# Human-Computer Interaction

Class Code: BSCS-F2015A

*Tariq Zaman Ph.D.*
CECOS University of I.T. & Emerging Sciences
F-5, Phase- VI, Hayatabad, Peshawar

CECOS
UNIVERSITY

# Design Rules

Designing for maximum usability
– the goal of interaction design

- Principles of usability
  – general understanding
- Standards and guidelines
  – direction for design
- Design patterns
  – capture and reuse design knowledge

# Design Rules

- We require *design rules*, which are rules a designer **can follow in order to increase the usability** of the eventual software product.

# Design Rules

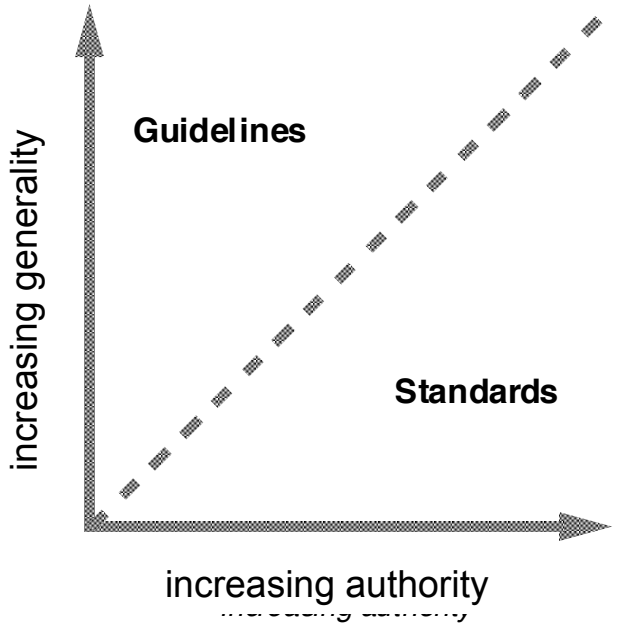
- We can classify these rules along two dimensions, based on the rule's
  - **authority and**
  - **generality**
- By **authority**, we mean an indication of whether or not the rule must be followed in design or whether it is only **suggested**
- By **generality**, we mean whether the rule can be applied to many design situations or whether it is focused on a more limited **application** situation.

# Design Rules

- Rules also vary in their level of abstraction, with some abstracting away from the detail of the design solution and others being quite specific.
- It is also important to determine the origins of a design rule.

# Types of design rules

- principles
  - abstract design rules
  - low authority
  - high generality
- standards
  - specific design rules
  - high authority
  - limited application
- guidelines
  - lower authority
  - more general application




The design rules are used to apply the theory in practice.

# Principles, Standards and Guidelines

- There exists distinction between principles, standards and guidelines.
  - Principles are derived from knowledge of the **psychological**, **computational** and **sociological** aspects of the problem domains and are largely independent of the technology.
  - They depend to a much greater extent on a **deeper understanding of the human element in the interaction.**

# Principles, Standards and Guidelines

- Guidelines are less abstract and often more **technology** oriented.
  - As they are also general, it is important for a designer to know what theoretical evidence there is to support them.
- A designer **will have less of a need to know the underlying theory** for applying a standard. However, since standards carry a **much higher level of authority**, it is more important that the theory underlying them be correct or sound.

# Principles to support usability

Learnability

the ease with which new users can begin effective interaction and achieve maximal performance

Flexibility

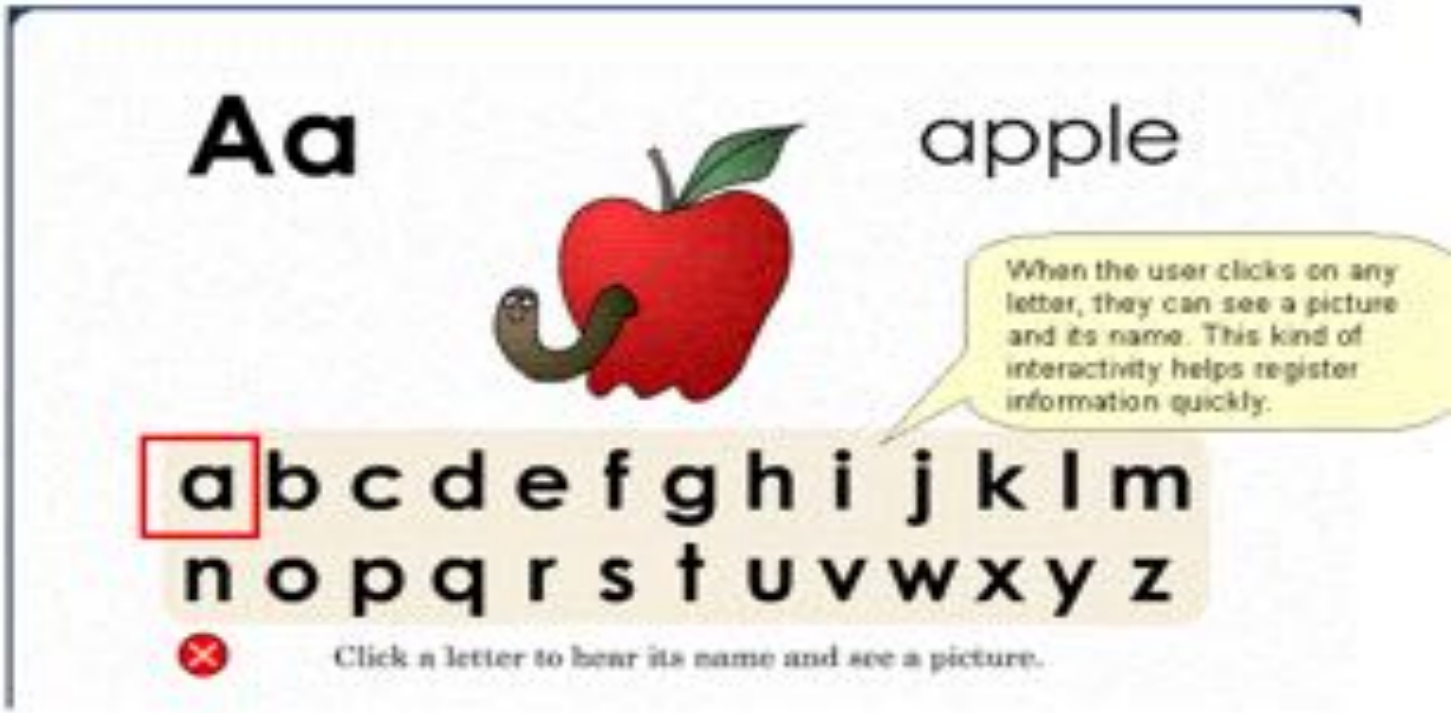
the multiplicity of ways the user and system exchange information

Robustness

the level of support provided the user in determining successful achievement and assessment of goal-directed behaviour

# Principles to support usability

- **Learnability:** the ease with which new users can begin effective interaction and achieve maximal performance



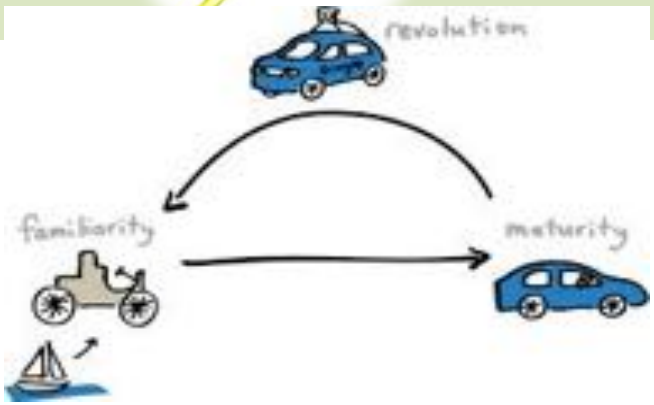
# Principles to support usability

- **Flexibility:** the multiplicity of ways the user and system exchange information


# Principles to support usability

- **Robustness:** the level of support provided the user in determining successful achievement and assessment of goal-directed behavior.

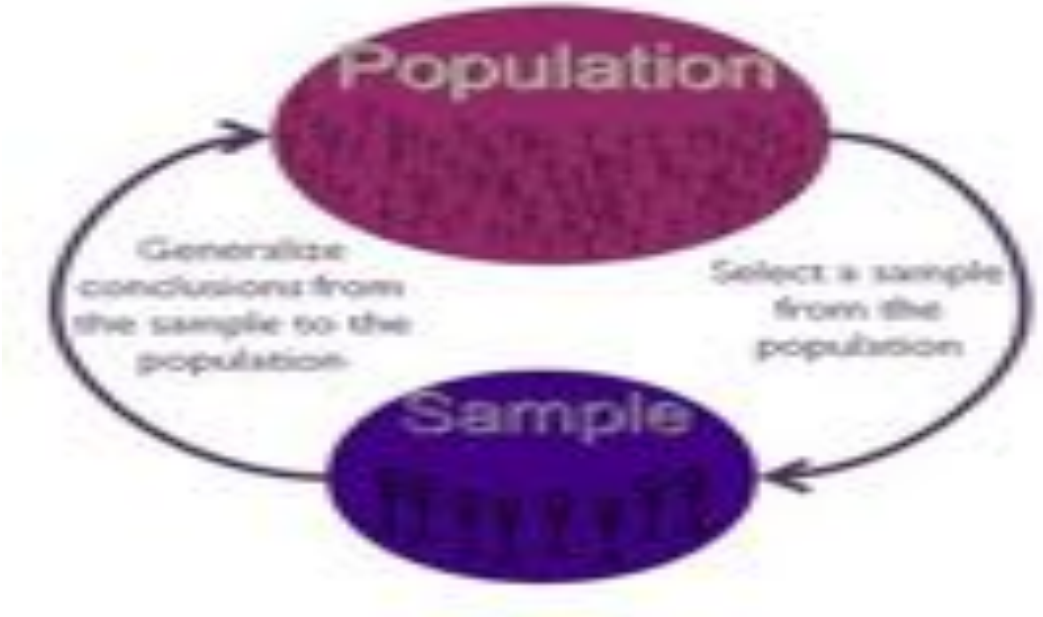

# Principles of learnability




- **Familiarity**
  - how prior knowledge applies to new system
  - guess-ability; metaphors; affordance


# Principles of learn-ability

- **Generalizability**
  - extending specific interaction knowledge to new situations
  - eg. cut/paste/copy across apps



# Principles of Learn-ability

- **Consistency**
  - likeness in input/output behavior arising from similar situations or task objectives

# Predictability

- Predictability of an interactive system means that the user's knowledge of the interaction history is sufficient to determine the result of his future interaction with it.
- There are many degrees to which predictability can be satisfied.



# Flexibility

- Flexibility refers to the multiplicity of ways in which the end-user and the system exchange information.

# Principles of Flexibility

Dialogue initiative

- freedom from system imposed constraints on input dialogue
  - user should be able to abandon, suspend or resume tasks at any point
- system vs. user pre-emptiveness

Multithreading

- ability of system to support user interaction for more than one task at a time
- concurrent vs. interleaving; multimodality

Task migratability

- passing responsibility for task execution between user and system
  - People get bored doing routine tasks and stop concentrating

# Principles of flexibility (ctd)

Substitutivity

- allowing equivalent values of input and output to be substituted for each other
- representation multiplicity; equal opportunity
  - don't force users to refer to objects by name if they can point to them

Customizability

- modifiability of the user interface by user (adaptability) or system (adaptivity)

# Robustness

- In a work or task domain, a user is engaged with a computer in order to achieve some set of goals.
- The robustness of that interaction covers features that support the successful achievement and assessment of the goals.


# Principles of robustness

Observability

- ability of user to evaluate the internal state of the system from its perceivable representation
- browsability; defaults; reachability; persistence; operation visibility

Recoverability

- ability of user to take corrective action once an error has been recognized
- reachability; forward/backward recovery; commensurate effort

# Principles of robustness (ctd)

Responsiveness

- how the user perceives the rate of communication with the system
- Stability
- *% completed*

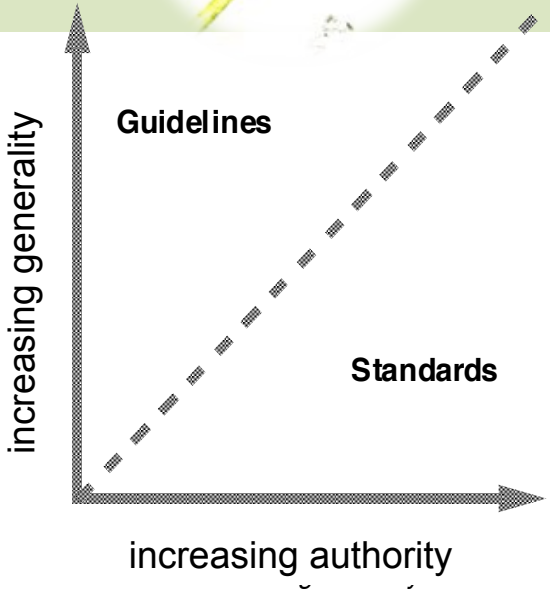
Task conformance

- degree to which system services support all of the user's tasks
- task completeness; task adequacy
- *Commands and shortcuts*

# Using Design Rules

Design rules

- suggest how to increase usability
- differ in generality and authority



# Standards


- set by **national or international bodies** to ensure compliance by a large community of designers standards require sound underlying theory and slowly changing technology.
- hardware standards **more common** than software high authority and low level of detail.
- ISO 9241 defines usability as effectiveness, efficiency and satisfaction with which users accomplish tasks.

# Underlying Theory –Hardware Standards

- Standards for hardware are based on an understanding of physiology or ergonomics/ human factors, the results of which are relatively well known, fixed and readily adaptable to design of the hardware.

# Underlying theory –software Standards

- Software standards are based on theories from psychology or cognitive science, which are less well formed, still evolving and not very easy to interpret in the language of software design.

# Change

- Hardware is more difficult and expensive to change than software, which is usually designed to be very flexible.
- Consequently, requirements changes for hardware do not occur as frequently as for software.
  - Since standards are also relatively stable, they are more suitable for hardware than software.

# ISO 9241

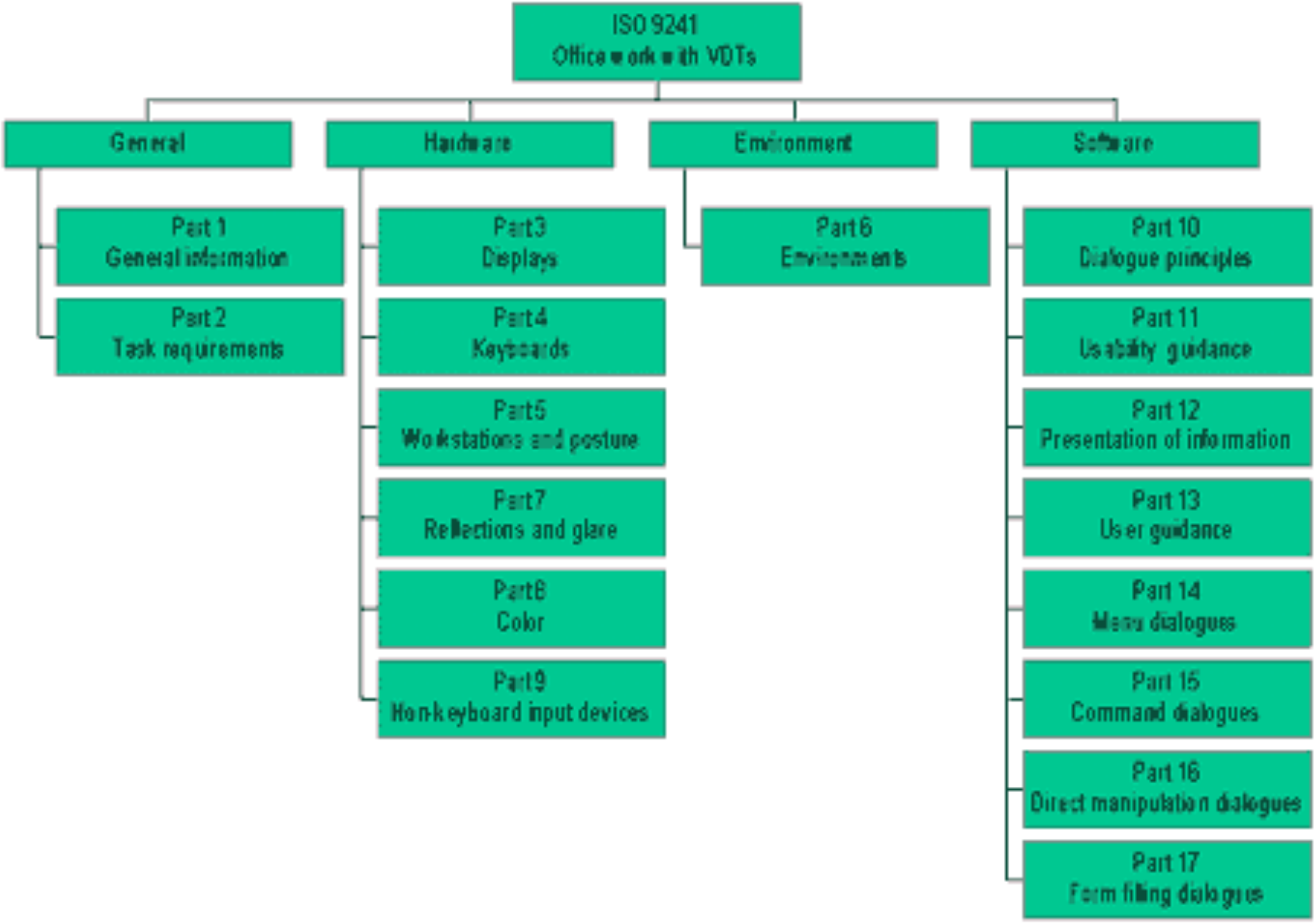
- One component of the ISO standard 9241, pertaining to usability specification, applies equally to both hardware and software design.

- **Usability:** The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.

# ISO 9241

- ***Ergonomics of Human System Interaction***, adopts traditional usability categories:
- **Effectiveness**
  - can you achieve what you want to?
- **Efficiency**
  - can you do it without wasting effort?
- **Satisfaction**
  - do you enjoy the process?

Organization of ISO 9241 by subject



# Example Metrics from ISO 9241

| Usability objective | Effectiveness measures | Efficiency measures | Satisfaction measures |
|---|---|---|---|
| Suitability for the task | Percentage of goals achieved | Time to complete a task | Rating scale for satisfaction |
| Appropriate for trained users | Number of power features used | Efficiency relative to expert user | Rating scale for ease of learning |
| Learnability | Percentage of functions learned | Time to learn criterion | Rating scale for ease of learning |
| Error tolerance | Percentage of errors corrected successfully | Time spent on correcting errors | Rating scale for error handling |

# Strength of Standards

- Lies in ability to force large communities to abide (so-called authority)
- Most standards are suggestions
- Some practices become de facto standards before formalization

# Guidelines

- More suggestive and general
- Many textbooks and reports full of guidelines
- Abstract guidelines (principles) applicable during early life cycle activities
- Detailed guidelines (style guides) applicable during later life cycle activities
- Understanding justification for guidelines aids in resolving conflicts

# Guidelines

- Creating guidelines documents causes debate!
- Criticisms that guidelines are:
  - Too specific
  - Incomplete
  - Hard to apply
  - Sometimes wrong
- But guidelines more often better than nothing, lead to steady improvements

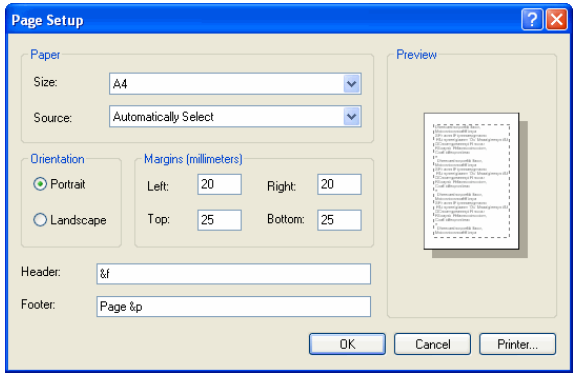
- National Cancer Institute government webpage guidelines for interface navigation:
  - Standardize task sequences
  - Ensure that embedded links are descriptive
  - Use unique and descriptive headings
  - Use radio buttons for mutually exclusive choices
  - Do not display unsolicited windows or graphics
- Note: there are do's and don'ts

# Guidelines for Interactive Design

- Mayhew
  - more recent, comprehensive, general guidelines in a catalog

# Guidelines

- Dialog styles
  - question and answer, form-filling, menu selection, function keys, command language, query, natural language, direct manipulation
  - Most guidelines applicable for implementation of any one of dialog styles in isolation
  - Must also consider mixing of styles in an application (Mayhew provides guidelines on this)



# Specific Guidelines

- Apple's HCI Guidelines: Human's Interface guidelines

# HCI Design Patterns

- An approach to **reusing knowledge about successful design solutions**
- Originated in architecture: Alexander
- A pattern is **an invariant solution to a recurrent problem within a specific context.**
- Examples
  - Light on Two Sides of Every Room (architecture)
  - Go back to a safe place (HCI)
- Patterns do not exist in isolation but are linked to other patterns in *languages* which enable complete designs to be generated

# HCI Design Patterns (cont.)

- Characteristics of patterns
  - capture design practice not theory
  - capture the essential common properties of good examples of design
  - represent design knowledge at varying levels: social, organisational, conceptual, detailed
  - embody values and can express what is humane in interface design
  - are intuitive and readable and can therefore be used for communication between all stakeholders
  - a pattern language should be generative and assist in the development of complete designs.

# Summary: Design Rules

Designing for maximum usability
– the goal of usability and design

- Principles of usability
  – general understanding
- Standards and guidelines
  – direction for design
- Design patterns
  – capture and reuse design knowledge

# Shneiderman's 8 Golden Rules

1. *Strive for consistency*
2. *Enable frequent users to use shortcuts*
3. *Offer informative feedback*
4. *Design dialogs to yield closure*
5. *Offer error prevention and simple error handling*
6. *Permit easy reversal of actions*
7. *Support internal locus of control*
8. *Reduce short-term memory load*

# Norman's 7 Principles

1. *Use both knowledge in the world and knowledge in the head.*
2. *Simplify the structure of tasks.*
3. *Make things visible: bridge the gulfs of Execution and Evaluation.*
4. *Get the mappings right.*
5. *Exploit the power of constraints, both natural and artificial.*
6. *Design for error.*
7. *When all else fails, standardize.*