Theories for evaluating the User Interfaces

Types of theories

- **Descriptive and explanatory**; these theories are helpful in developing consistent terminology for objects and actions, thereby supporting collaboration and training.
- **Predictive theories** these theories enable designers to compare proposed designs for execution time or error rates
- **Motor-task performance Theory-** Based on Motor task Performance like pointing with a mouse.
- **Perceptual Theories** – These theories are successful in predicting reading times for free text, lists, formatted displays, and other visual or auditory tasks.

Information foraging theory

- Web designers have emphasized information-architecture models with navigation as the key to user success.
- Web users can be considered as *foraging* for information, and therefore the effectiveness of the *information scent* of links is important.
- A high-quality link, relative to a specifc task, gives users a good scent (or indication) of what is at the destination. For example, if users are trying to find an executable demonstration of a software package, then a link with the text "download demo" has a good scent.
- The challenge to designers is to understand user tasks well enough to design a large web site such that users will be able to find their way successfully from a home page to the right destination, even if it is three or four clicks away.

Explanatory Theories-Taxonomy

Taxonomy– part of Explanatory or descriptive Theory.

- Imposes order by Classification of a complex set of phenomena into understandable categories;
- Taxonomies facilitate useful comparisons, organize topics for newcomers, guide designers

Examples of taxonomies

1. A Taxonomy for different kinds of input devices (direct versus indirect, linear versus rotary, 1-,2-,3- or higher dimensional) .
2. Taxonomy of tasks (structured versus unstructured, novel versus regular)
3. Taxonomy of personality styles (convergent versus divergent, field-dependent versus independent), technical aptitudes (spatial visualization, reasoning)
4. Taxonomy of user experience levels (novice, knowledgeable, expert),
5. Taxonomy of user-interface styles (menus, form fiIlin, commands).

- Models for forming theories

Levels of analysis Theories

An approach to forming explanatory Theories is to form four levels to separate concepts. The four-levels are conceptual, semantic, syntactic, and lexical model

- 1. The *conceptual level* is the user's "mental model" of the interactive system.
- Two examples of mental models for image creation are paint programs that manipulate pixels and drawing programs that operate on objects.
- Users of paint programs think in terms of sequences of actions on pixels and groups of pixels, while users of drawing programs apply operators to alter and group objects.
- Decisions about mental models affect each of the lower levels.

- 2. The *semantic level* describes the meanings conveyed by the user's input and by the computer's output display.
- For example, deleting an object in a drawing program could be accomplished by undoing a recent action or by invoking a delete-object action.
- Either action should eliminate a single object and leave the rest untouched.

- 3. The *syntactic level* defines how the user actions that convey semantics are assembled into complete sentences that instruct the computer to perform certain tasks. For example, the delete-files action could be invoked by a multiple object selection, followed by a keystroke, followed by a confirmation.
- 4. The *lexical level* deals with device dependencies and with the precise mechanisms by which users specify the syntax (for example, a function key or a mouse double-click within 200 milliseconds).

Stages-of-action models

- Another approach to forming theories is to portray the stages of action that users go through in trying to use interactive products such as information appliances, office tools, and web interfaces.
- Norman (1988) offers seven stages of action, arranged in acyclic pattern, as an explanatory model of human -computer interaction
  - 1. Forming the goal
  - 2. Forming the intention
  - 3. Specifying the action
  - 4. Executing the action
  - 5. Perceiving the system state
  - 6. Interpreting the system state
  - 7. Evaluating the outcome

- The user forms a conceptual intention, reformulates it into the semantics of several commands, constructs the required syntax, and eventually produces the lexical token by the action of moving the mouse to select a point on the screen.
- The stages are placed in the context of *cycles of action* and *evaluation*.
- The seven-stages model leads to identification of the *gulf of execution* (the mismatch between the user's intentions and the allowable actions) and the *gulf of evaluation* (the mismatch between the system's representation and the user's expectations)

- A stages-of-action model helps to describe user exploration of an interface
- . As users try to accomplish their goals, there are four critical points where user failures can occur:
- (1) users can form an inadequategoal,
- (2) users might not find the correct interface object because of an incomprehensible label or icon,
- (3) users may not know how to specify or execute a desired action, and
- (4) users may receive inappropriate or misleading feedback.
- The latter three failures may be prevented by improved design or overcome by time-consuming experience with the interface (Franzke, 1995).
- Refinements of the stages-of-action model have been developed

GOMS and the keystroke-level model

- User actions are decomposed broken into small measurable steps.
- Two important models:

1. the *goals, operators, methods, and selection* (GOMS) model and
2. the *keystroke-level model* .

GOMS Model

- Users begin by formulating goals (edit document) and subgoals (insert word).
- Then users think in terms of operators, vhich are "elementary perceptual, motor, or cognitive acts, whose execution is necessary to change any aspect of the user's mental state or to affect the task environment" (press up-arrow key, move hand to mouse, recall file name, verify that the cursor is at end of file).
- Finally, users achieve their goals by using methods (move cursor to desired location by following a sequence of arrow keys).
- The selection rules are the control structures for choosing between the several methods available for accomplishing a goal (delete by repeated backspace versus delete by selecting a region and pressing the Delete button

- GOMS works nicely for describing steps in decision making while carrying out interaction tasks, such as text editing in a manuscript.
- For example, a user can move a fragment of text by highlighting, cutting, and then pastting.
- A selection rule determines how to highlight the text if it is a single word (by double clicking) or a phrase (by clicking, moving, and SHIFT-clicking).

Keystroke Model

- It is a simplified version of GOMS
- Predicts times for error-free expert performance of tasks by summing up the times for keystroking, pointing, homing, drawing, thinking, and waiting for the system to respond.

Widget-level theories

- Hierarchical decomposition is a useful tool for dealing with complexity,
- Simplifications are made based on the higher level user-interface building tools .
- A model based on the widgets (interface components) is created instead of dealing with atomic level features. .
- Once a scrolling-list widget was tested to determine user performance as a function of the number of items and the size ofnthe window, the performance of future widget users could be predicted automatically.
- The prediction would have to be derived from some declaration of the task frequencies, but the description of the interface would emerge from thenprocess of designing the interface.
- A measure of layout appropriateness (frequently used pairs of widgets should be adjacent, and the left-to-right sequence should be in harmony with the task-sequence description) would also be produced to guide the designer in a possible redesign.
- Estimates of the perceptual and cognitive complexity plus the motor load would be generated automatically

Context of use Theories

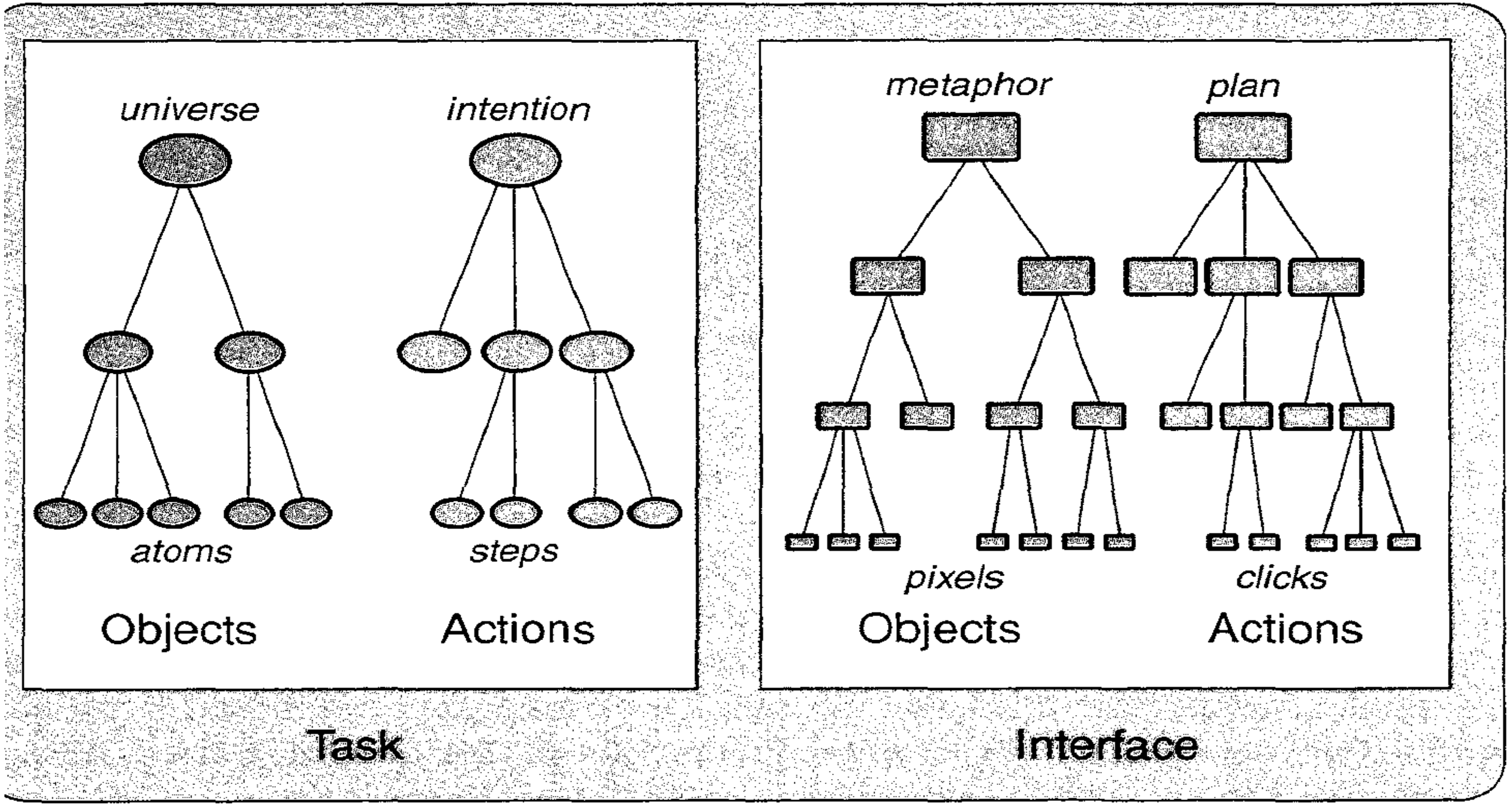
- The physical and social environments are inextricably intertwined with use of information and computing technologies.
- Design cannot be separated from patterns of use.

- The cognitive model of orderly human plans that were executed when needed was insufficient to describe the richer and livelier world of work or personal usage.
- Users' actions were situated in time and place, making user behavior highly responsive to other people and to environmental contingencies.
- If users got stuck in using an interface, they might ask for help, depending on who was around, or consult a manual (if it were available).
- If they were pressed for time, they might risk some shortcuts, but if the work
- was life-critical they would be extra cautious. Rather than having fixed plans,
- users were constantly changing their plans in response to the circumstances.

- Context-of-use theories are especially relevant to mobile devices and ubiquitous computing innovations.
- Such devices are portable or installed in a physical space, and they are often designed specifically to provide place-specific information, such as a city guide on a portable computer or a museum guide that gives information on a nearby painting. A taxonomy of mobile device applications could guide innovators:
- • *Monitor* blood pressure, stock prices, or air quality and give *alerts* when normal ranges are exceeded.
- • *Gather* information from meeting attendees or rescue team members and *spread* the action list or current status to all.
- • *Participate* in a large group activity by voting and *relate* to specific individuals by sending private messages.
- • *Locate* the nearest restaurant or waterfall and *identzfy* the details of the current location.
- • *Capture* information or photos left by others and *share* yours with future visitors.
- These five pairs of actions could be tied to a variety of objects (such as photos, annotations, or documents), suggesting new mobile devices and services.
- They also suggest that one way of thinking about user interfaces is by way of the objects that users encounter and actions that they take

Object-Action Interface Model

- Graphical User Interfaces have replaced command languages,
- intricate syntax has given way to relatively simple direct manipulations applied to visual representations of objects and actions.
- The emphasis is now on the visual display of user-task objects and actions.
- For example, a collection of stock-market portfolios might be represented by leather folders with icons of engraved share certificates; likewise, actions might be represented by trash cans for deletion, or shelf icons to represent destinations for portfolio copying.


universe
intention
atoms
steps
Objects
Actions
Task
metaphor
plan
pixels
clicks
Objects
Actions
Interface

- Doing object-action design starts with understanding the task. That task includes the universe of real-world objects with which users work to accomplish their intentions and the actions that they apply to those objects
- The high-level task objects might be stock-market listings, a photo library, or a personal phone book.
- These objects can be decomposed into information on a single stock, for example, and finally into atomic units, such as a share price.
- Task actions start from high-level intentions that are decomposed into intermediate goals and individual steps.
- Once there is agreement on the task objects and actions and their decomposition, the designer can create the metaphoric representations of the interface objects and actions.
- Interface objects do not have weight or thickness; they are pixels that can be moved or copied in ways that represent real-world task objects with feedback to guide users.
- Finally, the designer must make the interface actions visible to users, so that users can decompose their plans into a series of intermediate actions, such as opening a dialog box, all the way down to a series of detailed keystrokes and clicks