two or more partners
- This formalism can be used to algorithmically verify that interaction fulfills certain properties, such as supporting undo or recovery from errors



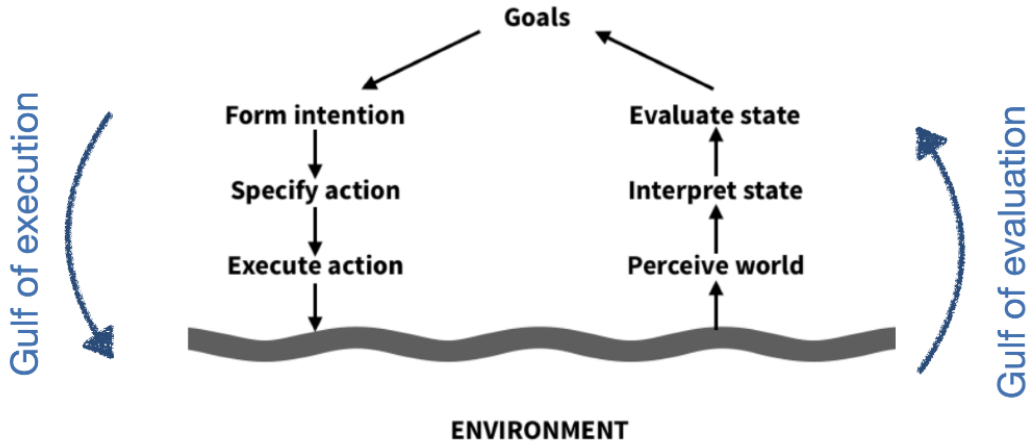

(a) A phone booth




(b) FSM model

# Dialogue as goal-directed action

- The defining cognitive challenge in dialogue according to this view is understanding the communication partner, such that the appropriate next turn can be taken
- Dialogue is intentional: users are attempting to drive the computer to a particular desired state
- However, because of known limitations of human cognition, users cannot do this perfectly. They need to engage in:
    1. Planning: deciding what to do
    2. Inference: interpreting the computer's state
- Gulf-of-execution: knowing what to do to have a desired state change in the computer
- Gulf-of-evaluation: knowing what the current state in the system is by interpreting system output



# Mixed-initiative interaction

- **Mixed initiative interaction** is the idea of organizing interaction in dialogue where both the computer and the human can take initiative by coupling an automated service with direct manipulation
- As an example of such an interface, consider an email interface that automatically analyzes incoming emails
  - Such a system can infer whether an email is about scheduling a meeting and suggest or automatically invoke a calendar to assist the user in achieving this goal
  - Such a system is possible if the system is capable of accurately inferring the user's goal and assessing the costs and benefits in providing an automated action or suggestion
  - Even if the system fails to trigger a calendar, the user can initiate this action by clicking an appropriate icon at any time in the interface
- A mixed initiative interface needs to infer the user's goals so that it can act upon them
- Once the system has inferred the user's beliefs it, needs to decide what action to take
- It is also possible to ask the user about their goal, and whether to trigger such a question can also be estimated by calculating the expected utility gain in performing this action
- When an automated action is taken it is important to consider the timing as incorrect timing of automated actions can appear distracting to the user

# Principles of mixed-initiative interfaces (1/2)

- **Developing significant value-added automation:** only automate if a direct manipulation solution will be inferior
- **Considering uncertainty about a user's goals:** systems should consider the uncertainty of users' actions and other behavior and incorporate such uncertainty into its automation mechanisms
- **Considering the status of a user's attention in the timing of services:** systems should be mindful of users' attention and the timing when invoking automated services
  - Systems should leverage such understanding when deciding on the costs and benefits of deferring actions to more beneficial times
- **Inferring ideal action in light of costs, benefits, and uncertainties:** systems should take into account the expected value of taking automated actions given the costs, benefits and uncertainties associated with context-specific decisions
- **Employing dialog to resolve key uncertainties:** if the system is uncertain about the user's intent the system should ask the user, after having considered the cost of interrupting the user
- **Allowing efficient direct invocation and termination:** since the system will be unlikely to always automate functions successfully it is important users can directly trigger and terminate functions

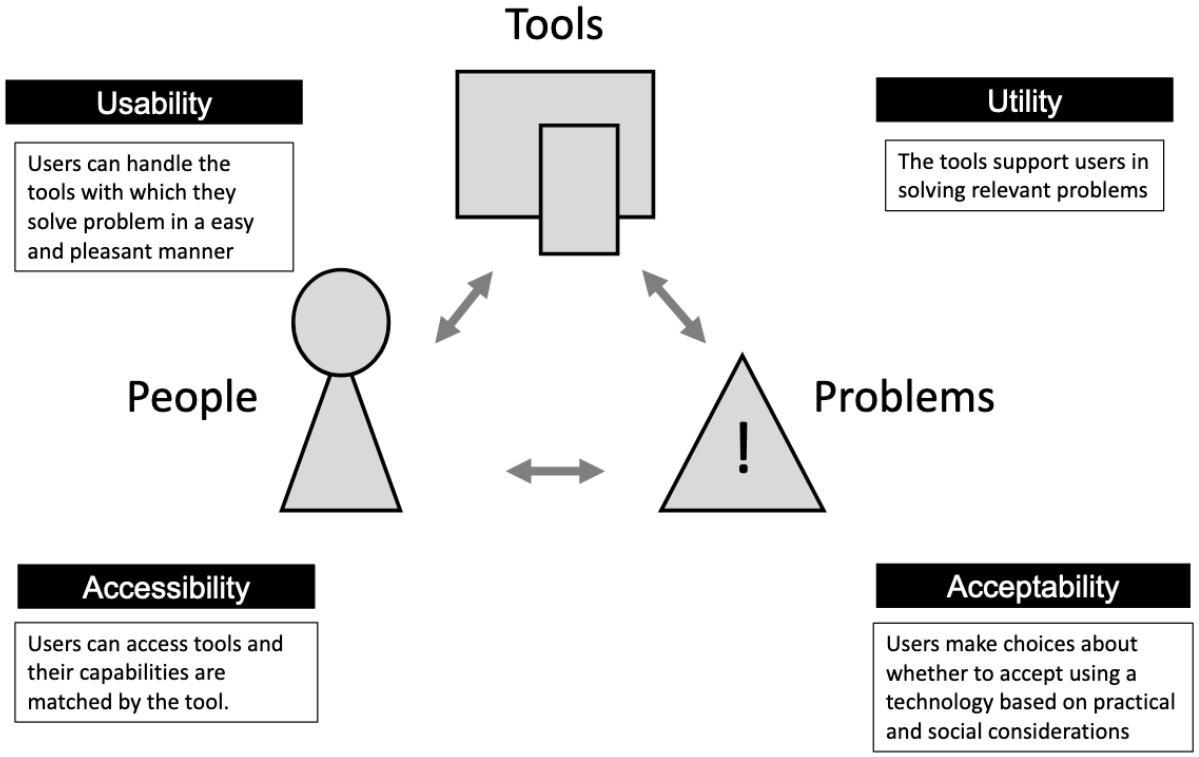
# Principles of mixed-initiative interfaces (2/2)

- **Minimizing the cost of poor guesses about action and timing:** system triggered alerts and suggestions should be designed such that if happen to be inaccurate or poorly timed, the cost of interruption of the user is minimized
  - This can, for example, be achieved by making suggestions less distractive, use natural time outs if there is no user response, and provide swift user actions to dismiss suggestions
- **Scoping precision of service to match uncertainty, variation in goals:** systems may benefit by dynamically adjusting their use such that if a system operates under a high uncertainty of the user's goals the system performs less automation in order to avoid interrupting the user with poor suggestions
- **Providing mechanisms for efficient agent-user collaboration to refine results:** systems should be designed to allow users to refine or complete an analysis initiated by the system
- **Employing socially appropriate behaviors for agent-user interaction:** any interruptions by a system should be compatible with the social expectations of the user being interrupted and offered automated services
- **Maintaining working memory of recent interactions:** systems should retain recent interactions and allow the user to refer to prior objects, actions and services in their interaction with the system
- **Continuing to learn by observing:** systems should continually learn and adapt their models about users' goals and needs

# Tool Use

# Tool use

- The idea of tool use in HCI is that a computer system is a tool for *controlling something else*
- The computer, when viewed as a tool, is manipulated by users
- The user's goal is to do something beyond the interface itself, to affect some desired change in the world
- By using tools, users extend their own capabilities



# Utility

- The **utility** of an interactive system concerns its match with the tasks users want to do
  - High utility: the match is good
  - Low utility: the tasks that users want to do are not supported
- The tasks that users want to do with it may be existing tasks or some tasks they have not yet realized
- Users may realize their wants only when using a system, which makes the prediction of utility inherently hard
- Even routine users of a software may not be able to articulate the utility they gain from it
- Utility is about the relation between functionality and users' needs and wants
- *Getting-the-right-design* and *getting-the-design-right*
  - Whether there is functionality in a system that in principle can do what is needed is utility: getting-the-right-design
  - Whether people can do anything, in practice, concerns—among other things—usability: getting-the-design-right

# Usability

- **Usability** concerns how easily computer-based tools may be operated when users try to accomplish a task
- Usability is different than *utility*
- Usability concerns whether users can *actually use* the product in a way that makes it possible to realize its utility
  - Utility is about whether that goal was important for the user to begin with
- The ideal is that the user can use the tool without unnecessary effort so that the use itself becomes direct, transparent, and unnoticeable
- The ISO 9241 11 definition of usability:
  - ”the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use”

# Usability

- Usability is **relational**
  - It arises as an interplay between people, tasks (problems), and interactive systems (tools)
- Usability **emergent**: it only arises when people actually use tools to achieve goals
  - Because of this, it makes no sense to talk about the usability of an interactive system without considering the users and their task
- Usability is **measurable** and **multi-dimensional**
- The table shows two models of usability dimensions (ISO and Nielsen)

| Model | Dimensions | Definition | Example measures |
|---|---|---|---|
| **ISO** | Effectiveness | The accuracy and completeness with which users achieve goals | Binary task completion, error rates, quality of outcome |
| | Efficiency | The resources spent by users to achieve goals | Task completion time, input rate, mental effort, learning |
| | Satisfaction | The users' comfort with and positive attitudes towards the use of the system | Preference, perception of ease-of-use, willingness to recommend to others |
| **Nielsen** | Easy to learn | The user should be able to rapidly start getting work done with the system | Self-reported learning effort; Number of hours required to achieve a minimum level of proficiency |
| | Efficient to use | Once users are proficient, how high level of productivity can they achieve | Task completion time and efficacy in representative tasks |
| | Easy to remember | Users should be able to return to a system without needing to relearn how it works | Times a manual or help is needed; change in user performance after a period of non-use |
| | Few errors | Users should be able to operate a system with as few errors as possible | Number of erroneous or hazardous actions |
| | Subjectively pleasing | Users should find a system pleasant to use | Several questionnaire-based metrics |

# Example: system usability scale (SUS)

- Users should give their immediate response to the questions below, after having used the interactive system to be assessed
- SUS consists of ten questions answered on a Likert scale (from “strongly disagree” to “strongly agree”, coded 1 to 5)
- The aggregated score is the output, not the answers to the individual questions
- The aggregated score is indicative of the satisfaction with the interactive system and may be compared over versions of the system and across systems

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very awkward to use (changed from "cumbersome" as recommended by [8].).
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Brooke, J. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* **189**(194): 4-7.

# Acceptability

- People may choose to use a particular tool, they might use something else to solve the task, or they might give up altogether in solving the task
  - This concerns the **acceptability** of the tool, that is, whether users eventually choose to use a tool or not, assuming they are given that choice
- **Practical acceptability** includes cost, the reliability of the interactive system, and its compatibility with other systems
  - Perceptions of utility and usability may also enter the judgment of practical acceptability
- **Social acceptability** concerns whether interactions map well to the social norms and roles in the settings where they are use
- Acceptability includes the choice to *not* use a system
  - The studies of non-use in HCI suggest some processes and reasons why an otherwise useful, accessible, and usable interactive system may nevertheless not be used by the intended user group
  - Such considerations include bias and accessibility

# Accessibility

- An interactive system may offer utility and be usable, however, it may be so for a very limited group of people
- Frequently we want to ensure that products may be used by as large a group of users in as diverse situations as possible
- We call that want **accessibility**
- Communities with accessibility concerns include the following:
  - (1) blind or low-vision; (2) deaf or hard of hearing; (3) autism; (4) intellectual or development disability; (5) motor or physical impairment; (6) cognitive impairment; (7) older adult; (8) general disability or accessibility; and (9) other
  - However, each individual user uniquely manifests these at different degrees and in different combinations
- A challenge in accessibility research in that there are many disabilities and solution strategies that work for one community, or even one particular disability, may not be applicable elsewhere
- Users may have multiple disabilities, which may raise unique requirements

# Automation

# Automation

- **Automation** is the allocation of tasks to machines
- Computer systems include several examples of everyday automation:
  - Autocompletion and error correction: when entering text on a mobile device, language models propose completions to phrases and correct typos
  - Enhancement: photographs that we take on mobile devices are automatically enhanced, saving post-snap photo editing
  - Decision-support: AI models are used to calculate insurance fee suggestions for insurance analysts based on customers given available health and other data
  - Shared control: semi-autonomous vehicles can drive some parts of a route autonomously
  - Multimodal input: gesture recognition and speech recognition are available in smart devices, such as smartwatches and intelligent assistants
  - Recommendations: recommender engines are not only used for movies and music, but also in creative activities, such as proposing possible slide layouts to an information worker or poster designs to a graphic designer

# Types and levels of automation

- The type of automation maps to a particular stage in human information processing
- The level of automation ranges from fully autonomous (10) to no automation (1)

| Type of automation | Human information processing stage |
|---|---|
| Acquisition automation | Sensory processing |
| Analysis automation | Perception/working memory |
| Decision automation | Decision making |
| Action automation | Response selection |
| Adaptive automation | The type and level of automation is allowed to vary depending on context. |

| Level of automation | Description |
|---|---|
| 10 | The computer decides everything, acts autonomously, ignoring the human. |
| 9 | The computer informs the human, only if it, the computer, decides to. |
| 8 | The computer informs the human only if asked. |
| 7 | The computer executes automatically then necessarily informs the human. |
| 6 | The computer allows the human a restricted time to veto before automatic execution. |
| 5 | The computer makes a decision to perform automatic execution if the human approves. |
| 4 | The computer suggests one alternative action. |
| 3 | The computer narrows the selection of alternative actions down to a few. |
| 2 | The computer offers a complete set of decisions/action alternatives. |
| 1 | The computer offers no assistance: the human must take all decisions and actions. |

# Types of automation

- **Acquisition:** system sensing and registering of input data
- **Analysis:** automation of information analysis
  - Involves inferential processes
  - Low-level automation: extrapolation or prediction of data over time
  - Moderate level of automation: a system integrating multiple sources or input variables
- **Decision:** deciding and selecting appropriate actions among decision alternatives
  - Automation that replaces or augments human decision making
  - Examples of decision automation include route planning and route adaption, for example, to avoid bad weather, and systems providing medical diagnosis support
  - The difference between analysis and decision automation is that decision automation means the system must make implicit or explicit assumptions about costs and values inherent in all decisions
- **Action:** the machine is partially or fully executing an action choice
  - Range from low-level, such as collapsing several key presses into one, to high-level, such as automated driving systems
- **Adaptive:** the type and level of automation is allowed to vary depending on context
  - For example, a situational demand requiring a rapid response makes the system take over control

# User-centric evaluation criteria

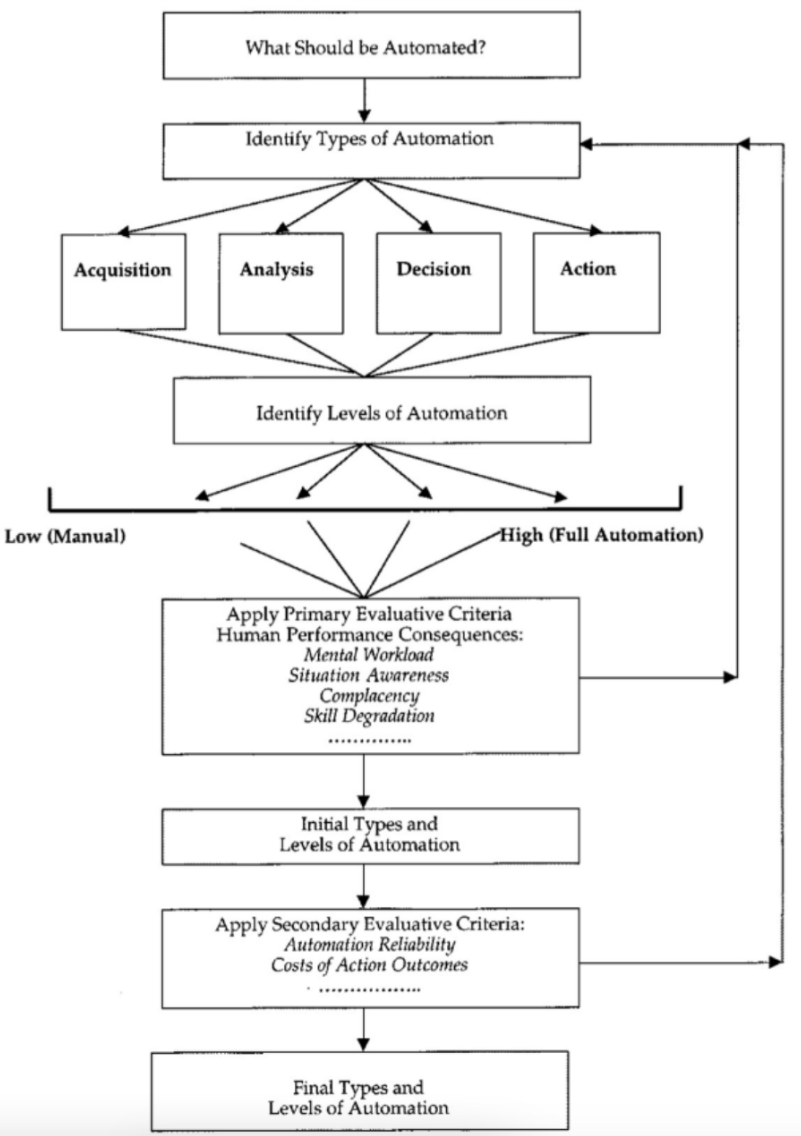
- Automation can both reduce and increase users' **mental workload**
  - For example, automation can enable an organization of information that makes it is easier for users to understand
  - However, mental workload can also increase due to systems that, for example, are difficult to initiate or engage, difficult to understand, or require extensive physical operations
- Automation can also affect users' **situation awareness**
  - Automation may reduce users' awareness of the system and its dynamics, including emergent consequences
  - In general, humans tend to be less aware of changes in a system or environment if those changes are under the control of another agent
- Automation can give rise to **complacency** arising from overtrust or overconfidence in automation
  - For example, a user may fail to monitor automation or information sources if the automation is at a high level and is perceived to always be accurate
  - There is a risk of errors being introduced due to users overly trusting imperfect automated information analysis, such as filtering or prediction
- Automation may result in **skill degradation**
  - Users forgetting how to carry out functions or witnessing their skills decay

# System-level evaluation criteria

- **Automation reliability:** the extent the system is effective in automation
  - True positive rate (sensitivity): the ability of a system to correctly diagnose a problem or identify an object
  - False positive rate (false alarm rate): the system's inability to separate positive from negative cases
  - A high false alarm rate may cause fatigue among users and may foster distrust in the system
- **Costs of decision and action outcomes: a**ny potential benefits of automation has to be compared with and weighted against any possible disadvantages
  - Disadvantages include an increased mental workload, a reduced situation awareness, an increase in complacency, and a risk of skill degradation

# Types and level of automation framework

1. Identify what should be automated
2. Identify the types of automation that are applicable
3. Identify the levels of automation that are suitable for the tasks
4. Evaluate based on primary evaluation criteria, the human performance consequences of automation
   - Revisit the identification of types and levels of automation following new insight from evaluations based on primary evaluation criteria
5. Arrive at initial types and levels of automation for the task
6. Apply secondary evaluation criteria, automation reliability and the costs of decisions and outcomes
   1. Revisit the identification of types and levels of automation following new insight from evaluations based on secondary evaluation criteria
7. Arrive at final types and levels of automation that are suitable for the tasks




R. Parasuraman, T. B. Sheridan, and C. D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* **30**(3):286–297, 2000.

# Practice

# Practice

- Interactive computing systems are astonishingly flexible
- Consider for example all the different ways you could send an email: all the applications you could choose to do it, the messages that could be written, the recipients that could be chosen, and all the different settings that could be chosen
- This overwhelming freedom raises a question:
  - *Why is the large degrees of freedom not choking our ability to use computers; how come we are able to get our things done—at least most of the time?*
- The answer is **practice**

# Example: observing the development of practice in groupware use

- Practice is a central issue with systems that aim to intertwine with their users' work, particularly how they communicate and collaborate
- Over five months, Orlikowski conducted interviews, reviewed documents, and carried out observations to study the introduction and initial use of *Notes* in a consulting company
- *Key finding 1*: how individuals think about interactive systems and their work influence how they adopt systems, in this case, *Notes*
  - When individuals are confronted with new technology, they will use their understanding of earlier technology to understand the new system
  - Those frames can be shaped by information about the new system as well as by training: Orlikowski showed that neither of these was done extensively
  - Thus, users of *Notes* mainly understood it as an individual technology, rather than as groupware, and had a weak understanding of what the system could do
- *Key finding 2*: the organization of the company influences how the system ends up being used
  - Reward systems, policies for conducting work, and work norms shape how individuals in practice view and use *Notes*
  - For instance, the company did not formulate new policies around sharing access to documents in the system or new ways of working to use *Notes* to its potential

W. J. Orlikowski. Learning from notes: Organizational issues in groupware implementation. In *Readings in Human-Computer Interaction*, pages 197–204. Elsevier, 1995.

# Adaption and adoption

- One of the critical ideas of interaction-as-practice is that interactive systems never fully prescribe how they are used in real life
- Instead, users ignore, alter, work around or creatively modify all interactive systems as part of their practice
- For example, people give input false data to trick interactive systems into providing the desired results, and keep manual or duplicate systems to work around inaccurate or wrong systems
- As another example, people use digital cameras for a host of other tasks beyond taking pictures, including for scanning documents or for periscopes to be able to see at a concert
- The point here is that interaction and use in practice is often different from that anticipated and assumed by designers

# Personalization

- **Personalization** is about changing the appearance of interactive systems, that is, it focuses on *non-functional* changes
- Dispositions to personalize depend on the user, the system, and the context of system use, such as whether one owns the computer system and frequency of use
- Cognitive effects of personalization include improved ease of use because certain parts of the system are easier to see
  - It may also be that the system as a whole is easier to recognize (such as an choosing an unusual icon for an application to make it stand out)
- Social effects include personalization that reflects one's personal identity, for instance, a feeling distinguished from other users of the same system
  - It may also show group identity, for example, allegiance to a particular brand or sports club
- Emotional effects include feelings of fun and positive associations
  - They also include reducing boredom and an increased feeling of ownership or attachment to the interactive system

# Tailoring

- Adaptations may also involve users intentionally modifying the *functionality* of interactive systems
  - This is not merely about the appearance of the system but any range of activity from users tinkering with settings and features to users reprogramming parts of the system using macros, build-in programming languages, or other means of modifications
- Such activities are called **tailoring** and there are three types of such tailoring:
- **Customization:** users adapt the product by selecting attribute values from a selection
  - For example, users may adapt the appearance and contents of their email replies
- **Integration:** users add new functionality or features to the interactive system by linking existing components
  - For example, integration may happen through macros, accelerators, or scripts
- **Extension:** users add new code or otherwise program the system

# Appropriation

- **Appropriation** describes such adaptation and adoption processes, which may also change workflows, division of labor, and organizational processes
  - Thereby, appropriation does not particularly concern modifying interactive systems but on all the changes that might be associated with interaction with a system in practice
- Appropriation is likely to occur in practice since:
  1. Designers are unlikely to understand all tasks or environments in which a product is used
  2. Users' needs and situations change

# Frequent appropriation moves

- Actively champion the use of an interactive system
- Substitute parts of the interactive system for other systems
  - This may be using a manual backup system instead of the intended system
- Use an entirely different way of accomplishing work due to a perceived deficiency in the interactive system
- Criticizing the interactive system by comparing it to other ways of accomplishing the work
- Interpreting the interactive system, for instance by explaining the meaning of functionality to others or prescribing how to use the system
- Attempt to make others reject using the interactive system or otherwise actively preventing its usage
- Being slow in taking up the system or otherwise contributing to inertia in its uptake

# Guidelines for appropriation

- **Allow interpretation:** avoid fixed meaning, but include elements where users can add their own meanings
  - For example, some email clients allow users to flag emails with colors (e.g., yellow, green, red), leaving the meaning to the user
- **Provide visibility:** providing visibility of system functioning and status will help users understand how to develop appropriations
  - Appropriating a black box, the internal working of which is unclear, is inherently hard
- **Expose intentions:** tell to the user what the intended use of a function is
  - For example, instead of simply asking a user to generate a strong password, the user can be reminded that the goal is to protect the user's data (intention)
- **Support, not control:** some tasks can be thought as procedures that need to be executed from the beginning to the end in a particular order
  - Designs should not force users to do tasks in a particular way (control) but allow them flexible variation (support)
- **Pluggability and configuration:** users should be allowed to create systems on their own
- **Encourage sharing:** by allowing users to share their appropriations, they become reappropriated by others
- **Learn from appropriation:** observing the ways users appropriate technology can provide a source of insight for product development

A. Dix. Designing for appropriation. In Proceedings of HCI 2007 The British HCI Group Annual Conference University of Lancaster, UK 21, pages 1–4, 2007.

- Open access (PDF at link)
- Further reading:
  - Part IV: Understanding Interaction
  - Chapters 16–22