Interaction Styles

- Algebraic languages
- Data files
- Command lines
  - Interaction as a dialogue
- Line editors
  - User maintains mental state model
  - Need extra commands to communicate and alter context
- WYSIWYG
- Modeless interaction
- Menus
  - Recognition over recall
  - Observing menu effects provide feedback
- Pointing devices
- Graphical displays
- Icons and windows

Direct Manipulation

- Objects of interest should be continuously visible
- Operations on objects should involve physical actions (using a pointing device to manipulate the representation)
- Actions that the user makes should be rapid and offer reversible, incremental changes
- The effect of actions should be immediately visible
- The set of commands should be modest, with expert users able to expand it

Style Guidelines

There are issued by OS manufacturers typically with the intention of making all applications that run on the OS appear as part of a family. They provide corporate branding but are not concerned with usability beyond providing a common look which may help users recognize certain types of controls.

Heuristic Evaluation

Usability evaluated by a panel of experts working from a list of usability heuristics. The experts should have different backgrounds and apply the heuristics to the application by theme selves, screen-by-screen and working through at least twice. Finally they report back to the group to create a common report.

Heuristics might include:

- Visibility of system status
- Match between the system and the real world
- User control and freedom (easy way to leave undesired states)
- Consistency and standards (including platform conventions)
- Error prevention
- Recognition over recall
- Flexibility and efficiency of use (e.g. accelerators)
- Aesthetics and minimalism
- Helping users recognize, diagnose and recover from errors
- Help and documentation (ideally not necessary though!)

This technique is popular, simple, cheap and justifiable. However, it provides little opportunity to address deeper system problems and does not provide a systematic means to generate solutions.

User Models

Gestalt Laws Of Perception

- Proximity: elements close together organize into units
- Similarity: objects that look alike tend to be grouped together
- Good continuation: lines are continuous if they do not bend sharply
- Closure: we prefer to see regular shapes, inferring occlusion to do so

Visual Tasks

- Depth perception
- Face recognition (Chernoff faces have been used for information displays!)
- Visual search
  - Pop-out effects: variation in brightness and orientation
  - The time to find one item among similar items is governed by Hick's Law: $T = k \log_2(n + 1)$

Physical Output

Picking something up involves approach (high speed) and homing (low speeds, hands formed into the right shape) phases.

Successive strikes can be made most quickly with alternate hands than with the fingers of the same hand.

Fitt's Law describes the time needed to seek to something with the cursor: for an amplitude (inter-point distance) A and width (along the line of movement) W, $T = k \log_2(\frac{A}{W} + 1)$.

Memory

People can recall between 5 and 9 things at a time: 7 ± 2. These units of memory are called *chunks*.

We can improve learning and memory by providing rich associations: many related connections. This is exploited by UIs that mimic the real world.

Visual working memory and verbal short term memory are independent, and this can be exploited by associating images with items to be remembered (*dual coding*).

Problem Solving

Generalized Problem Solver (GPS) operates in a search space and acts to reduce the difference between the current state and the desired state. The model of decomposing problems into subgoal hierarchies is subject to stack overflows in real humans!

Keystroke Level Model

This model decomposes a task to be evaluated into a number of unit operators:

Letter	The Time It Takes To..	Value
K	Press a key	0.12-1.2s
H	Move your hands to home	0.4s
P	Point with a mouse	Fitt's Law
D	Draw with a mouse	Obsolete
R	Respond to an action	Varies
M	Mentally prepare	1.35s

The rules state that every operation must be preceded by mental preparation, except those operations that form a so-called chunk.

This model is only suitable for making relative judgments about expert users performing a routine task, and is hampered by the ambiguous mental preparation rules. However, it does have the advantage of being quantitative.

GOMS

GOMS (Goals, Operators, Methods and Selection) was an attempt to extend KLM. The operators (action taken by the user) corresponded to the KLM operators and users would also have methods (sequences of operations) for accomplishing some goal. The model attempted to provide quantitative models of goal hierarchy decomposition that included selection of operators and methods, the use of working memory and the learning process!

Its problems included the facts that it had no representation of dual coding or the Gestalt laws of perception and that it couldn't deal with the flexibility of users and the fact that 35% of time in an application is spent in error handling, which it didn't attempt to account for.

Mental Models

These describe the structure of mental representations that people use for everyday reasoning and problem solving. An example is the application of the model of flowing water to electricity supply that leads to people ensuring that all plugs have a device attached to them. Users make inferences by a process of mental simulation of these models, which we can attempt to duplicate in HCI research.

User-Oriented Design

Prototyping

By showing a working version to clients they may begin to form a mental model of system behavior. This can be done by an iteratively refined rapid prototype. An alternative is deep prototyping, which fully implements one aspect of the system before developing the remainder of the system. The prototype may even take the form of a low fidelity prototype such as paper using the so-called Wizard of Oz technique.

Empirical Techniques

This are based on observation of users, using statistical methods such as the t-test to determine whether particular hypotheses are supported by the evidence or not.

Researchers try to obtain environmental validity for their results, i.e. they are obtained in an environment which is as close as possible to the one in which their software will be used.

Surveys can be used to solicit user opinion, with a mix of open and closed questions (requiring a Boolean answer or one on a Likert scale). Open questions require a methodological coding technique to survey the content of responses across the population.

Questionnaires are surveys administered in writing and are typically used to obtain information from larger audiences.

Think aloud studies try to get subjects to talk about what they are thinking by means of a verbal protocol. It can be difficult to get this to happen reliably however.

Subjective reports or those with untested usability hypotheses are *bad*!

Cognitive Walkthrough

This technique addresses usability of a system for inexpert users that may be attempting a novel task. In order to carry this out, a behavior model is needed. This describes how a notional user sets a goal, searches for actions, selects an action and evaluates the feedback of the system.

The evaluation further requires a description of the type of users of the system and their relevant knowledge, a description of the tasks that will be used in the evaluation and a list of the correct actions that are required to perform the task as anticipated by the designer. To perform the evaluation, the interface designer and a group of peers (including a scribe and facilitator) simulate the notional user performing the representative task. At each step the interface is examined and the group tell a story consistent with the user model about why the user chooses the particular action the designer had in mind. This will need to take into account the users goal, the control accessibility and quality of its match with the goal, as well as the feedback the system provides.

This is widely believed to be realistic, but assumes that the evaluators are knowledgeable designers who are capable of applying relevant theories of cognitive psychology to the task. As it is more structured than heuristic evaluation it is less likely to suffer from subjectivity.

Task-Oriented Analysis

You can mitigate the disconnect between system commissioners and users by conducting a series of structured interviews with a range of system users. They must encourage cooperation from threatened or anxious users. The structure comes from a common question set which can be used to later aggregate responses.

Observational studies are conducted by means of a video study or user diary. This class of techniques precludes interaction between researcher and subject.

In contrast, ethnographic field studies have the researcher engaging with the user, possibly over a substantial period of time and multiple contexts. A full record is made of activities and artifacts. This can be an aid to requirements capture as the problem may be understood from a totally new angle.

A simpler technique may involve constructing user personas, which may be derived from the above channels or market research data.

Field tests of finished products have been reported as successful in the past, for example with Quicken.

Use Case Design

In this user-centered technique the system is described from the point of view of abstract actors which represent user roles. They engage in use cases, which are narratives of some specific interaction that the actor conducts with the system.

Cognitive Dimensions Of Notations

The dimensions are a vocabulary for design discussion, based on the observation that perfection is impossible and trade-offs necessary. It deal with system that provide a notation, which is a means of talking about an object such as a musical score or language syntax, and an environment which allows the notation to be viewed and manipulated, such as a tablet interface for drawing musical notes or a keyboard interface for entering them. Some example dimensions are:

- Premature commitment
- Hidden dependencies (important links are not visible, e.g. spreadsheet formulae, goto statements)
- Secondary notations (extra information in means other than formal syntax, possibly extending to helper devices: what you would write on the form if you could print it)
- Viscosity (resistance to change, e.g. changing spelling from the UK to US)
- Closeness of mapping
- Consistency
- Diffuseness (verbosity of the notation)
- Error-proneness
- Progressive evaluation
- Abstraction

Subdevices may be identified which make use of specialized notations to help with the job. They are *helper* devices, such as Post-It notes, and *redefinition* devices that change the main notation is some way, such as defining a keyboard shortcut.

Users interact with notations in some limited set of ways:

- Search (navigating the notational structure, without the notation changing)
- Incrementation (adding to an notation without changing the structure)
- Modification (changing the structure, possibly without adding content e.g. repurposing a spreadsheet)
- Transcription (moving between notations)
- Exploratory design (combines incrementation and modification with the characteristic that the desired final state is not known in advance)

Evaluation consists of identifying the notation, its medium, and the environment, and then the notations for any identified subdevices. Each notation is discussed in terms of the dimensions, identifying where the system characteristics in that dimension are inappropriate to the user activity. Design maneuvers can then be discussed to manipulate that dimension, potentially introducing trade-offs.

The dimensions are useful in situations where there is no correct or preferred action sequence, as in a programming task. They offer the potential to provide more information than other evaluation methods regarding the relative needs of users.