

CS3724 Human-Computer Interaction

User Interaction Guidelines

Copyright © 2005 H. Rex Hartson and Deborah Hix.

Topics

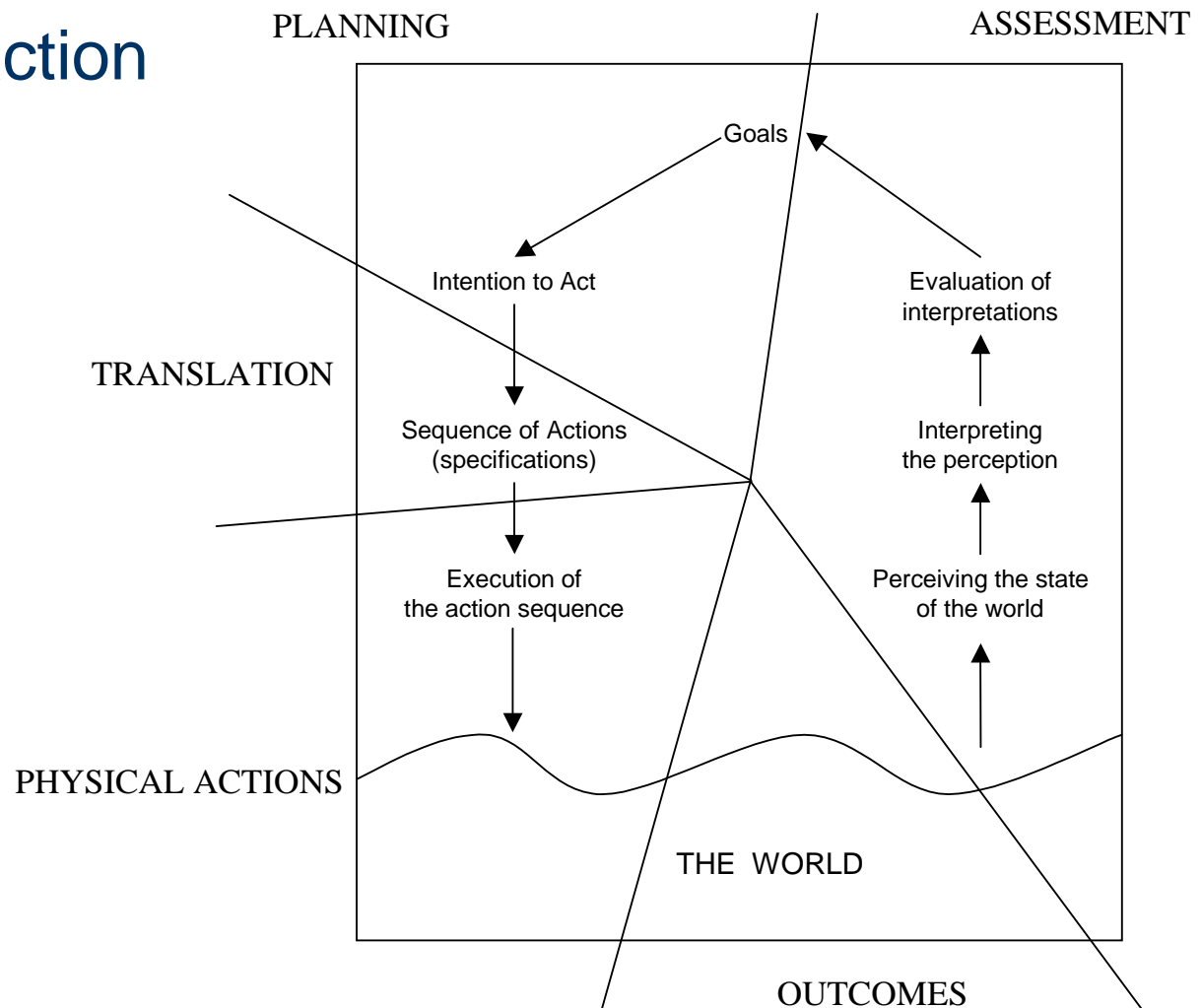
- Organizing usability issues: The User Action Framework
- The Interaction Cycle
- Affordances
- Selected usability design guidelines
- In-class exercises on guidelines

The User Action Framework (UAF)

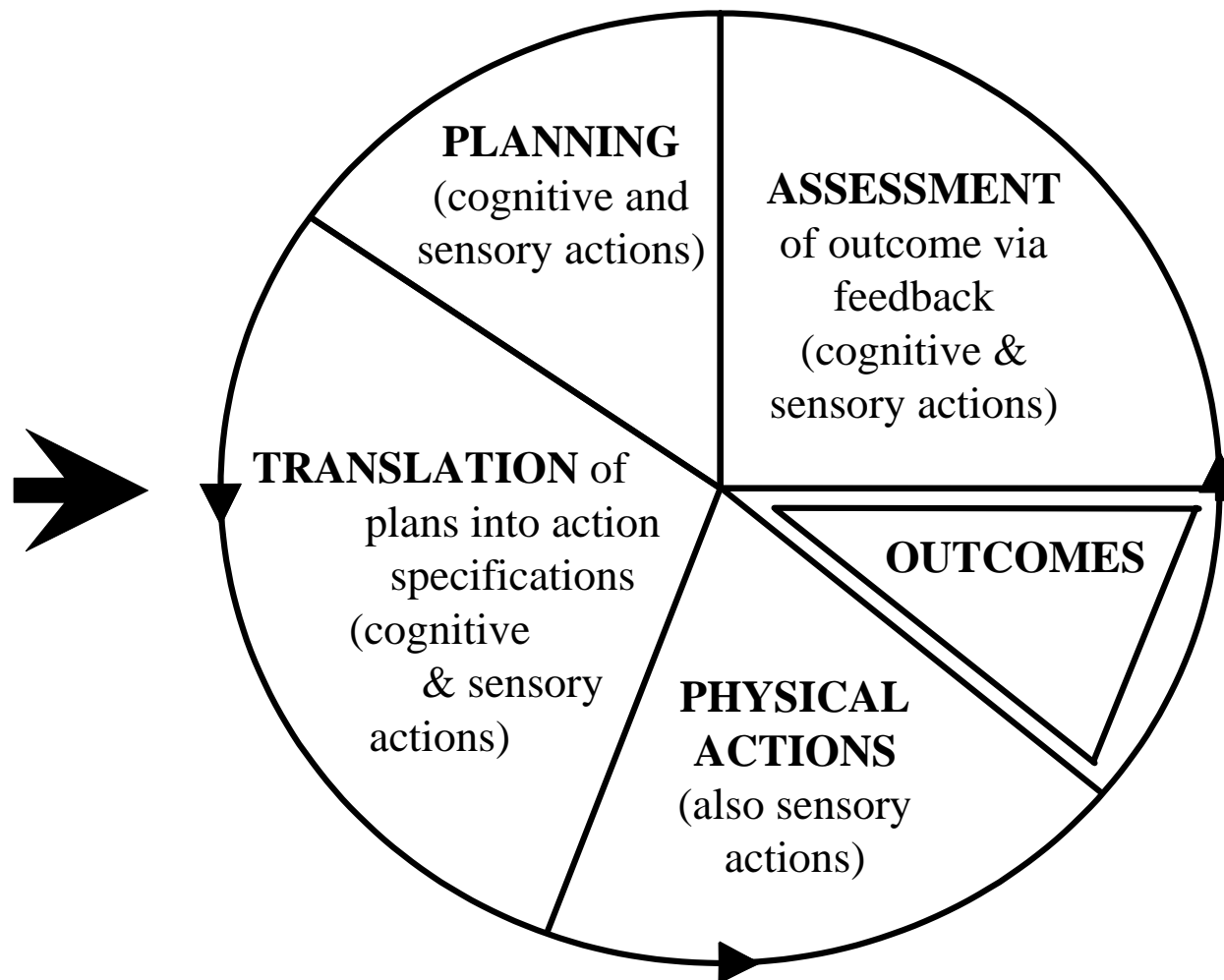
- The Interaction Cycle is highest level of categories in the UAF
 - Adapted and Extended Norman's "stages of action"
Norman, D.A. Cognitive engineering. Chapter 3 in *User centered system design: New perspectives on human-computer interaction*. Norman and Draper ed. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986, 31-61.
- All about what users **think**, **do** and **see** during cycle of interaction with computer
 - **THINK** – represents all cognitive actions
 - **DO** – represents all physical actions
 - **SEE** – represents all sensory actions

Norman's 'Stages of Action' Model

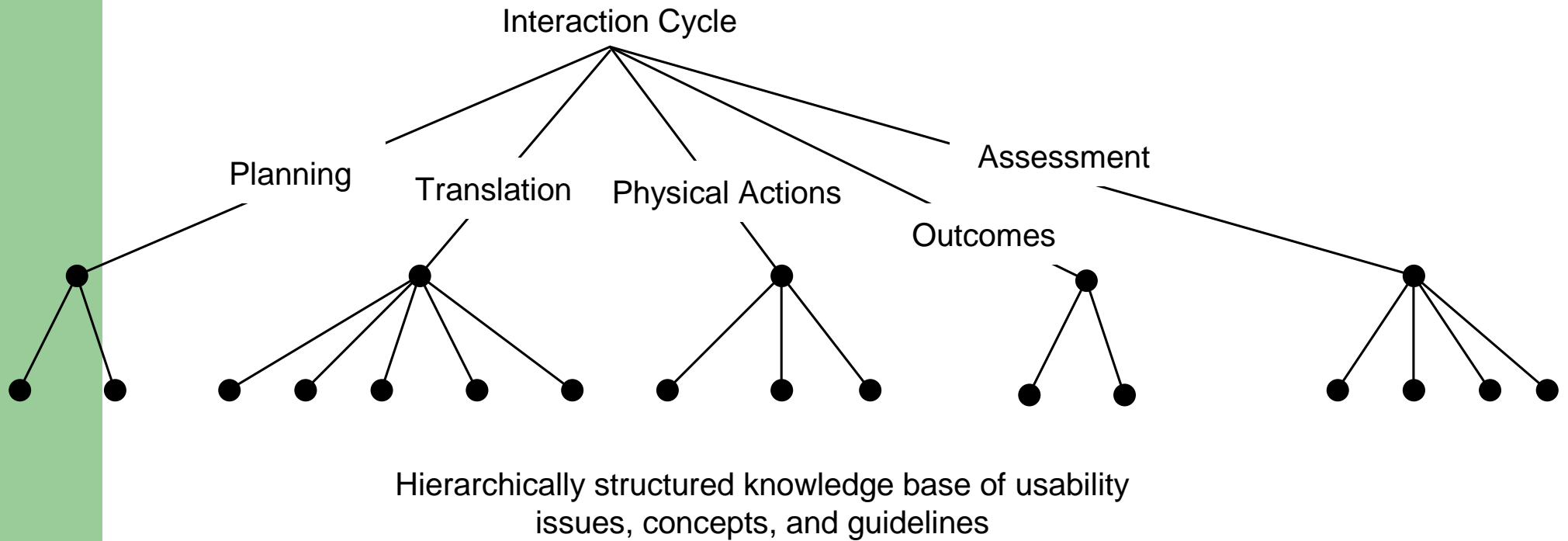
- User interaction with any machine



Transition to Interaction Cycle



Transition to User Action Framework



Norman vs. UAF

- Norman's model and our Interaction Cycle is about what users do within a cycle of interaction with a computer or other machine
- The UAF is about design, about how interaction designs support users in performing sensory, cognitive, and physical actions during interaction with a machine

The User Action Framework

- Hierarchical knowledge base of usability concepts and issues organized on Interaction Cycle
 - Organizes usability concepts in terms of user actions during task performance
 - Works for interaction with any kind of machine, any style of interaction
 - Puts usability problems in context of relevant design guidelines and principles
 - Not claimed complete, but self-extending

The User Action Framework

- Integrated framework for UE tools for:
 - Usability inspection
 - Usability problem analysis (problem extraction and diagnosis)
 - Usability problem reporting
 - Usability data management
 - **Design guidelines**

Affordances

- Affordances* – an essential concept
 - An affordance gives or provides something that helps a user do something
 - In interaction design, affordances are attributes of user interface artifacts that help users perform tasks

Affordances

- Cognitive affordance
 - A cognitive affordance is a design feature that helps, aids, supports, facilitates, or enables thinking and/or knowing about something
 - Example: Clear and precise words in button label enabling users to understand meaning of button in terms of functionality behind it and consequences of clicking on it
 - Plays starring role in interaction design for less experienced users

Affordances

- Physical affordance
 - A physical affordance is a design feature that helps, aids, supports, facilitates, or enables physically doing something
 - Example: Adequate button size and easy-to-access location enable users to click easily on the button
 - Plays starring role in interaction design for experienced (power) users

Affordances

- Sensory affordance
 - A sensory affordance is design feature that helps, aids, supports, facilitates, or enables user in sensing (e.g., seeing, hearing, feeling) something
 - Includes design features or devices associated with visual, auditory, haptic/tactile, or other sensations
 - Plays critical supporting role to cognitive affordance and physical affordance
 - Example: Button label text in font large enough and appropriate color to support legibility

Affordances

- Functional affordance
 - Functionality of non-IU software
 - Adds purpose for physical affordance
 - Adds sense and goal orientation to design discussion
 - Connects usability with usefulness (of system functionality)
 - About higher-level user enablement in the work domain

Affordances

- Affordances work together in design
 - Example: Devices for opening doors (round doorknobs and lever-type door handles)
 - Visual design of each conveys cognitive affordance via implied message “this is what you use to open a door”
 - Doorknob and lever suggest, each in own way, grasping and rotating for door operation
 - But message is received only because of shared cultural conventions

Affordances

- Affordances work together in design
 - Door operating devices also provide physical affordance, to help users open and close doors
 - Some devices work better than others as physical affordances
 - Push-bar on double doors is another type of physical affordance for doors

Affordances

- False cognitive affordances misinform and mislead
 - Example: Web page links that only look like buttons

Simple Search

Advanced Search

Browse

Register

Submit to CoRR

About NCSTRL

OAI

Help

- The “booby-trapped” X in a pop-up advertisement
- Horizontal line in Web page that falls at bottom of screen

Affordances

- Trails of user-made artifacts
- Tape added to shovel handle, Post It note added to monitor or keyboard

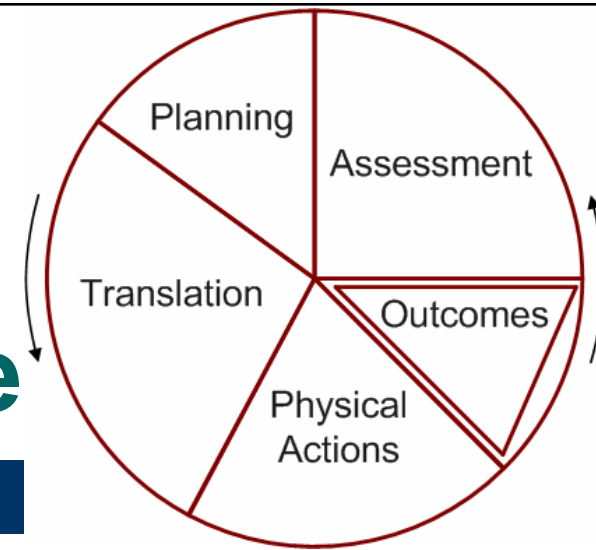


Affordances

- Trails of user-made artifacts

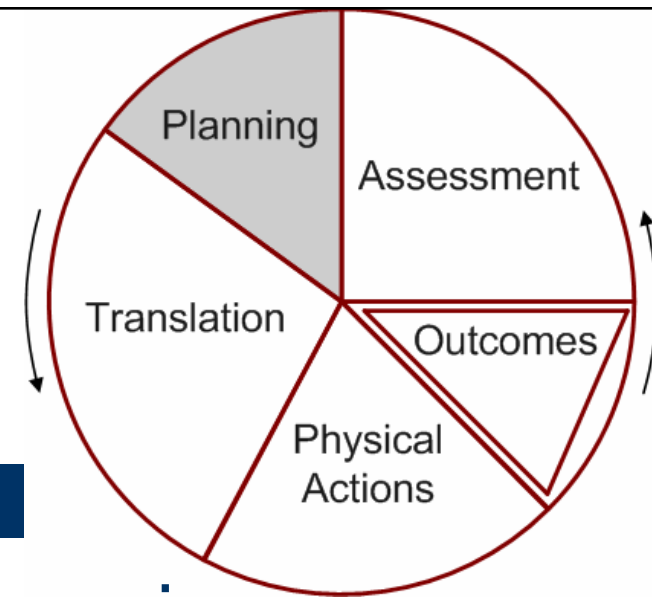


The User Interaction Cycle



- Simplest view of the Interaction Cycle
 - Planning (What to do)
 - Translation to determine actions (How to do it)
 - Physical actions (Doing it)
 - Assessment of outcome (Did it turn out right?)
- Next: Selected design guidelines for each part of the Interaction Cycle

Planning – User Model of System

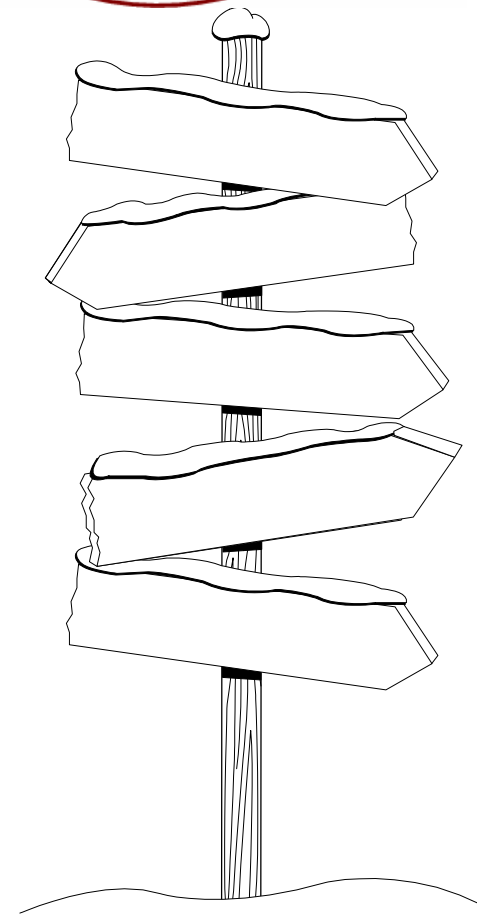
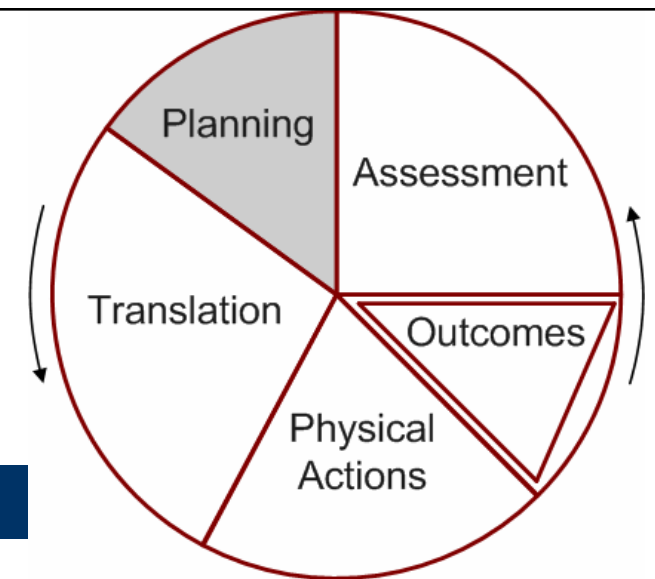


- Provide clear model of how users view system in terms of tasks
 - Help users with system model, metaphors, work context
 - Help users plan goals, tasks
 - Help users decompose tasks logically

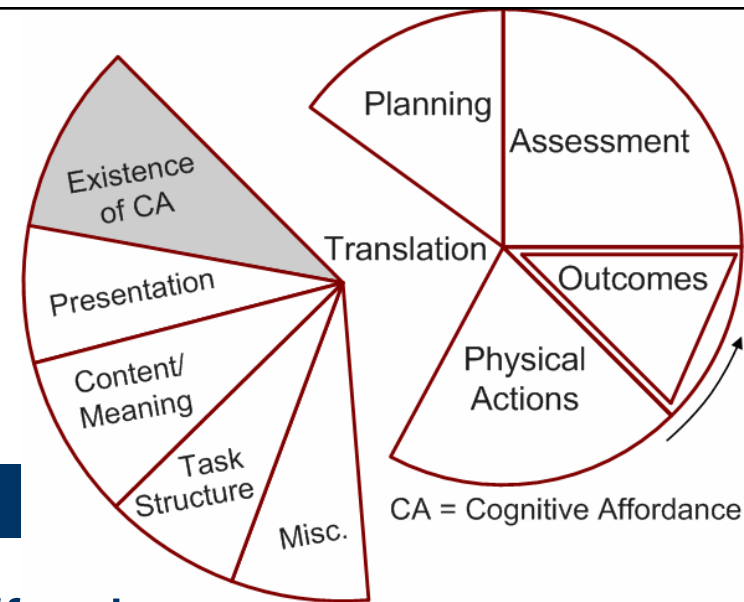
Monsters

Planning - User Awareness

- Make possibilities clear what users can do at every point
- Keep users aware of system state for planning next task
- Keep users aware of task progress (what's been done and what's left do)

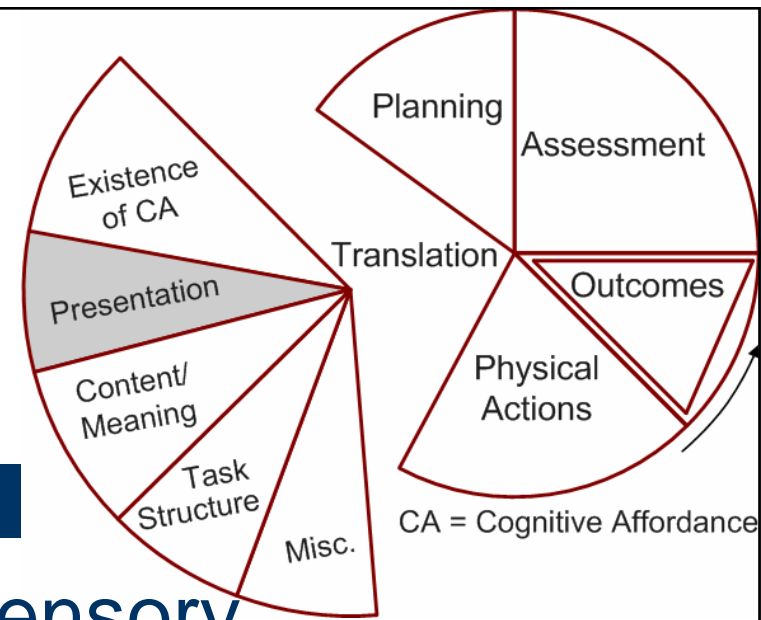


Translation - Existence



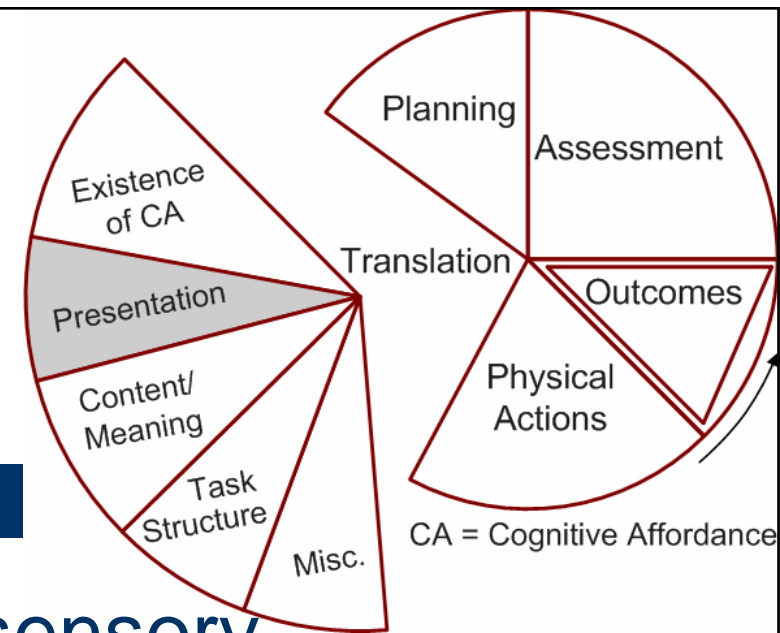
- Existence of effective cognitive affordances – cues (e.g., in labels, data field formats, icons) that help users get access to system functionality
 - Help users know/learn what action are needed to carry out intentions
 - Users get knowledge from experience, training, AND: 0cognitive affordances in design
 - Help users predict outcome of actions
 - Help users determine what to do to get started

Translation - Presentation



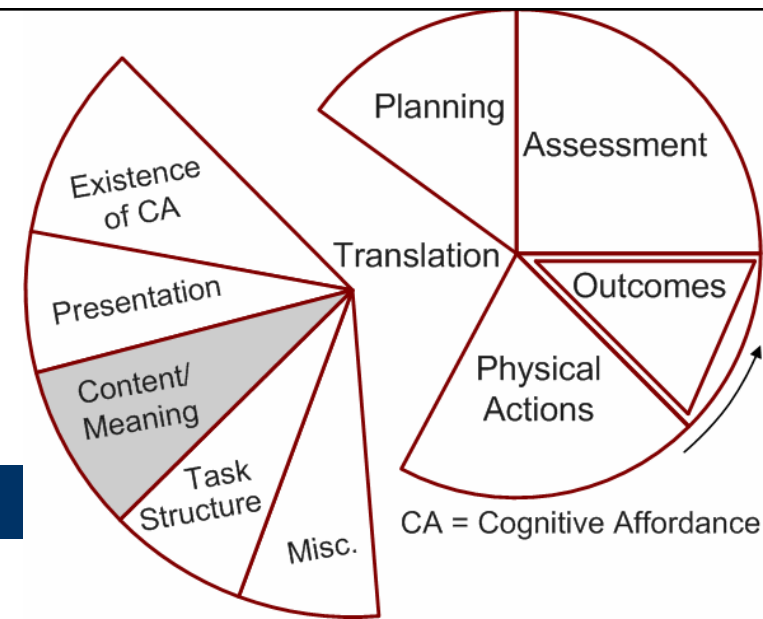
- Support user with effective sensory affordances in **presentation** of cognitive affordances
 - Make noticeable
 - Object contrast, size layout complexity, location with respect to user focus
 - Make legible, readable (discernable)
 - Font size, font type, font color, font contrast
 - Avoid irritation in presentation of cognitive affordances (e.g., color, blinking, audio)

Translation - Presentation



- Support user with effective sensory affordances in **presentation** of cognitive affordances
 - Control complexity with effective layout, organization, and grouping
 - Avoid screen clutter

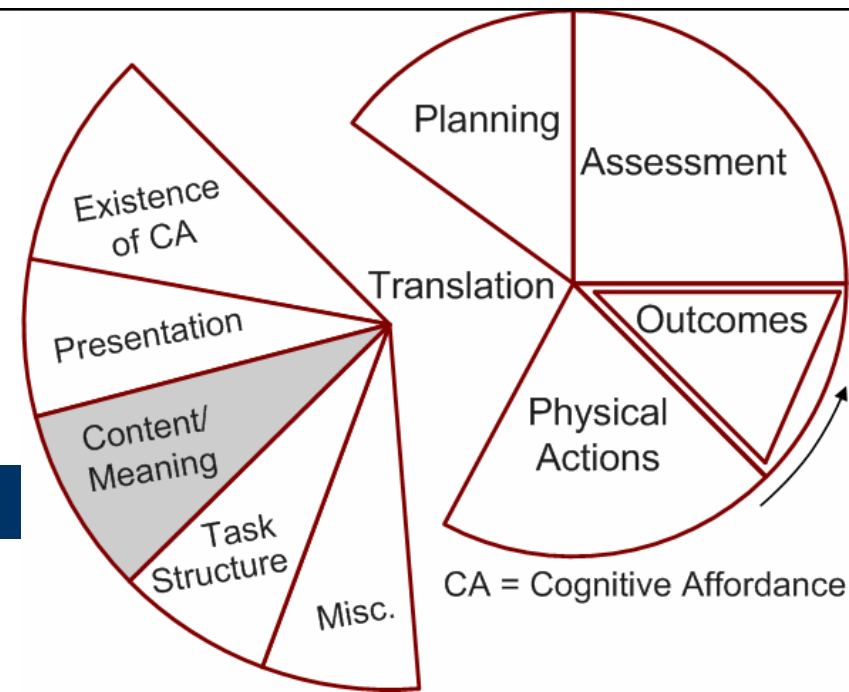
Translation – Content/Meaning



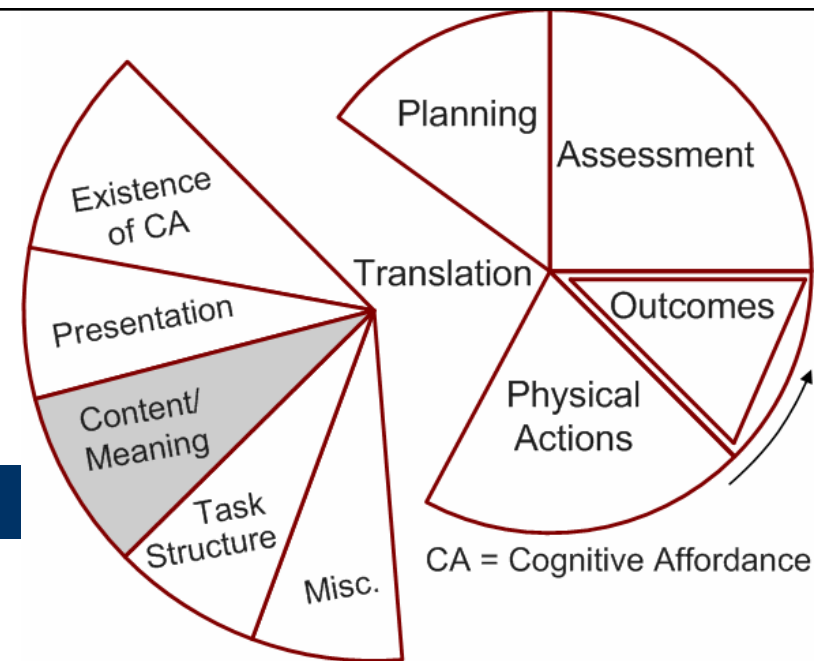
- Help user determine actions with effective content/meaning in cognitive affordances
 - Design for clarity
 - Use precise wording in labels, menu titles, menu choices, icons, data fields
 - Example: “adjust speed” or “clockwise to increase speed” rather than “adjust” or “speed”
 - Use dynamically changing labels when toggling (e.g., Play/Pause, Partial view/Full view)

Translation – Content/Meaning

- Design for clarity
 - Provide clearly marked exits
 - Example: “Return to XYZ” instead of Cancel or Exit
 - Provide clear “Do It” mechanism
 - Example: for add-record task, use “Add Record” instead of just Ok or Return
 - Be predictable; help users predict outcome of actions (feedforward)
 - Predictability helps learning and error avoidance



Translation – Content/Meaning



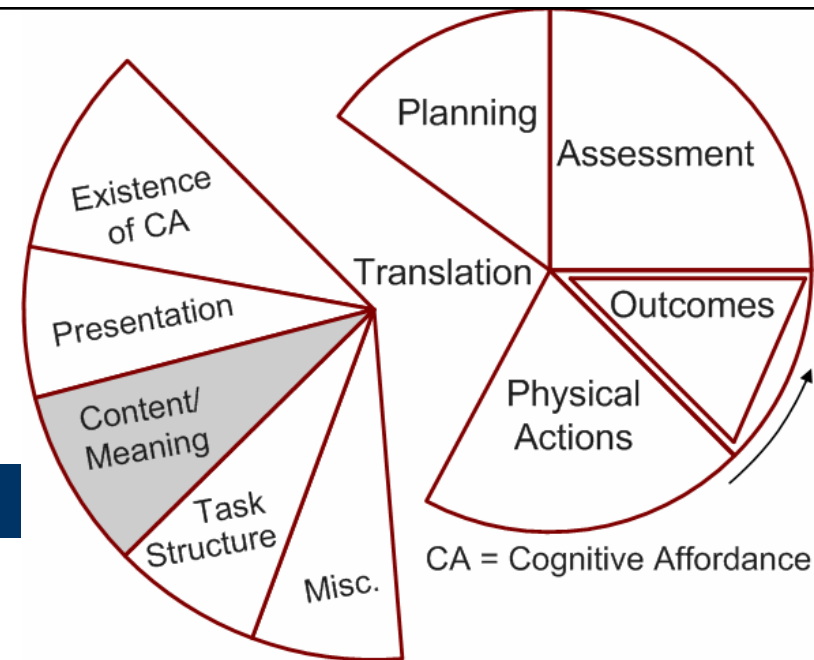
– Design for clarity

- Be consistent (a guideline with interpretation difficulties)
 - Consistency: Similar semantics $\leftarrow \rightarrow$ similar syntax (wording or user actions)
 - Use consistent wording in labels for menus, buttons, icons, fields
 - Custom style guides help consistency
- Use appropriate layout and grouping by function to convey content and meaning

Continue vs. retry

Group by function

Translation – Content/Meaning



– Design for clarity

- Furnish useful defaults (e.g., most likely values, cursor position)

Use most likely date as default; also shows format

- Support human memory limits with recognition over recall

Show file names

– Design for cognitive directness

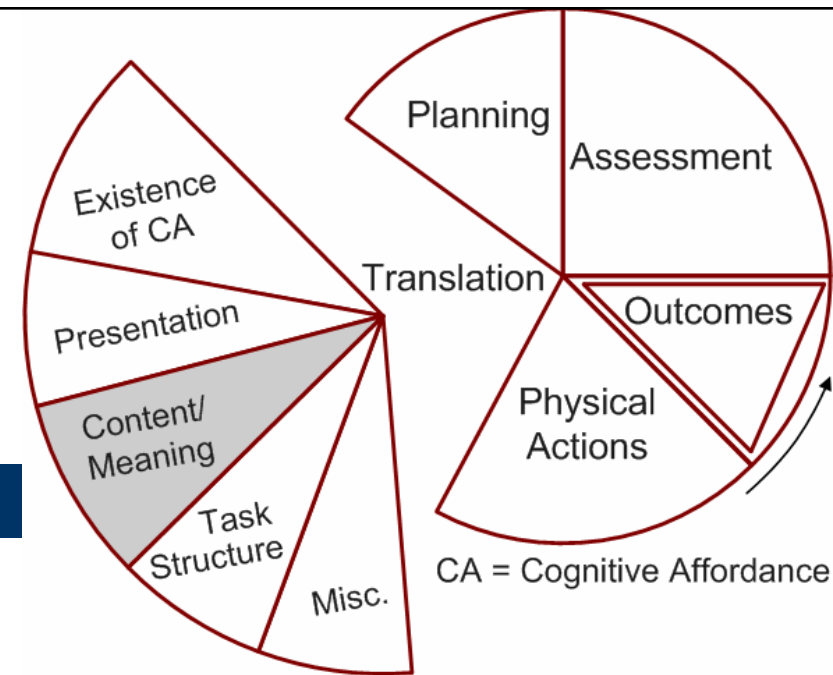
- Minimize mental transformations
- Examples

– Dreamweaver ftp function

DreamWeaver arrows

– Others, thanks to Paul Kemmerling

Translation – Content/Meaning

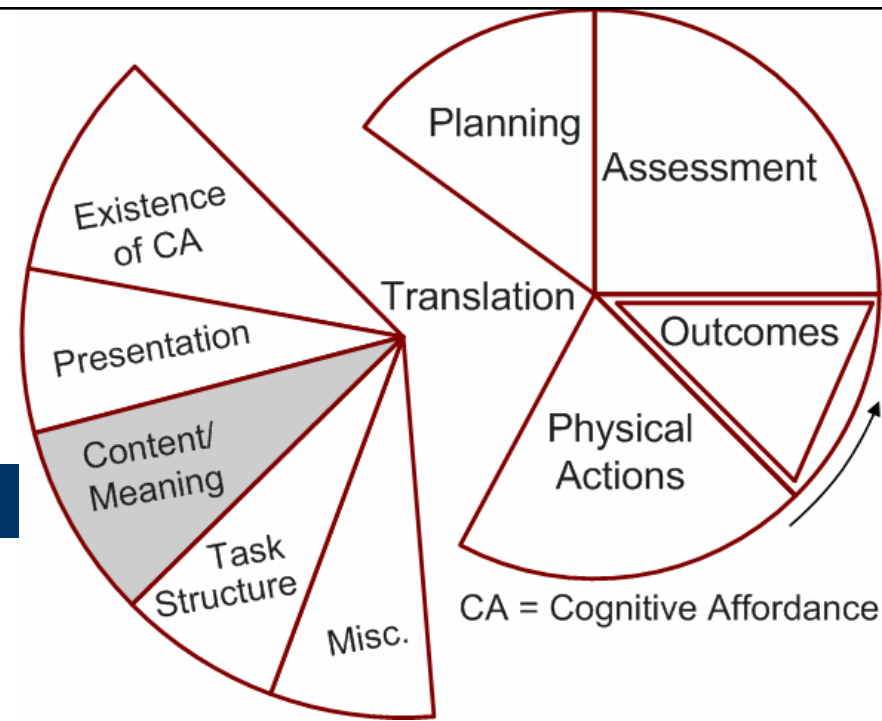


- Design for completeness
 - Use enough words for unambiguous labels
 - Long labels are not necessarily bad
 - Give enough information for users to make confident decisions
 - Revert to saved Outlook exit msg
 - Prevent loss of productivity due to hesitation, pondering
 - Give enough alternatives for user needs

Erase now?

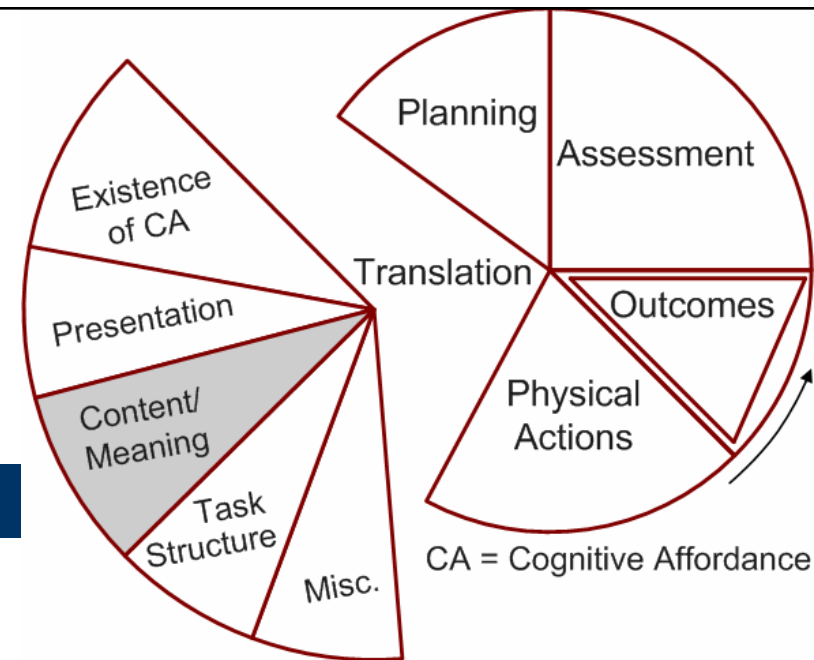
Translation – Content/Meaning

- Help users avoid errors
 - Example
 - Disable buttons, menu choices to make inappropriate choices unavailable
 - Gray out to make inappropriate choices **appear** unavailable
- Offer constructive help for error recovery
 - “To err is human; forgive by design”
 - Provide clear way to undo (multiple levels) and reverse actions

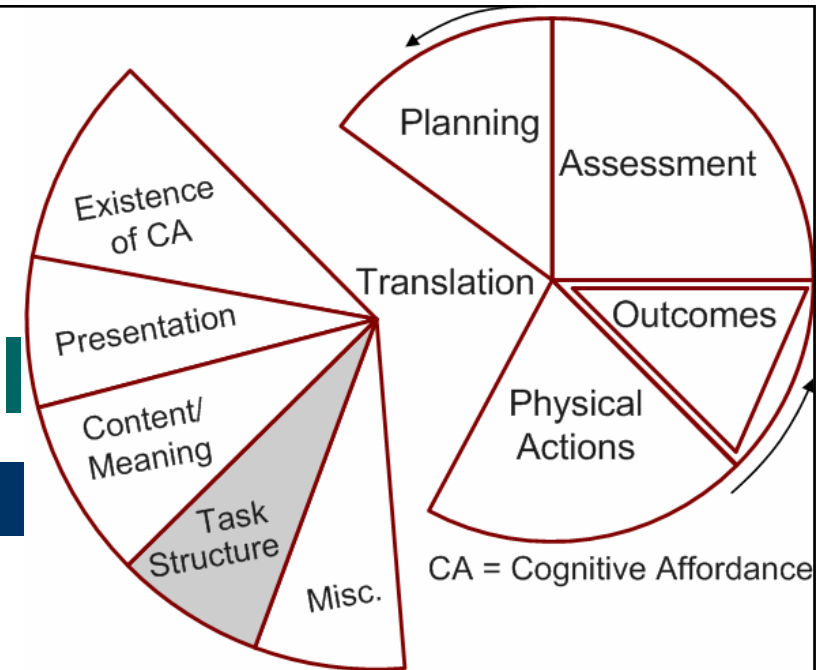


Translation – Content/Meaning

- Design carefully for modes
 - Modes are states where actions have different meanings
 - Distinguish modes clearly
 - Avoid confusing modalities
 - Users cannot easily shift focus
 - Even works against expert users

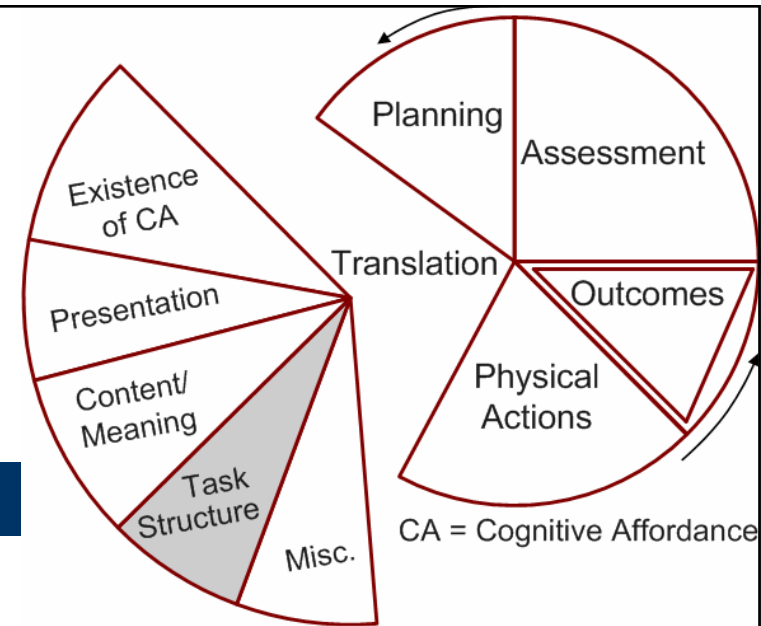


Translation – Task Structure and Control



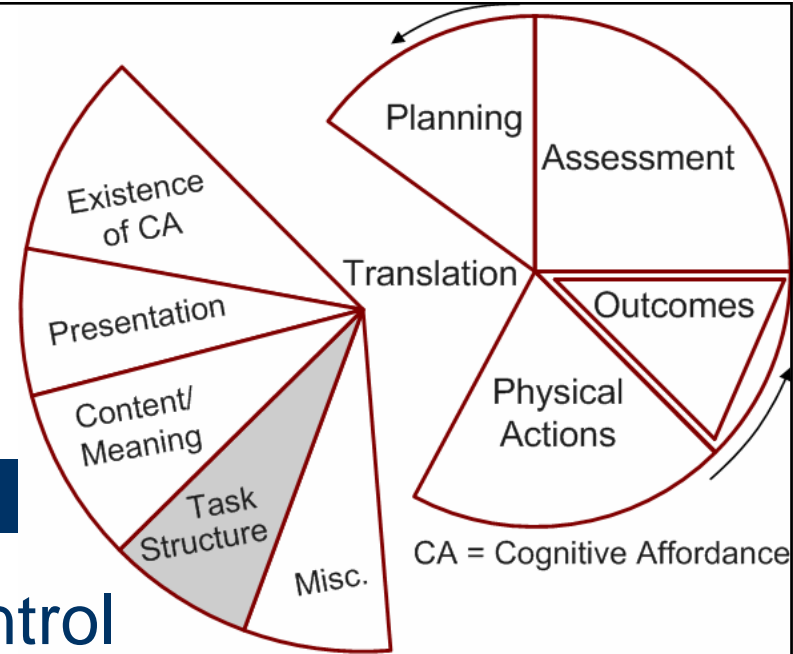
- Support user with effective task structure and interaction control
 - Keep users in control
 - Avoid *feeling* of loss of control (e.g., bossy attitude projected to users)
 - Example: “You need to answer your mail” or “Enter next command” vs. “Ready for next command”

Translation – Task Structure and Control



- Design task structure for flexibility and efficiency
 - Provide alternative ways to perform task
 - Provide shortcuts (e.g., hot keys)
 - Make the most of user's work
 - Examples:
 - User fills out part of form & goes away; don't let them return to empty form
 - Retain user preferences; retain navigation through directory structures !!
 - Avoid requirement to retype or copy from one place to another

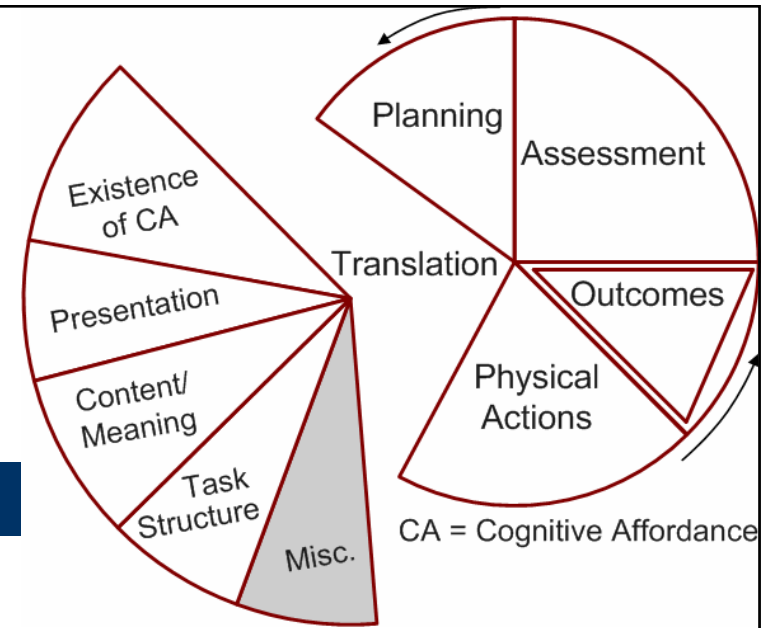
Translation – Task Structure and Control



- Design natural interaction control
 - Give direct manipulation support
 - Example: Direct editing of text object instead of requiring user to ask system to put it away
 - Anticipate likely related tasks; support task thread continuity
 - Example: if message suggests something, offer an easy way to do it
- Always provide a way to ‘bail out’
 - Example: Error message box has buttons for Task A, Task B (not enough: needs Cancel, too)

[New folder in Save as](#)

Translation – Different User Classes



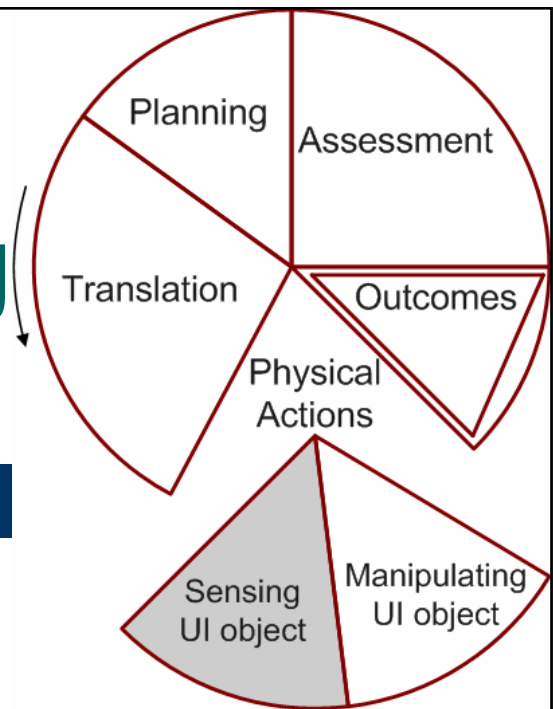
- Accommodate different user classes with preferences

Lead	Novice
Follow	Intermittent
Get out of the way	Expert

System's a snap

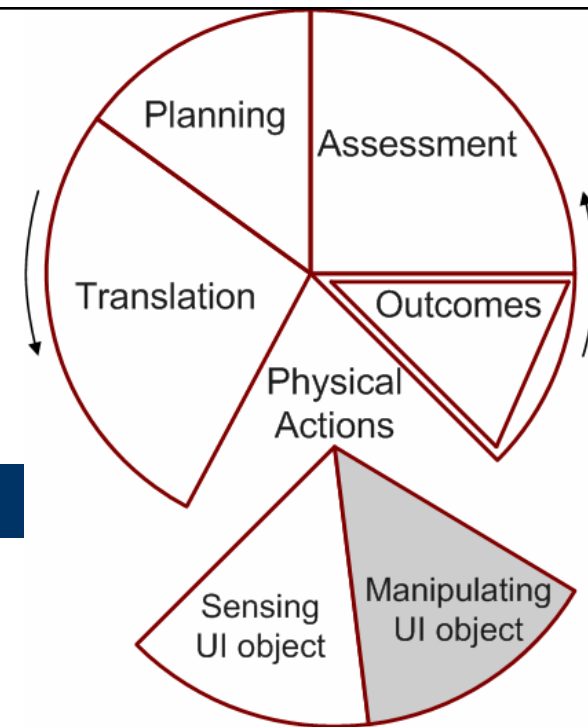
- Don't let affordances for new users be performance barriers to experienced users

Physical Actions – Sensing Objects to Manipulate



- Support user with effective sensory affordances for **sensing** physical affordance – e.g., help in *seeing* objects to manipulate
 - Make objects discernable, legible, noticeable
- Support user with effective physical affordances for manipulating objects – help in *doing* actions

Physical Actions – Doing Actions

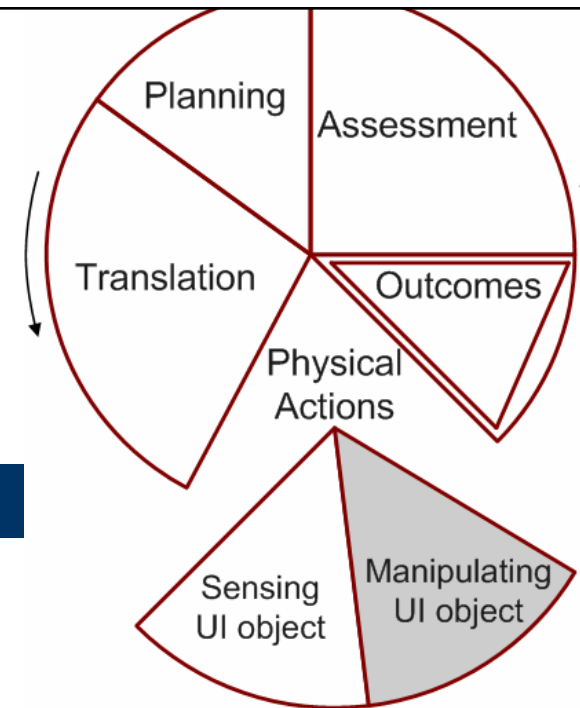


- Avoid physical awkwardness
 - Example: Time-consuming switches between multiple input devices (e.g., mouse and keyboard, touchscreen)

Awkwardness

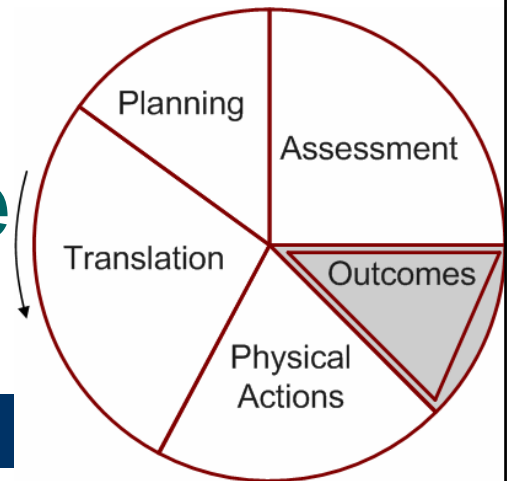
- Accommodate physical disabilities-limited motion, motor control, vision, hearing

Physical Actions – Doing Actions



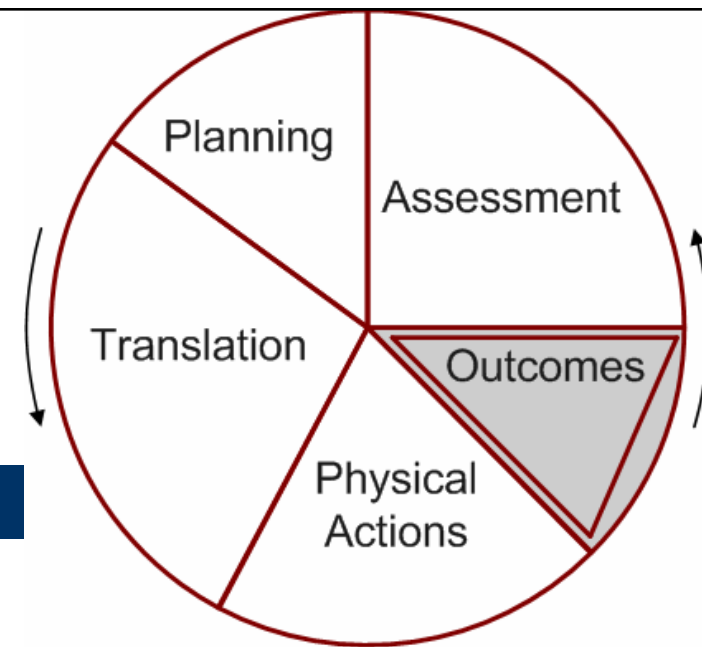
- Design layout to support Fitts' law, manual dexterity
 - Support hand-eye coordination limits by making selectable objects large enough
 - Locate related clickable objects close together
 - Avoid fatigue, slow performance
 - But not too close
 - Avoid erroneous selection

The System's Turn – Outcome and System Response



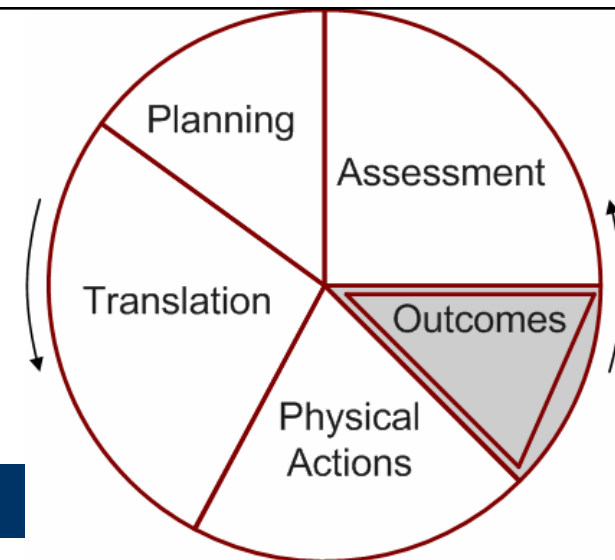
- Outcome is internal computation or state change
 - Not directly visible to user
 - Interaction designer must make visible via feedback in system response
- System response is only way user knows about outcome of actions
 - System response can contain:
 - Feedback – information about course of interaction so far
 - Information display – results of outcome computation
 - Feed-forward – information about what to do next

System Response



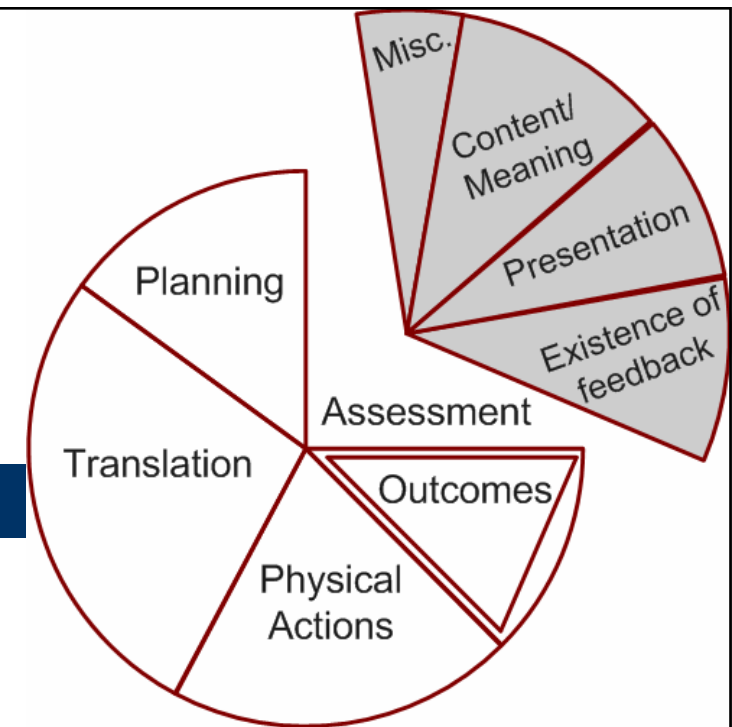
- System response example
 - “The value you entered for your name was not accepted by the system.” (feedback → Assessment)
 - “Please try again using only alphabetic characters.” (feed-forward → Translation)

Outcomes



- Includes all issues about system (non-UI) functionality
 - Missing features
 - Non-UI software bugs
- Avoid too much automation and *real* loss of control
 - Example: Changing folder name “IRS” to “Irs”
 - Unnecessarily preemptive dialogue box interrupts normal planning

Assessment



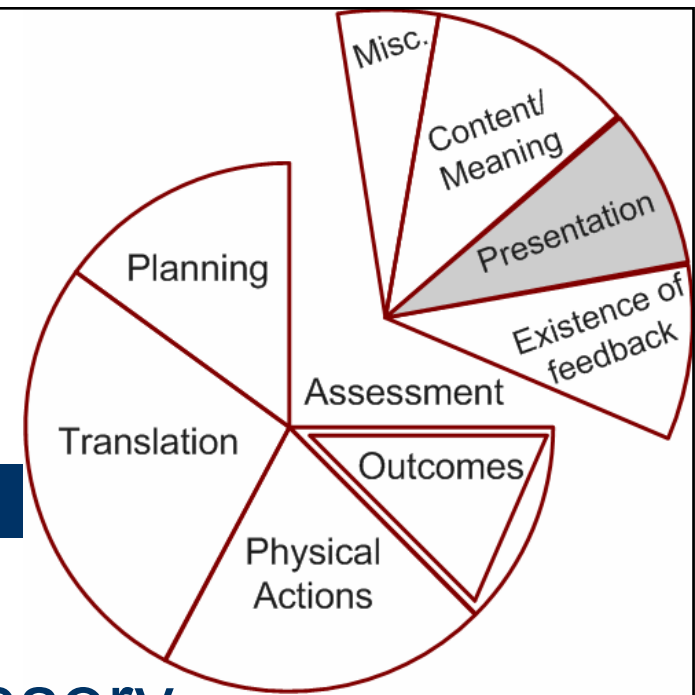
- Assessment issues are similar, parallel to those for Translation
 - Existence (of feedback)
 - Presentation (of feedback)
 - Content, meaning (of feedback)

Assessment - Existence



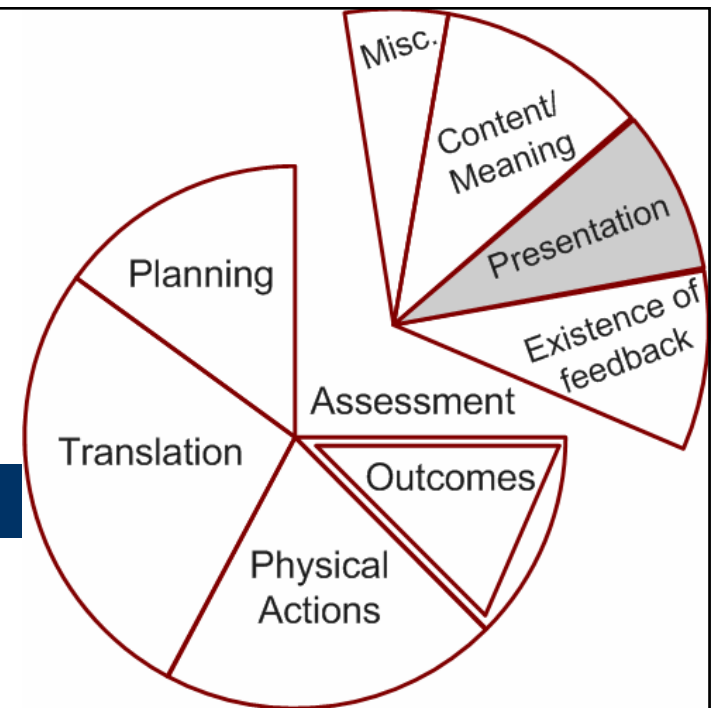
- Make sure of **existence** of feedback
 - Provide feedback
 - No news is no news!
 - Feedback keeps users on track
 - Provide progress report on long operations (e.g., percent-done indicator)
 - Request confirmation as a kind of intervening feedback, to prevent errors (especially for potentially destructive actions)
 - But don't overuse and annoy

Assessment – Presentation



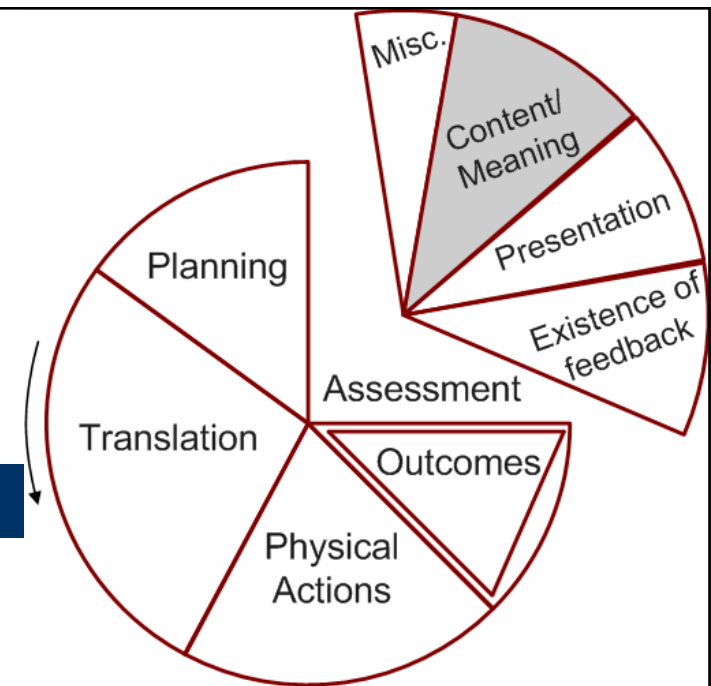
- Support user with effective sensory affordances in **presentation** of feedback
 - Make feedback noticeable
 - Locate feedback within user focus of attention
 - Make large enough to see
 - Present feedback promptly
 - Make feedback persistent (avoid flashing)

Assessment – Presentation



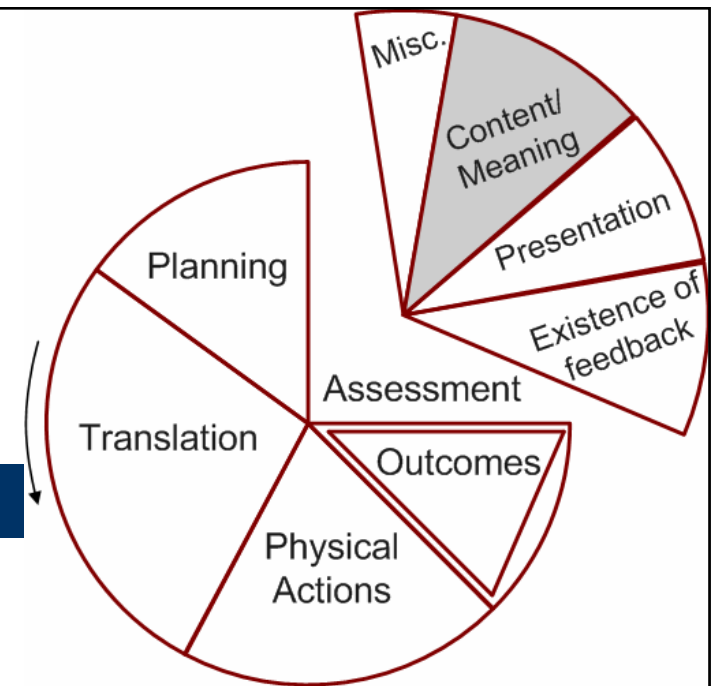
- Use most effective presentation medium
 - Consider audio as alternative channel
 - To get attention if heavy task or sensory work load
 - For vision impaired users

Assessment – Content, Meaning



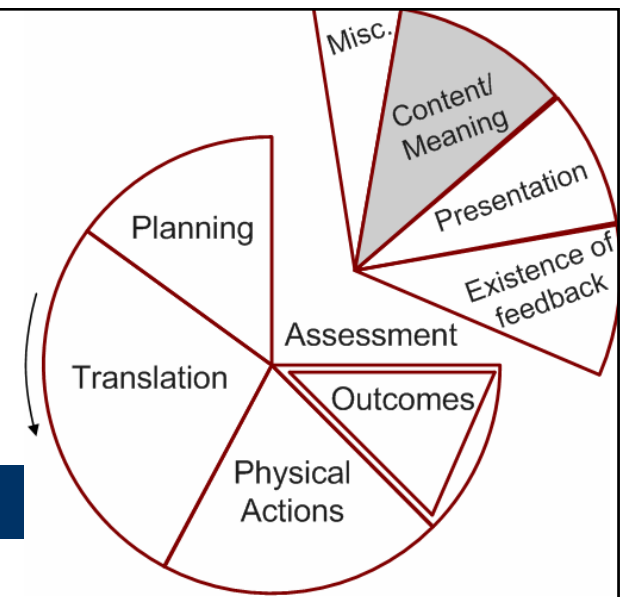
- Support user with effective **content/meaning** in feedback
 - Design for clarity
 - Support clear understanding of outcome (system state change), so users can assess effect of actions
 - Give clear indication of error conditions

Assessment – Content, Meaning



- Design for completeness
 - Provide enough feedback so users can be either confident their command worked or certain about why it didn't
- Help users understand what the real error is
- Provide helpful, informative error messages, not “cute” unhelpful messages

Assessment – Content, Meaning



- Design feedback wording (especially error messages) for positive psychological impact

Psychological

- Make system take blame for errors
- Be positive, to encourage
- Avoid violent, negative, demeaning terms

Try again

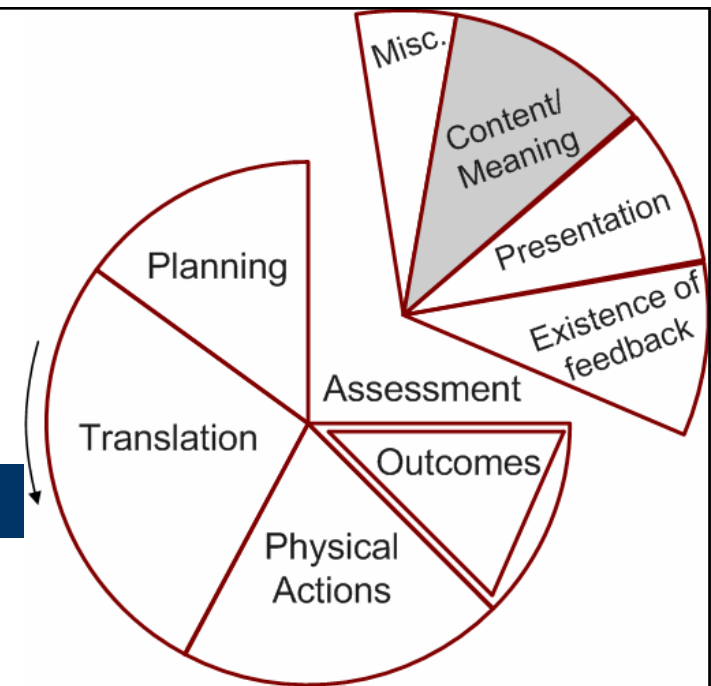
- Avoid use of “illegal”

Illegal

- Employ user-centered wording (language of user and work context) in displays, messages, other feedback

Gobbledygook

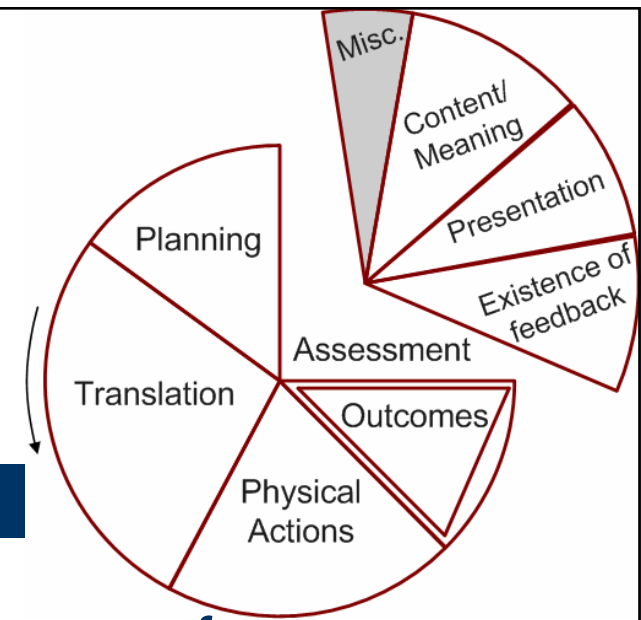
Assessment – Content, Meaning



- Design for consistency
 - Label outcome (e.g., title of new screen or dialogue box) consistently with starting point and action (e.g., button label or menu choice)
- Organize feedback for ease of understanding
 - Provide user control over amount and detail of feedback
 - Give only most important information; more on demand

[Madam - match](#)

Assessment – Information Displays



- Organize information displays for ease of understanding
 - Provide user control over amount and detail of feedback
 - Give only most important information; more on demand
 - Eliminate unnecessary words
 - Group related information
 - Control density of displays; use white space to set off
 - Columns are easier to read than wide rows
 - Use abstraction per Shneiderman’s “mantra”: Overview first; zoom and filter; details on demean

Overall

- Overall issue, not just in one part of Interaction Cycle (e.g., overall wording, style, color)
- Examples of overall style issues
 - Use user-centered (language of user and work context) wording
 - Avoid anthropomorphism-attributing human characteristics to non-human objects
 - Avoid poor attempts at humor
 - Easy to do badly
 - Easily misinterpreted

Suspicious confirmed

Overall

- Examples of overall style issues
 - Avoid irritation in displays (e.g., color, blinking, audio, offensive messages)
 - Use pastels, not bright colors
 - Be aware of color conventions (especially red)
 - Allow user settings, preferences (e.g., sounds levels, blinking, color)
 - Watch out for focusing problem with red and blue

Red and blue

Overall

- Examples of overall style issues
 - Make presentation of text legible
 - Use mixed case for extensive text
 - Avoid too many different fonts, sizes
 - Use legible fonts
 - Make font size large enough for all users
 - Use color other than blue for text
 - Avoid red, except for urgency
 - Use good contrast (color and intensity) with background
 - Accommodate sensory disabilities and limitations (e.g., visually challenged, color blind)

Design Guidelines: Conclusions

- Be cautious using guidelines
 - Need careful thought, interpretation
 - In application, they can conflict and overlap
 - They do not guarantee usability
 - Using guidelines does NOT eliminate need for usability testing
- Design by guidelines, not by politics or personal opinion

Daytime

Jim Foley: “The only correct answer to any UI design question is: It depends”