The Design of Everyday Things  
*by Donald Norman*

**Norman, D. A. (1988). The Design of Everyday Things.**  New York: Doubleday.   ISBN: 0-385-26774-6.    Call Number: TS171.4 .N67 1990.

A popular book that will motivate the importance of human factors in the design of everything we use. This reading is also included as an introduction to concepts such as "affordances" and "knowledge in the world" versus "knowledge in the head"  [DS]

Jump To:

* [About the Author / Reason for Writing the Book](http://www.sharritt.com/CISHCIExam/norman.html#about)
* [Chapter 1:  The Psychopathology of Everyday Things](http://www.sharritt.com/CISHCIExam/norman.html#1)
* [Chapter 2:  The Psychology of Everyday Actions](http://www.sharritt.com/CISHCIExam/norman.html#2)
* [Chapter 3:  Knowledge in the Head and in the World](http://www.sharritt.com/CISHCIExam/norman.html#3)
* [Chapter 4:  Knowing What to Do](http://www.sharritt.com/CISHCIExam/norman.html#4)
* [Chapter 5:  To Err is Human](http://www.sharritt.com/CISHCIExam/norman.html#5)
* [Chapter 6:  The Design Challenge](http://www.sharritt.com/CISHCIExam/norman.html#6)
* [Chapter 7:  User-Centered Design](http://www.sharritt.com/CISHCIExam/norman.html#7)

About the author:

Donald Norman wrote this book and “The Invisible Computer”.  His primary area of research is human-centered design.  He is a professor at UCSD in Cognitive Science and Psychology, and has held high positions working at Apple and HP.

Norman wants to fully utilize the potential of technology and the computer by supporting human tasks first, while attempting to make the supporting technology transparent to users by making them easy to use, easy to learn and easy to understand.

From <http://cogsci.ucsd.edu/~norman/> and [www.jnd.org](http://www.jnd.org/)

Reason for writing the book:

Donald Norman wrote the book for many reasons.  The initial thought was the frustration that he encountered with ‘everyday things’.  His inability to operate simple things was frustrating, and after feeling flustered and confused at his inability to operate things, he realized that much of the problem was due to poorly designed interface.  This made him realize that people shouldn’t feel guilty or stupid for their inability to operate devices.  The fault was with the unintuitive interface- that shifts the problem space to the design of a good interface.

**(**[**top**](http://www.sharritt.com/CISHCIExam/norman.html#top)**)**

***Summary of the Book:***

Chapter 1:  The Psychopathology of Everyday Things

Users shouldn’t need an engineering degree to figure out what a device does

He uses the example of aesthetically pleasing glass doors- how we can get trapped or not able to pass through them because there are no clues on how to use them

*VISIBILITY - o*ne of the most important aspects of design – interface must have visible features, inferring the right messages to us

*Natural Signals –*the ‘natural’ or common understanding of objects and their perceived use

*Natural Design –*design that takes advantage of ‘natural signals’

*MAPPINGS –*the link between what *you want to do* and what *is perceived possible.*It is the relationship between moving a control, and the results in the real world.

*Natural Mapping –*takes advantage of physical analogies and cultural standards for immediate understanding

*AFFORDANCES –*the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used   
(e.g. a chair *affords* sitting; glass *affords* seeing through, breaking; wood *affords* solidity, opacity, support, carving)

Affordances provide us clues on how to operate a device

*CONSTRAINTS –*limits to the perceived operation of a device (e.g. a small hole vs. a large hole- we might be able to use only one finger in the small hole, while we might be able to use multiple fingers in a large hole)

*CONCEPTUAL MODEL –*our mental simulation of a device’s operation (mental model?) These can be based on MAPPINGS, AFFORDANCES and CONSTRAINTS.

*MENTAL MODEL –*models people have of themselves, others, their environment, and the things they interact with (CONCEPTUAL MODELS are part of this)

The mental model of a device is formed by interpreting its perceived actions and its visible structure.

*System Image* – the visible part of the device being used.  If incomplete / contradictory, the user cannot easily use the device.

*Feedback –*sending information back to the user about what action has actually been done and what result was accomplished

*Two principles of designing for people:*

1. *good conceptual model*
2. *make things visible*

Norman’s Conclusion:  Design is not an easy task.  Technology is a paradox because it is supposed to make our lives easier when it often makes it more difficult.  However, this is not an excuse for poor design.

**(**[**top**](http://www.sharritt.com/CISHCIExam/norman.html#top)**)**

Chapter 2:  The Psychology of Everyday Actions

People feel bad, sorry, frustrated, stupid for not knowing how to operate mechanical things, especially if the task appears to be trivial

The world, and everyday things, are filled with misconceptions

*Aristotle's naive physics -*our 'naive' way of explaining the phenomenon we witness in everyday life - often very practical but incorrect.  People often have naïve, incorrect explanations for real world phenomenon (cranking the thermostat all the way will make us reach a desired temperature faster)

Coincidence can set our ‘causal’ wheels rolling.  What matters is that we ‘perceive’ causality, and whether or not that causality exists, we think it is there.  Often we perceive causality that isn’t there and often ignore the real cause.  This can create a problem / crisis later because we have a bad explanation of what is happening (3 Mile Island)

*Spiral of silence / conspiracy of silence* - not reporting errors / misconceptions that you think are your fault (you're in the minority and don't want to be singled out).  Even though this may not be true- the majority might be having the same problem, and we need to find out.

*Learned helplessness -* after failing to do a talk multiple times, people often decide that they cannot do the task (they are helpless)

*Taught helplessness -* perceived difficulty in one task generalizes to the whole, so that we feel (self-blame) that we cannot do tasks (such as in mathematics, where each successive task requires complete understanding of previous tasks).  A sort of self-fulfilling prophecy that we are unable to accomplish a task due to previous difficulty / failure.

***7 Stages of Action:  An Approximate Model***

*(Execution)*           Goals           *(Evaluation)*                               /         \  
          Intention to Act               Evaluation of interpretations  
                  v                               ^  
        Sequence of Actions              Interpreting the perception  
                  v                               ^  
Execution of the action sequence         Perceiving the state of the world  
                  v                               ^  
                     \        (THE WORLD)      /

*7 Stages of Action:  1 for goals, 3 for execution, 3 for evaluation:*

* Forming the Goal
* Forming the intention
* Specifying an action
* Executing the action
* Perceiving the state of the world
* Interpreting the state of the world
* Evaluating the outcome

*THE GULF OF EXECUTION:*does the system provide actions that correspond to the intentions of the user?

*THE GULF OF EVALUATION:*  does the system provide a physical representation that can be directly perceived and that is directly interpretable in terms of the intentions and expectations of the user?

Each of the seven stages are good for checking that the gulfs of execution and evaluation are bridged.  How easily can one:

*Determine The Function of the Device?*

*Tell What Actions Are Possible?                                 Tell if System is in Desired State?*

*Determine Mapping from Intention                            Determine Mapping from System  
to Physical Movement?                                            State to Interpretation?*

*Perform the Action?                                                Tell What State the System is In?*

These questions boil down to the principles of design from Chapter 1:  ***Visibility, A good conceptual model, Good mappings, and Feedback***

**A Great Explanation of Norman's Gulfs of Execution and Evaluation:**<http://www.it.bton.ac.uk/staff/rng/teaching/notes/NormanGulfs.html>

**(**[**top**](http://www.sharritt.com/CISHCIExam/norman.html#top)**)**

Chapter 3:  Knowledge in the Head and in the World (Memory)

**We can have precise behavior on how to do a task without precise knowledge of the task due to 4 reasons:**

1. *Information is in the  world*:  much of the information required to do the task can reside in the world.  Behavior results from combining information in the head with information in the world.
2. *Great precision is not required:*  precision, accuracy and completeness of knowledge are seldom required.  Perfect behavior will happen if there is sufficient knowledge to distinguish the correct choice from the others.
3. *Natural constraints are present.*  The world restricts the allowed behavior.  The physical properties of objects constrain possible operations (ways we can use / manipulate objects).  Each object has a set of physical features that limit its relationships to other objects, the operations that can be done on it, etc.
4. *Cultural constraints are present.*  Society has evolved numerous artificial conventions that govern acceptable social behavior.  These cultural conventions must be learned, but once learned apply to a wide variety of circumstances.

These four reasons reduce the number of alternatives and reduce the amount of information required to be stored in memory to successfully complete the task.

**Memory** is *knowledge in the head*

* Often grouped into *short term memory*and *long term memory*
* Three important categories of memory:
  + Memory for arbitrary things (without meaning / relationships)
  + Memory for meaningful relationships (with something else)
  + Memory through explanation (some explanatory mechanism)
* Typically requires learning, is efficient, and not easily retrieved

**Memory** is also *knowledge in the world*

* Reminding (signal and a message)
* Natural Mappings (arrangement, like stove controls example)
* Typically easily retrieved whenever visible / audible, no learning required, but slowed up by the need to interpret the external information

There are three aspects to mental models (types of conceptual models?):

* the *design model*(the conceptualization the designer had in mind)
* the *user’s model*(what the user develops to explain the operation of the system)
* and the *system image* (the system’s appearance, operation, way it responds, manuals / instructions included with it)

Ideally, the design model and user model are the same.  The designer must ensure that the system image is consistent with and operates according to the proper conceptual model.

**(**[**top**](http://www.sharritt.com/CISHCIExam/norman.html#top)**)**

Chapter 4:  Knowing What to Do

We mess up when there is more than one possibility / option of things to do

Building the Lego motorcycle:  semantic and cultural constraints, as well as the shape (clues) of the pieces allow us to figure out easily how the pieces are assembled together

***Constraints:***

*Physical constraints –*physical limitations, based on shape, size, etc.

*Semantic constraints –*limitations based on the meaning of the situation (Lego motorcycle:  rider must face forward… windshield goes in front of face, etc.)

*Cultural constraints –*limitations based on accepted cultural conventions.  (Lego motorcycle:  signs are meant to be read, thus the ‘police’ sign should be right side up.  The red light goes on the rear, because red is culturally defined to mean ‘stop’, etc.)

*Logical constraints –*logically induced limitations (Lego motorcycle:  all pieces should be used, with no gaps, etc.)

**Constraints are important in suggesting what we should do- so they should not be deceiving.  An object should suggest (afford) what it does (only one predictable outcome- GOOD MAPPING).**

For example:  an array of identical looking switches is a bad design

While the above mainly focuses on constraints and mappings, we must remember to use good **visibility and feedback.**Crucial parts must be visible (doors must have door handles) and we need feedback to verify we completed the task successfully (a good display, showing what just happened)

**(**[**top**](http://www.sharritt.com/CISHCIExam/norman.html#top)**)**

Chapter 5:  To Err Is Human

Language has built in functions to allow us to correct ourselves when we stumble, mess up, etc.  Artificial devices often do not- a mistake can cause chaos.

Slips are the most common error:  when we intend to do one thing and accidentally do another (automatic behavior problem)

Types of Slips:

* Capture Errors (two action sequences have common initial stages - an alternative action 'captures' your attention)
* Description Errors (two objects are physically alike enough to mess up - like throwing clothes in the toilet)
* Data-Driven Errors (recalling the wrong piece of data – confusing two numbers)
* Associative Activation Errors (event activates a similar but wrong response)
* Loss-of-Activation Errors (forgetting to do something or part of the act)
* Mode Errors (when devices have multiple modes and our actions are for the wrong mode)

Well designed things should allow us to detect slips by using feedback (a clear discrepancy between actual and intended result).   For example, in computers, when destroying a file, it is good to ask for confirmation to verify the user wants to do an irrevocable action)

Human cognition is extremely complex and difficult to understand, but better understanding of this will allow us to design better systems with less human error.

* Conscious vs. subconscious
* Deep / narrow vs. shallow / wide tasks
  + if shallow, width is acceptable (choosing a flavor of ice cream:  many choices, but only one decision)
  + if narrow, depth is acceptable (following a recipe:  few decisions, many steps)

Design should allow for human error:

* 1. Understand causes of error and try to minimize them
  2. Make it possible to undo actions
  3. Make it easier to discover when errors occur and make them easy to fix
  4. Think of tasks as imperfect approximations of what the user wants to do

Forcing Functions:  If need be, use lockout devices (force a sequence of actions so that the user can’t enter a dangerous place)

A good design philosophy:  p. 140 - summarizes principles discussed thus far

**(**[**top**](http://www.sharritt.com/CISHCIExam/norman.html#top)**)**

Chapter 6:  The Design Challenge

Often good design is an evolving process:  a design is tested, problems are found, design is modified.  Process repeats and continues until resources run out.

Design is a constant battle between **usability and aesthetics**.  Problems occur when one dominates over the other too much.

Designers are not end-users, often clients are not either.

Often we have selective attention:  we focus too much one thing and reduce attention to other vital things.  (such as sticking a knife into a toaster to get the burning bread out.)  Designers have a hard time conceiving of all of the possible ways that people will use things!

Often designers mess with convention when designing things (faucet examples, p. 166)

**Two deadly temptations for the designer:**

1. *Creeping featurism* – keep adding useless features until it’s too difficult to use
2. *The Worshipping of False Images -* make it complex because it looks cool

Often there is no perfect answer to a problem.  We MUST consider the tradeoffs of our design, and weigh the options to come up with the best solution.

“THE INVISIBLE COMPUTER OF THE FUTURE” is mentioned at the end of the chapter… where we do tasks and the computer is transparent (we are not using the computer, we are completing a task)

**(**[**top**](http://www.sharritt.com/CISHCIExam/norman.html#top)**)**

Chapter 7:  User-Centered Design

The point of the book was to advocate a **user-centered design** which is a philosophy that things should be designed with the needs and interests of the user in mind, making products that are easy to use and understand.

***Design Should:***

1. make it easy to determine what actions are possible at any moment
2. make things visible, including the conceptual model of the system, the alternative actions, and the results of actions.
3. Make it easy to evaluate the current state of the system
4. Follow natural mappings between intentions and the required actions; between actions and the resulting effect; between the information that is visible and the interpretation of the system state

Basically we should be able to (1) figure out what to do (2) tell what is going on

***Principles for making difficult tasks simple ones:***

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

***There are three aspects to mental models:***

* the *design model*(the conceptualization the designer had in mind)
* the *user’s model*(what the user develops to explain the operation of the system)
* and the *system image* (the system’s appearance, operation, way it responds, manuals / instructions included with it)

Ideally, the design model and user model are the same.  The designer must ensure that the system image is consistent with and operates according to the proper conceptual model.

***Ways to simplify the structure of tasks:***

* keep the task much the same, but provide mental aids (simple mental aids provide cues about what we should do)
* use technology to make visible what would otherwise be invisible, thus improving feedback and the ability to keep control (and hide stuff that is irrelevant to completing the task)
* Automate, but keep the task much the same (remove unnecessary steps of a task)
* Change the nature of the task (use technology to simplify something)

But remember NOT TO TAKE AWAY CONTROL FROM THE USER!

***Bridge the gulfs of Execution and Evaluation:***

* Make things visible so users know what actions are possible
* Make things visible so people can see the results of their actions
* The system should have actions that match the users’ intentions

***Design for Error:***

* Make it so mistakes aren’t too critical, undoable, etc.

***Make things difficult?***

* Sometimes a difficult design is good- it forces us to deliberately focus on what we’re doing (focus on it)
* Good for dangerous equipment, operations, secret doors, etc.

***Make things easy to use?***

* To make something easy to use, match the number of controls to the number of functions and organize the panels according to function.
* To make something LOOK easy, minimize the number of controls.

*Remember, tools not only control WHAT we do, but HOW we do it and the way we VIEW ourselves, society and the world!  Our design can change a task, a society, and the world.*

*The world of the future can be looked forward to with pleasure, contemplation and dread.  How will we be able to handle more information that is more complex and control it with more ease?  The answer:  this book- the design of everyday things.  We must fight for and reward good design, and do the opposite for bad design.*

**(**[**top**](http://www.sharritt.com/CISHCIExam/norman.html#top)**)**

|  |  |
| --- | --- |
| Jump To: | [1](http://www.sharritt.com/CISHCIExam/norman.html#1) [2](http://www.sharritt.com/CISHCIExam/norman.html#2) [3](http://www.sharritt.com/CISHCIExam/norman.html#3) [4](http://www.sharritt.com/CISHCIExam/norman.html#4) [5](http://www.sharritt.com/CISHCIExam/norman.html#5) [6](http://www.sharritt.com/CISHCIExam/norman.html#6) [7](http://www.sharritt.com/CISHCIExam/norman.html#7) |

[About the Author / Reason for Writing the Book](http://www.sharritt.com/CISHCIExam/norman.html#about)  
- [Chapter 1:  The Psychopathology of Everyday Things](http://www.sharritt.com/CISHCIExam/norman.html#1)  
- [Chapter 2:  The Psychology of Everyday Actions](http://www.sharritt.com/CISHCIExam/norman.html#2)  
- [Chapter 3:  Knowledge in the Head and in the World](http://www.sharritt.com/CISHCIExam/norman.html#3)  
- [Chapter 4:  Knowing What to Do](http://www.sharritt.com/CISHCIExam/norman.html#4)  
- [Chapter 5:  To Err is Human](http://www.sharritt.com/CISHCIExam/norman.html#5)  
- [Chapter 6:  The Design Challenge](http://www.sharritt.com/CISHCIExam/norman.html#6)  
- [Chapter 7:  User-Centered Design](http://www.sharritt.com/CISHCIExam/norman.html#7)

# Norman's Gulfs of Execution and Evaluation

This lecture explores a very influential view of human problem solving which has been applied extensively to human-computer interface design.

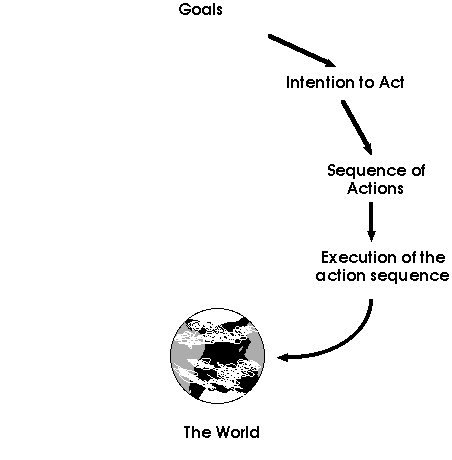
"The basic idea is simple.  To get something done, you have to start with some notion of what is wanted—the goal that is to be achieved.  Then, you have to do something to the world, that is, take action to move yourself or manipulate someone or something.  Finally, you check to see that your goal was made.  So there are four different things to consider: the goal, what is done to the world, the world itself, and the check of the world.  The action itself has two major aspects: doing something and checking.  Call these *execution* and *evaluation.*"  [Norman, p. 46.]

## The Seven Stages of Action

### 1    Forming the Goal

Something to be achieved.  Can be stated in a very imprecise way, e.g., "make a nice meal".

### Execution



#### 2    Forming the Intention

Goals must be transformed into intentions, i.e.,  specific statements of what has to be done to satisfy the goal.  E.g., "Make a chicken casserole using a can of prepared sauce."

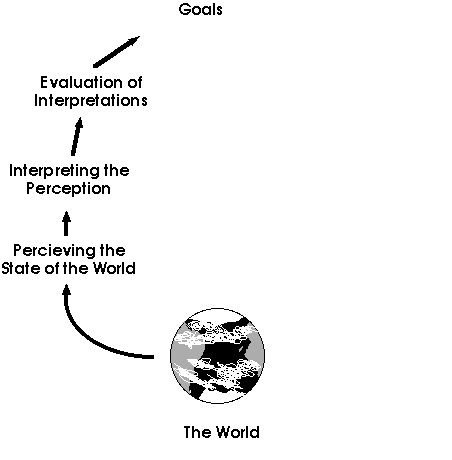
#### 3    Specifying an Action Sequence

What is to be done to the World.  The precise sequence of operators that must be performed to effect the intention.  E.g., "Defrost frozen chicken, open can, ..."

#### 4    Executing an Action

Actually doing something.  Putting the action sequence into effect on the world.  E.g., actually opening the can.

### Evaluation



#### 5    Perceiving the State of the World

Perceiving what has actually happened.  E.g., the experience of smell, taste and look of the the prepared meal.

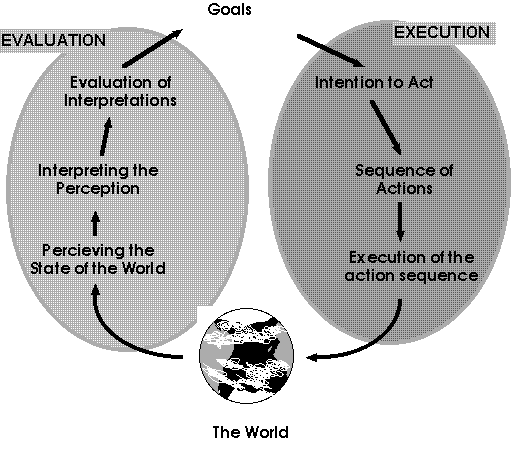
#### 6    Interpreting the State of the World

Trying to make sense of the perceptions available.  E.g., Putting those perceptions together to present the sensory experience of a chicken casserole.

#### 7    Evaluating the Outcome

Comparing what happened with what was wanted.  E.g., did the chicken casserole match up to the requirement of 'a nice meal'?

## Both Gulfs as a Single Diagram



## Practical Application

### Design Questions

The seven stages of action prompt the following design questions: [Norman, p. 53]

How easily can one:

1. determine the function of the device?
2. tell what actions are possible?
3. determine mapping from intention to physical movement?
4. perform the action?
5. tell  what state the system is in?
6. tell if system is in desired state?
7. determine mapping from system state to interpretation?

### Principles of Good Design

The significance of these questions can be summed up as the following principles of good design:

* *Visibility.*  By looking, the user can tell the state of the device and the alternatives for action.
* *A good conceptual model.*  The designer provides a good conceptual model for the user, with consistency in the presentation of operations and results and a coherent, consistent system image.
* *Good mappings.*  It is possible to determine the relationship between actions and results, between the controls and their effects, and between the system state and what is visible.
* *Feedback.*  The user receives full and continuous feedback about the results of actions.  [Norman, p. 53]

## References

Norman, D.A.  1988  "The Design of Everyday Things."  MIT Press

Interaction Design:  Beyond Human Computer Interaction  
*by Jennifer Preece*

**Preece, J., Rogers, Y., & Sharp, H.. (2002). *Interaction Design: Beyond Human-Computer Interaction*.**  
A typical undergraduate level textbook to introduce you to the field, including both scientific background and usability design methods. One of the few that adequately addresses affective measures. [DS & DN]

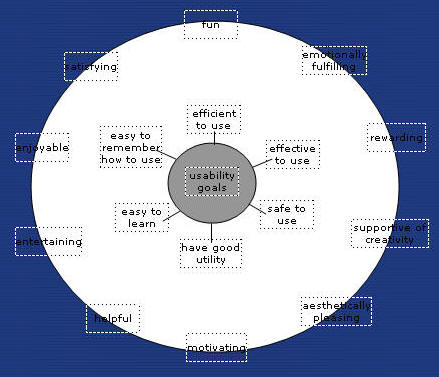
**Jump To:** [1](http://www.sharritt.com/CISHCIExam/preece.html#ch1)  [2](http://www.sharritt.com/CISHCIExam/preece.html#ch2)  [3](http://www.sharritt.com/CISHCIExam/preece.html#ch3)  [4](http://www.sharritt.com/CISHCIExam/preece.html#ch4)  [5](http://www.sharritt.com/CISHCIExam/preece.html#ch5)  [6](http://www.sharritt.com/CISHCIExam/preece.html#ch6)  [7](http://www.sharritt.com/CISHCIExam/preece.html#ch7)  [8](http://www.sharritt.com/CISHCIExam/preece.html#ch8)  [9](http://www.sharritt.com/CISHCIExam/preece.html#ch9)  [10](http://www.sharritt.com/CISHCIExam/preece.html#ch10)  [11](http://www.sharritt.com/CISHCIExam/preece.html#ch11)  [12](http://www.sharritt.com/CISHCIExam/preece.html#ch12)  [13](http://www.sharritt.com/CISHCIExam/preece.html#ch13)  [14](http://www.sharritt.com/CISHCIExam/preece.html#ch14)  [15](http://www.sharritt.com/CISHCIExam/preece.html#ch15)

***Table of Contents:***

[Ch. 1:  Intro to Interaction Design](http://www.sharritt.com/CISHCIExam/preece.html#ch1)  
[Ch. 2:  Conceptual  Models and Interface Metaphors](http://www.sharritt.com/CISHCIExam/preece.html#ch2)  
[Ch. 3:  Cognition and Mental Models](http://www.sharritt.com/CISHCIExam/preece.html#ch3)  
[Ch. 4:  Conversational, Coordination, Awareness Mechanisms and Collaboration](http://www.sharritt.com/CISHCIExam/preece.html#ch4)  
[Ch. 5:  Affective Aspects of Interfaces and User Frustration](http://www.sharritt.com/CISHCIExam/preece.html#ch5)  
[Ch. 6:  The Process of Interaction Design and Life-Cycle Models](http://www.sharritt.com/CISHCIExam/preece.html#ch6)  
[Ch. 7:  Needs and Requirements (Scenarios, Use Cases, Essential Use Cases, HTA)](http://www.sharritt.com/CISHCIExam/preece.html#ch7)  
[Ch. 8:  Prototyping, Conceptual Design and Physical Design](http://www.sharritt.com/CISHCIExam/preece.html#ch8)  
[Ch. 9:  Ethnography and Participatory Design](http://www.sharritt.com/CISHCIExam/preece.html#ch9)  
[Ch. 10:  Introduction to Evaluation](http://www.sharritt.com/CISHCIExam/preece.html#ch10)  
[Ch. 11:  Evaluation Paradigms, DECIDE Framework](http://www.sharritt.com/CISHCIExam/preece.html#ch11)  
[Ch. 12:  Observing Users](http://www.sharritt.com/CISHCIExam/preece.html#ch12)  
[Ch. 13:  Interviews, Questionnaires, Inspections and Walkthroughs](http://www.sharritt.com/CISHCIExam/preece.html#ch13)  
[Ch. 14:  User Testing, GOMS, KLM, Fitt's Law](http://www.sharritt.com/CISHCIExam/preece.html#ch14)  
[Ch. 15:  Example Applications of Interaction Design](http://www.sharritt.com/CISHCIExam/preece.html#ch15)

Preece's Summary of the Chapters:  <http://www.id-book.com/chapter_index.htm>

Remember, **usability** is of utmost importance to Interaction Design, while **user-experience** follows (see graph on p. 19 of book)

  
usability goals:  at center of Interaction Design  
user-experience goals:  outer ring of diagram (secondary to usability goals)

Chapter 1:  What is Interaction Design?

*Main Goals of this Chapter:*

* *Explain the difference between good and poor interaction design*
* *Describe what interaction design is and how it relates to HCI and other fields*
* *Explain what usability is*
* *Describe what is involved in the process of interaction design*
* *Outline the different forms of guidance used in interaction design*
* *Enable you to evaluate an interactive product and explain what is good and bad about it in terms of the goals and principles of interaction design*

**Good and Poor Design**

* Central concern of interaction design:  products that are **USABLE**
  + **Easy to use**
  + **Effective**
  + **Enjoyable**
* Example:  Marble interface versus Voicemail:  an incoming message signaled by a marble dropping through- you grab it and drop it to play the message (good interface but breaks down if system gets more complex)

**What to Design**

* Who will use it?
* Where are they going to be used?
* What kinds of activities will it support?
* A key question:  How do you optimize the users' interactions with a system, environment or product, so that they match the users activities that are being supported and extended

**Match goals to users - get them involved**

* Take into account what people are good and bad at
* Consider what might help people with the way they currently do things
* Thinking through what might provide quality user experiences
* Listening to what people might want and getting them involved in design
* Using tried and tested user-based techniques during the design process

***Interaction Design:***

**Definition:  "Designing interactive products to support people in their everyday and working lives"**

Interaction Design comes from a multidisciplinary background, extends and enhances the way people work, communicate and interact

*Interaction Design involves four basic activities:*

1. Identifying needs and establishing requirements
2. Developing alternative designs that meet those requirements
3. Building interactive versions of the designs so that they can be communicated and assessed
4. Evaluating what is being built throughout the process

**Evaluating what has been built is the heart of Interaction Design**

*Three characteristics of the Interaction Design Process:*

1. Users involved throughout the development of the project
2. Specific usability and user experience goals should be identified, documented and agreed upon at the beginning
3. Iteration through the four activities (above) is inevitable

**The Goals of Interaction Design:  Usability Goals & User Experience Goals**

**- Usability Goals:**concerned with meeting a usability criteria (e.g. efficiency)

* **Effectiveness** - how good  system is at doing what it is supposed to
* **Efficiency** - the way a system supports users in carrying out their tasks
* **Safety** - protecting the users from dangerous conditions / undesirable situations
* **Utility** - extent to which the system provides the right kind of functionality so that users can do what they need or want to do
* **Learnability** - how easy a system is to learn to use
* **Memorability** - how easy a system is to remember how to use, once learned

**- User Experience Goals:** User experience is what the interaction with the system *feels* like to the users (subjectively)

* ***Satisfying; enjoyable; fun; entertaining; helpful; motivating; aesthetically pleasing; support creativity; rewarding; emotionally fulfilling***

**Usability:  Design and Usability Goals**(generalized abstractions)

* ***Norman's Design Principles:***
  + ***Visibility*** - functions can be seen
  + ***Feedback*** - necessary part of interaction
  + ***Constraints*** - ways of restricting what kinds of interaction can take place
  + ***Mapping*** - relationship between controls and what happens
  + ***Consistency*** - similar operations / use similar elements for achieving similar goals
  + ***Affordance*** - attribute of an object that allows people to know how to use it

**Usability Principles / Heuristics** (heuristics are design principles used in practice - more prescriptive usability principles that are used as a basis for evaluating a system / prototype)

* ***Nielsen's 10 Usability Principles:***
  + ***Visibility of System Status***
  + ***Match between system and real world***
  + ***User control and freedom***
  + ***Consistency and standards***
  + ***Help users recognize, diagnose and recover from errors***
  + ***Error prevention***
  + ***Recognition rather than recall***
  + ***Flexibility and efficiency of use***
  + ***Aesthetic and minimalist design***
  + ***Help and documentation***

**There are always tradeoffs with usability - can't over constrain things, because it limits how much info is displayed**

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 2:  Understanding and Conceptualizing Interaction

have a clear understanding of **what, why and how**you are going to design something before writing any code.

Goals of the chapter:

* Explain what is meant by the problem space
* Explain how to conceptualize interaction
* Describe what a conceptual model is and explain the different kinds
* Discuss the pros and cons of using interface metaphors as conceptual models
* Debate the pros and cons of using realism versus abstraction at the interface
* Outline the relationship between conceptual design and physical design

Understanding the Problem Space

- the problem with solving a problem on the nuts and bolts level is that critical usability goals and user needs can be overlooked

- the design of physical aspects are best done AFTER we understand the nature of the problem space

- to understand the problem space:  clarify usability and user experience goals.  **Make explicit your implicit assumptions and claims.**

Framework for making your implicit assumptions explicit:

* reason through your assumption about why something might be a good idea
* this enables you to see the strengths and weaknesses of the proposed design

Conceptual Models

A conceptual model is **a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave and look like, that will be understandable by the users in the manner intended.**

"The most important thing to design is the users conceptual model." (David Liddle, '96)

To Develop a Conceptual Model:

* Envision the proposed product based on users' needs and requirements
* Do **iterative testing**
* What kind of interaction mode would support this?  Which interaction mode to use, and which interaction style to use?
* Concrete solutions to support the above comes last
* Development should be done using:  **iteration, using a number of methods, by sketching out ideas, storyboarding, and scenarios, and by making use of prototypes**

Two types of conceptual models are:

* Those based on **activities**

**1.  Instructing:**  describes how users carry out their tasks through instructing the system what to do (1-way process:  like word processing, CAD, email)

**2.  Conversing:**based on the idea of a person conversing with the system where the system acts as a dialogue performer (2-way process:  such as search engines, advisory systems, etc)

**3.  Manipulating and Navigating:**manipulating and navigating through a virtual world by using users' knowledge of the real world (like video games, virtual reality)

**4.  Exploring and Browsing:**  allows people to browse / navigate through information

* Those based on **objects**

based on objects or artifacts, and are more specific than those based on activities (focus on a particular object in a particular context - for example:  a spreadsheet, based off of a ledger sheet)

The best type of conceptual model to use depends on the nature of the activity.  Often the best answer is a hybrid (such as shopping on the Internet).  However, mixing conceptual models will raise the complexity of the system.

Interface Metaphors

definition:  a conceptual model that has been developed to be similar in some ways to aspects of a physical entity, but that also has its own behaviors and properties

Interface metaphors combine the familiar with new concepts

Benefits:  a good orientation device  
Drawbacks:  often the metaphor looks / feels like the physical entity, when they should just map the familiar with the unfamiliar so that users can learn the new (unfamiliar)

There is a growing opposition to metaphors because they can break the rules of the object they represent, they can be too constraining, can conflict with design principles, can cause misunderstanding of system functionality, can limit the designer's imagination, and can have overly literal translation of existing bad design.  [See Metaphors description for more information.](http://www.sharritt.com/CISHCIExam/metaphors.html)

Interaction Paradigms

- moving away from WIMP interface / paradigm

- new paradigms:  ubiquitous computing, pervasive computing, wearable computing, augmented reality, attentive environments

**From Conceptual Models to Physical Design**

Interaction design is an ITERATIVE PROCESS, involving:

* cycling through various design processes and different levels of detail
* thinking through a design problem
* understanding users needs
* coming up with possible models
* prototyping models
* evaluating them
* thinking about design implications
* making changes
* etc...

The book describes ways of DOING INTERACTION DESIGN

* First pass: thinking about the problem space
* Second pass: more extensive information gathering about users’ needs and problems
* Third pass: continue explicating the requirements through models
* Fourth pass: Fleshing out models using variety of user-centered methods. Such as: prototyping, storyboarding, physical objects, informally asking users what they think.

Issues in testing prototypes:

* Way information is to be presented and interacted with
* What combinations of media
* Kinds of feedback
* What combinations of input and output devices to use
* Whether to provide agents and in what format
* Whether to design operations to be hardwired or through physical objects or software
* What kinds of help to provide and in what format

Physical design decisions come out of conceptual decisions (i.e. what information, how to structure graphical objects, what feedback navigation and mechanisms, what kinds of icons…).

* These kinds of design decisions need user testing to ensure usability goals.

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 3:  Understanding Users

This chapter focuses on **USERS and COGNITION.  Cognitive aspects of Interaction Design include:**

* what humans are good and bad at
* how this knowledge can be used to *inform* design of technologies that,
* *extend* human capabilities and
* *compensate* for their weaknesses

Main aims of chapter:

* What cognition is and why it is important for I-D
* Main ways cognition has been applied to I-D
* Number of examples from cognitive rese4arch
* Explain what mental models are
* Give examples of conceptual frameworks useful for I-D
* Enable you to try to elicit a mental model and understand what it means

Norman said there are two **modes**of cognition:  **Experiential**(real world experiences) and **Reflective** (thinking, comparing, deciding, etc).  Both are necessary for everyday life.

**Cognition has been described in SIX KINDS OF PROCESSES:**

1. ***Attention*** - selecting things to concentrate on
2. ***Perception / Recognition*** - how information is acquired from the environment via sense organs and translated into experiences (vision is the most dominant)
3. ***Memory*** - recalling various knowledge.  We filter what knowledge to process / memorize. (most researched area)
4. ***Learning***- how to do something (like learning to use a program)
5. ***Reading / Speaking / Writing*** - using language
6. ***Problem Solving / Planning / Reasoning / Decision Making*** - involves reflective cognition

Often designers try to emulate the physical world with designs in the digital world.  Sometimes this works well, other times it doesn't.

Conceptual Frameworks for Cognition:

* Mental Models
* Information Processing
* External Cognition

Users' Mental Models

- defined as:  when people are using a system, they develop knowledge of how to use the system and to lesser extent how the system works.

- the mental model is used to help people carry out tasks.  It can also give suggestions on what to do in unpredictable situations

- in cognitive psychology, mental models are defined as some sort of internal construction of the external world that are manipulated enabling predictions and inferences to be made

- w/r/t system design:  ideally, the users' mental models should match the designer's conceptual model

- to increase transparency- might make system image easier to learn (p. 95 example?)

Information Processing

- another approach to conceptualize how the mind works:  through metaphors and analogies

- thinks of the mind as an information processor

- mental representations can be images, mental models, rules, other knowledge forms

**the human processor model** (Card, et. al 1983) is the best known approach (see p. 96)

* model predicts which cognitive processes are involved when a user interacts with a computer, allowing for calculations to be made on how long it will take a user to complete a task
* this is helpful for comparing different interfaces (efficiency)
* the approach is based on modeling mental **activities that happen exclusively in the head.**  There are always external cues in the environment... so how truly representative are these models?

*- there has been an increase in people studying cognitive activities 'in the wild' - in the context in which they take place* (how can things in the environment aid human cognition and lighten the cognitive load?)

Alternative frameworks have been suggested:  **External cognition and Distributed Cognition**

External Cognition

main idea:  people interact with or create information through using a variety of external representations (books, etc.)

- an impressive array of technology has been created by humans to aid cognition (calculators, pens, etc)

- these tools have combined with external representations to extend and support our ability to carry out cognitive activities.

- some of the main goals of this:

* *Externalizing to reduce memory load* (external representations / cues as reminders)
* *Computational offloading* (using a tool / device to carry out a computation - like a pen / paper to do a math problem) Note:  representation of the task is key- imagine the difficulty of multiplication if the numbers were represented as Roman numerals
* *Annotating / Cognitive tracing*(modifying representations to show changes - like crossing something off a list

Back to Interaction Design:  PROVIDE EXTERNAL REPRESENTATIONS AT THE INTERFACE TO REDUCE MEMORY LOAD (visualizations, cues, etc).

Informing Design:  From Theory to Practice

Theories, models and frameworks provide abstractions for thinking about phenomena.  They provide generalizations, but can be difficult to digest.  For this reason researchers have tried to make them more practical by providing design principles / concepts, design rules, analytic methods and design / evaluation methods.

This has helped - for instance - the human processor model (Card, 83) which has been simplified into GOMS.

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 4:  Designing for Collaboration and Communication

humans are SOCIAL beings

Purpose of the chapter:  ***look at ways interactive systems could be developed to support and extend communication and collaboration between peoples.***

Social Mechanism in Communication and Collaboration:

Rules, procedures and etiquette have been established to help people know how to behave in social groups, such as:

* **Conversational mechanisms**- to help the flow of talk and to help overcome breakdowns
* **Coordination mechanisms**- to allow  people to work / interact together
* **Awareness mechanisms**- to find out what is happening, what others are doing and to let others know what is happening

**Conversational Mechanisms:**

* "turn-taking" helps coordinate conversation
* Implicit cues and Explicit cues (indirect vs. direct / obvious)
* Turn taking rules:  speaker chooses next speaker by asking question / request, etc.
* Back channeling, body orientation, gaze, gestures are used to signal to others the flow of conversation
* Farewell rituals help end a conversation (bye, see ya later)
* Breakdowns in conversation occur when someone is ambiguous and it gets misinterpreted (followed by a re-questioning)
* Conversations can take the form of arguments, discussions, debate, chat, etc.

How to design collaborative technologies to support conversation:

* First, how do technology-mediated conversations compare to FTF?  Do the same rules apply?  Are there more breakdowns?  How do they get repaired?
* Design implications:  A key issue has been to determine how to allow for and support people to carry on communicating as if they were in the same place, even thought they are geographically separated.
* Some existing apps:  phone, videophone, email, IM, videoconferencing, chatrooms, SMS texting
* How successful are these?  Do they mimic or extend existing ways of conversing?

Synchronous CMC: (p.112 table 4.1)

* real-time conversations, like a chatroom
* Pros:  more informed of what's going on, can allow shy people to talk more, if video support:  can allow nonverbal communication to occur
* Cons:  bandwidth issues can cause video to get choppy, very hard to establish eye contact, people can behave badly behind the mask of an avatar

Asynchronous CMC:

* remote communication at various times, such as email, newsgroups
* Pros:  read at any time, flexible response, easier to say things, can contact many people easily
* Cons:  Flaming, spamming, new message overload, don't know when people will reply

Many new communication technologies combine the above, and try to provide new / novel ways to communicate

* Collaborative virtual environments, media spaces, shared drawing tools, tools for collaborative document creation
* Pros:  support talking while doing a task at same time, can be efficient to have multiple people working on the same thing at the same time, and greater awareness of what is going on
* Cons:  WYSIWIS:  we can't always see what people are referring to in a remote location, and floor control (file conflicts from multiple people working on the same thing at the same time)

**Coordination Mechanisms:**

Collaborative activities require us to coordinate with each other, so we need to figure out how to work with others to progress through the activities.  Examples include:

* Verbal and Nonverbal communication (commands, gestures, nods, shakes, hand raising, to coordinate their communication)
* Schedules, rules and conventions (to organize people who take part in a project- can be formal or informal)
* Shared external representations (allow people to make inferences about the changes / delays on their current project)

How to design collaborative technologies to support coordination:

Shared calendars, schedulers, project management tools, and workflow tools have been developed to support coordination activities.

People tend not to follow conventions, because they are often not socially acceptable.  Failure to make them socially acceptable can cause people to not use the system in the way intended or can cause them to abandon it totally.

**Awareness Mechanisms:**

These provide others with awareness of who is around, what is happening, who they are talking to.  This requires knowing when is an appropriate time to interact with others and to get / pass information.

How to design collaborative technologies to support awareness:

Function is to make others aware of the others they are collaborating with.  For example:

* Portholes:  a series of digitized images showing people in their offices from various locations (led to increased sense of community)
* Notification systems:  users notify others, rather than being monitored, and provide information about shared objects and progress of collaborative tasks (so others can see each other and their progress)

**Ethnographic studies of collaboration and communication**

A main approach to informing the design of collaborative technologies that takes into account the social concerns is to carry out an ethnographic study.

This can be a home, work, school, public place (setting)

Since Suchman's "Seminal Work" many companies have invested in ethnographic studies to see how work actually gets done in a range of companies (government too)

**Conceptual Frameworks:  Language / Action Framework and Distributed Cognition**

Language / Action Framework:

* people act through language
* goal to inform the design of systems to help people work more effectively by improving the way they communicate with each other
* this framework doesn't take into account the use of artifacts / external representations in everyday work

Distributed Cognition:

* describes what happens in a cognitive system:  interactions among people, artifacts they use, environment they work in
* the way that cognitive activity is described contrasts with others in that it focuses on not only what is happening in the head of individuals, but on what is happening across individuals and artifacts
* Examines:  distributed problem solving that is taking place, role of verbal / nonverbal behavior, coordination mechanisms that are used, communicative pathways that take place as collaborative activity progresses, and how knowledge is shared / accessed

**Summary:**

* **social mechanisms, like turn taking conventions, allow us to collaborate / coordinate our activities**
* **keeping aware of what others are doing and letting others know what you are doing are important aspects of collaborative learning / socializing**
* **many collaborative technologies (CSCW / groupware) systems have been built to support collaboration, especially communication and awareness**

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 5:  Understanding How Interfaces Affect Users

Goals of this chapter:  **AFFECTIVE LEARNING - a way to design systems to elicit positive responses from users (feeling at ease, being comfortable, enjoying the experience)**

* How can the appearance of the interface elicit positive responses from the user?
* How can user frustration be caused by an interface?
* How do interface agents (anthropomorphism) and synthetic characters affect us?

**Affective Aspects of HCI**

* traditionally, HCI has been about designing efficient and effective systems
* recently, HCI has moved towards considering how to design interactive systems to make people respond in a certain way (to be happy, to be trusting, to learn, to be motivated)
* It has been suggested that computers be designed to recognize and express emotions in the same way that humans do
* How can interactive systems be designed (both deliberately and inadvertently) to make people respond in certain ways?

**Expressive Interfaces**

* Colors, icons, sounds, graphical elements and animations are used to make the "look and feel" of an interface appealing
* *A benefit is that these embellishments provide reassuring feedback to the user that can be both informative and fun- which can affect the usability of the interface*
* People are willing to put up with certain aspects of an interface (slow download rate, etc) if the end result is very appealing and aesthetic
* Aesthetics have been shown to have a positive effect on people's perception of the system's usability
* Some friendly interfaces:  Microsoft's 'at home with Bob' interface, 3D metaphors (living rooms, etc), agents in the guise of pets (dog) that talk to the user.  These make users feel more at ease and comfortable.
* User-created expressiveness:  **emoticons** - these provide non-verbal type expression in interfaces not originally intended to have this :-)  Also, icons / shorthand have been used to add emotion to SMS texting (I 12 CU 2NITE)

**User Frustration**

*Can be caused by:*

- application doesn't work / crashes  
- system won't do what the user wants it to do  
- user's expectations aren't met  
- when the system doesn't provide enough information to enable the user to know what to do  
- vague error messages  
- condemning error messages  
- when the interface's appearance is patronizing, gimmicky, noisy  
- when the user does a long series of steps to complete a task, only to discover an error was made and they have to start over

*Things to avoid:*

- Gimmicks:  "under construction"  
- Error messages:  "the system has unexpectedly quit" (see Shneiderman's guidelines for error messages:  don't use "FATAL" "INVALID" or "BAD", or long hex codes - try to provide context sensitive help)  
- Overburdening the user:  forcing them to upgrade, install plugins, do housekeeping just to use a product  
- Unpleasant Appearance:  too much crap on the page, featuritis (too many features), weird sound effects, childish interface, poorly laid out input devices

*Dealing with User Frustration:*

- people will vent, by beating the hell out of their computers, flaming, etc.  
- offer helpful error messages that offer a way to fix the problem, and offer hints, help guides, cartoon agents, etc. that can help point the user in the right direction  
- Reeves and Naas (1996) argue that computers should apologize when they mess up

**Anthropomorphism in HCI:  How much is enough?**

Anthropomorphism- assigning human traits to non-human things (dancing butter, talking soda cans, dogs, cars, etc.)  
- used heavily in advertising

People debate how much of this to use in system design.  They can add a human feel to the system, but can also get annoying.

*Which is more preferable?*  
- "Hello Matt!  Welcome back.  It's nice to see you again.  Now, what were we doing last time?  Ah, yes, problem five.  Lets get started again."  
- "User 24, commence exercise 5."

Or, when doing something wrong:  "Now Matt, that's not right, you can do better than that.  Try again." vs. "Incorrect, try again."

*The answer:*    
Pros:  Reeves and Naas (1996) found it is helpful to use praise in educational settings when people do something right.  It increased students willingness to continue working.    
Cons:  However, others argue this can make you feel stupid, anxious, inferior.  People hate when a computer character shakes their finger at them and says "you can do better than that, Matt, try again".  In this case, many prefer the impersonal message "Incorrect, try again".

**Virtual Characters**

virtual characters are becoming more common.  They can be used on the web, in video games, as learning companions, wizards, newsreaders, etc.  However, they can be misleading (people confide in them), they can be very annoying and frustrating ("Clippy" from MS Office 97, etc).

- categorized by degree of anthropomorphism:  synthetic characters, animated agents, emotional agents, embodied conversational agents

*Design Implications:  which one to use?*  
**- believability:**  the extent to which users come to believe an agent's intentions and personality  
**- appearance:**  better to use cartoon-like characters, or those resembling humans?  it depends on the situation- often the cartoon character is more believable / acceptable  
**- behavior:**  how does the agent move, gesture, refer to things?  facial expressions can show emotion (remember we want constructive feedback rather than conveying inferiority / stupidity / patronizing effect on users)

**Summary:**

* affective aspects are concerned with how interactive systems make people respond in emotional ways
* well designed interfaces can elicit good feelings in users
* expressive interfaces can provide reassuring feedback
* badly designed interfaces make people angry and frustrated
* anthropomorphism is increasingly used at the interface, through the use of agents and virtual screen characters

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 6:  The Process of Interaction Design

The ultimate goal of design is to develop a product that helps its users achieve their goals.  Developing a product must begin with gaining understanding of what is required of it.

The goals of this chapter are to:

* Consider what 'doing' interaction design involves
* Ask and provide answers for some important questions about the interaction design process
* Introduce the idea of a lifecycle model to represent a set of activities and how they are related
* Describe some lifecycle models from software engineering and HCI and discuss how they relate to the process of interaction design
* Present a lifecycle model of interaction design

What is Interaction Design?

dictionary:  "design is a plan or scheme conceived in the mind and intended for subsequent execution"

The plan or scheme must be informed with knowledge about its use and the target domain, together with practical constraints such as materials, cost and feasibility.

In Interaction Design, we investigate the artifact's use and target domain by taking a user-centered approach to development.  The users' concerns direct the development rather than technical concerns.

Design is also about trade-offs and about balancing conflicting requirements.  Generating alternatives is a key principle and one that should be encouraged in interaction.  "To get a good idea, get lots of ideas." (Mark Rettig)

Typically there is a group of designers.  Therefore, plans should be captured and expressed in a way that allows for review, such as sketches, descriptions in natural language, a series of diagrams, and building prototypes.

Four Basic Activities:

*1.  Identify needs and establish requirements*  
- Who is the target user?  
- What kind of support will the interactive product provide?

*2.  Develop alternative designs that meet those requirements*  
- suggest ideas to meet the requirements  
- Conceptual design:  produce the conceptual model for the product  
- Physical design:  consider the details of the product (colors, sounds, images, menu design, icons, etc.)  Alternatives are considered at every point.

*3.  Build interactive versions (so that they can be communicated and assessed)*  
- a software version is not required- paper based prototypes are quick and cheap to build  
- through role-playing, users can get a real sense of what it is like to interact with the product

*4.  Evaluate the designs (measure their acceptability)*  
- determine the usability of the product or design.  Criteria are:  how appealing is it?  how well does it match the requirements?  Is the product fit for the purpose?  
- Evaluation results are fed back into further design (FEEDBACK / ITERATIVE DESIGN PROCESS)

Three Characteristics of Interaction Design:

**1.  Focus on the USERS**  
- involve users in the interactive design process, provide opportunities for evaluation and user feedback

**2.  Specific usability and user experience goals**  
- identify and clearly document these at the beginning of the project.  They help designers to choose between different alternative designs

**3.  Iteration**  
- allows for designs to be refined.  It is always necessary to revise ideas in light of feedback, several times.  Innovation rarely emerges whole and ready to go.  Iteration is inevitable because designers never get the solution right the first time

Practical Issues in Interaction Design:

***1.  Who are the users?***  
- those who directly interact with the system.  However, can be any stakeholder:  purchaser, testers, people receiving products from the system  
- primary user:  directly use it  
- secondary user:  occasionally use it or use through intermediary  
- tertiary user:  affected by the system, or will influence its purchase  
- stakeholders:  people or organizations affected by the system who influence the system requirements

**2.  What do we mean by "needs"?**- we need to understand the characteristics / capabilities of users, what they are trying to achieve, how they currently achieve it, and whether they would achieve their goals more effectively if they were supported differently  
- characteristics that impact a product's design:  users physical characteristics (ergonomics:  size of hands, height, etc), strength of product (so a child can't break it), cultural diversity of intended users  
- representative users MUST be consulted!  
- users rarely know what is possible.  Therefore, users cannot tell us what they "need" to do achieve their goals.  
***- We need to examine existing task (Activity Theory?) and what the tasks' context, requirements, collaborative nature, and procedure is.  Then we can envision the task being done in a new way (scenarios, etc.)***

**3.  How do you generate alternative designs?**  
- it is easy to stick with something that is "good enough".  Humans stick to what they know works.  
- innovations arise from cross-fertilization from different applications- allows us to "break out of the box"  
- often browsing a collection of designs will inspire designers to consider alternative perspectives and solutions.  Designers are trained to consider alternatives, software people are not.  
- design is a process of balancing constraints and constantly trading off one set of requirements with another, and the constraints may be such that there are few viable alternatives available.  
- alternatives come from looking at other, similar designs, and the process of inspiration and creativity can be enhanced by prompting a designer's own experience and by looking at others' ideas and solutions.  ("Flair and creativity" : research and synthesis)

**4.  How do you choose among alternative designs?**- there are factors that are externally visible and measurable and those that are hidden from the users' view.  *Focus on the external / visible*.  
- prototypes can be used to evaluate with peers and users  
- fundamental user-centered design:  choose between alternative designs by letting users and stakeholders interact with them and by discussing their experiences, preferences and suggestions for improvement.  
- technical feasibility:  some are just not possible  
- quality thresholds:  usability goals lead to criteria.  This USABILITY CRITERIA need to be set early on and checked frequently.  
- usability engineering:  the process of writing down formal, verifiable and measurable usability criteria.  Some suggestions:

* Effectiveness:  Appropriate support?  Task coverage, information available
* Efficiency:  response time?  Performance measurements?
* Safety:  How safe?  How often does it crash / loose data?
* Utility:  Which functions are superfluous?
* Learnability:  How long does a novice take to learn?  High learning curve?
* Memorability:  How log to remember how to perform common tasks?

**Lifecycle Models:  Showing how the activities are related**

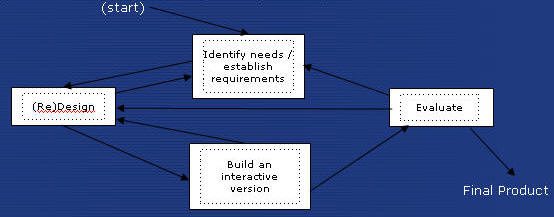
- lifecycle models:  represent a set of activities and how they are used; management tools; simplified versions of reality  
  
- some models from software engineering:  waterfall, spiral, RAD, etc

* Waterfall model:  a linear process where each step must be completed before moving to the next.  This is bad because there is no iteration, and modifications cannot be made to the design.  Users cannot evaluate prototypes
* Spiral model:  two features:  risk analysis and prototyping.  Alternatives are considered and encouraged.
* RAD (Rapid Applications Development):  takes a user centered view and tries to minimize the risk of changing requirements through the project.  A system or partial system must be delivered on a set of intervals.

- some models from HCI:  Star, usability engineering

* Star Lifecycle model:  does not specify order of activity.  All activities are highly interconnected.  You can move from one activity to another easily, but you MUST go through the evaluation activity (in the center).  Evaluation is central to this model
* Usability Engineering Lifecycle model:  provides a holistic view of usability engineering and a detailed description of how to perform usability tasks.  This is helpful for those with little experience.  Three phases:  requirements analysis; designing / testing / development; installation

- a simple lifecycle model for interaction design:  see p. 186, Fig. 6.7

  
  
(note how the above principles apply to this model)

**Summary:**

*There are four basic activities in the interactive design process:*  
1.  Identify needs and establish requirements  
2.  Develop ([re]design) alternative designs that meet those requirements  
3.  Build interactive versions of the designs so that they can be communicated and assessed  
4.  Evaluate them

*These are permeated with three principles:*1.  Involve users early in the design process and evaluation of the artifact  
2.  Define quantifiable & measurable usability criteria  
3.  Iteration is inevitable

Key characteristics of the interaction design process are explicit incorporation of user involvement, iteration and specific usability criteria.

Before you can begin to establish requirements, you must understand who the users are and what their goals are in using the device.

Looking at others' designs provides useful inspiration and encourages designers to consider alternative design solutions, which is key to effective design.

Usability criteria, technical feasibility, and users' feedback on prototypes can all be used to choose among alternatives.

Prototyping is a useful technique for facilitating user feedback on designs at all stages.

Lifecycle models show how development activities relate to one another.

The interaction design process is complementary to lifecycle models from other fields.

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 7:  Identifying Needs and Establishing Requirements

***This chapter talks about different ways to gather requirements by introducing:***

* **Types of Requirements**
* **Data Gathering Techniques**
* **Task Descriptions**
  + **Scenarios**
  + **Use Cases**
  + **Essential Use Cases**
* **Task Analysis**
  + **Hierarchical Task Analysis (HTA) (task analysis)**

What are we trying to achieve in this design activity?

1. Understand as much as possible about users, task, and context
2. Produce a stable set of requirements

How can we do this?

* Data gathering activities
* Data analysis activities
* Expression as 'requirements'
* All of this is **iterative**

Why bother getting it right?

* Typically, the requirements definition stage is the most common place for failure
* Getting the requirements right is crucial, because unclear objectives will cause a project to FAIL
* A 'user-centered approach' to development is the way to solve this problem (What do users want?  What do users 'need'?)

Requirements need clarification, refinement, completion and re-scoping.

Input:  requirements document (maybe)    
Output:  stable requirements

Why 'establish' requirements?

* 'Establish' requirements:  we *establish* requirements because -they arise from the data-gathering and interpretation activities and have been *established* from a sound understanding of the users' needs.
* Because of this, requirements can be *justified by* and *related back to* the data collected.

***Types of requirements:***

* **Functional requirements:**  what the system should do (historically this was the main function of requirements activity)
* **Non-functional requirements:**  memory size, response time, date product must be finished by, etc.
* **Data requirements:**  What kind of data needs to be stored, how will they be stored (database)?
* **Environment / Context of use requirements:** Circumstances in which product will be used / expected to operate
  + What are the *physical* (dusty, noisy, vibration, light, heat humidity, etc) requirements?
  + What are the *social* (sharing of files, displays, paper across distances, working individually, privacy of clients, etc.) requirements?
  + What are the *organizational* (hierarchy, IT department's attitude, user support, communication structure / infrastructure, training ability, etc.) requirements?
* **User requirements:**Who are they? - captures the characteristics of the intended user group
  + *Characteristics* - ability, background, attitude to computers
  + *System use* - novice, expert, casual, frequent
    - Novice:  step by step (prompted), constrained, clear information
    - Expert:  flexibility, access / power
    - Frequent:  short-cuts
    - Casual / infrequent:  clear instructions (such as menu paths)
* **Usability requirements:** (note:  different than user requirements) - these capture the usability goals and associated measures for a particular project.
  + learnability
  + throughput
  + flexibility
  + attitude

***Data Gathering Techniques:  (see p.214 / Table 7.1 for excellent graph)***

* **Questionnaires**
  + elicit specific information
  + can be YES / NO, multiple choice, comment
  + often used with other techniques
  + can give quantitative / qualitative data
  + good for answering specific questions from a large, dispersed group of people
* **Interviews**
  + forum for talking to people
  + can be structured, unstructured, or semi-structured
  + props, scenarios of use, prototypes can be used
  + good for exploring issues
  + can be time consuming, infeasible to visit everyone
* **Workshops / Focus Groups**
  + group interviews
  + good at gaining a consensus view and / or highlighting areas of conflict
* **Naturalistic Observation**
  + spend time with stakeholders in their day-to-day tasks, observing work as it happens
  + gain insight into stakeholders' tasks
  + good for understanding the nature / context of tasks
  + Requires time and commitment from a member of the design team, and can result in huge amounts of data
  + *ethnography* is one form of this
* **Studying Documentation**
  + procedures and rules are often written down in manuals
  + good source of data about the steps involved in an activity, and any regulations governing a task
  + not to be used in isolation
  + good for understanding legislation and getting background information
  + no stakeholder time, which is a limiting factor on the other techniques

Which of the above data gathering techniques to use?  The above techniques differ in the *amount of time, level of detail and risk associated with the findings*, and *the knowledge the analyst requires*

The choice of technique is also affected by the kind of task to be studied:

* sequential steps or overlapping series of subtasks?;
* high or low, complex or simple information?;
* task for a layman or skilled practitioner?

Problems with data-gathering:

* identifying and involving stakeholders:  users, managers, developers, customer reps?, union reps?, shareholders?
* involving stakeholders:  workshops, interviews, workplace studies, co-opt stakeholders onto the development team
* 'Real' users, not managers:  traditionally a problem in software engineering, not so bad now
* requirements management:  version control, ownership
* communication between parties:  within development team, with customer / user, between users
* domain knowledge distributed / implicit:  difficult to dig up and understand
* availability of key people
* political problems
* dominance of certain stakeholders
* economic / business environment changes
* balancing functional and usability demands

**Guidelines:**

* focus on identifying the stakeholders' needs
* involve all the stakeholder groups
* involve more than one stakeholder from each group
* use a combination of data gathering techniques
* start data interpretation soon after the data gathering session
* do an initial interpretation before deeper analysis
* use different approaches to different problems, such as *class diagrams* for object-oriented systems, and *entity-relationship (E-R) diagrams* for data intensive systems

***Task Descriptions:  (more on p. 226-231)***

* **Scenarios** - an *informal narrative story*, simple, 'natural', personal, not generalizable
* **Use Cases** - assume interaction with a system, assume detailed understanding of the interaction
* **Essential Use Cases** - abstract away from the details, does not have the same assumptions as use cases

*Scenario* for a shared calendar:

“The user types in all the names of the meeting participants together with some constraints such as the length of the meeting, roughly when the meeting needs to take place, and possibly where it needs to take place. The system then checks against the individuals’ calendars and the central departmental calendar and presents the user with a series of dates on which everyone is free all at the same time. Then the meeting could be confirmed and written into people’s calendars. Some people, though, will want to be asked before the calendar entry is made. Perhaps the system could email them automatically and ask that it be confirmed before it is written in.”

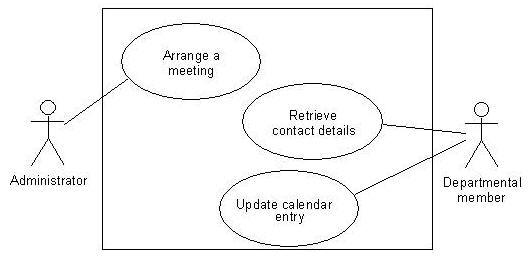
*Use case* for a shared calendar:

1. The user chooses the option to arrange a meeting.  
2. The system prompts user for the names of attendees.  
3. The user types in a list of names.  
4. The system checks that the list is valid.  
5. The system prompts the user for meeting constraints.  
6. The user types in meeting constraints.  
7. The system searches the calendars for a date that satisfies the constraints.   
8. The system displays a list of potential dates.  
9. The user chooses one of the dates.  
10. The system writes the meeting into the calendar.  
11. The system emails all the meeting participants informing them of them appointment

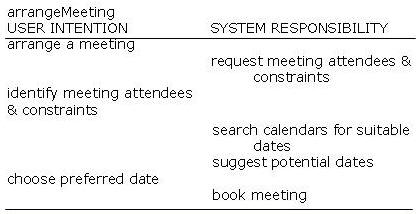
Alternative courses for a shared calendar:

Some alternative courses:   
    5. If the list of people is invalid,  
        5.1 The system displays an error message.  
        5.2 The system returns to step 2.  
    8. If no potential dates are found,  
        8.1 The system displays a suitable message.  
        8.2 The system returns to step 5.

*Example Use Case Diagram* for a shared calendar:



Example *Essential Use Case* for a shared calendar:



***Task Analysis:  (p.231-234)***

Task analysis is an umbrella term that covers techniques for investigating cognitive processes and physical actions, at a high level of abstraction and in minute detail.

***Hierarchical Task Analysis -***involves breaking down a task into subtasks, then sub-sub-tasks and so on.  These are grouped as plans which specify how the tasks might be performed in practice

* HTA focuses on physical and observable actions, and includes looking at actions not related to software or an interaction device
* HTA starts with a user goal which is examined and the main tasks for achieving it are identified
* These tasks are then divided into sub-tasks

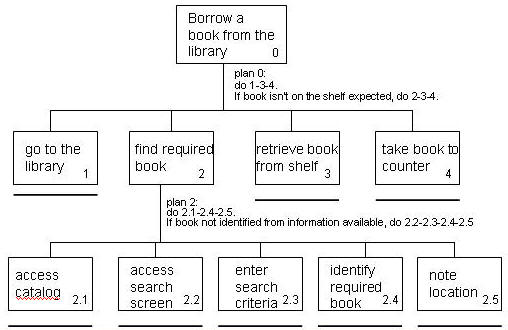
Example Hierarchical Task Analysis:  Borrowing a book from the library

0. In order to borrow a book from the library   
    1. go to the library   
    2. find the required book  
        2.1 access library catalogue  
        2.2 access the search screen  
        2.3 enter search criteria  
        2.4 identify required book   
        2.5 note location  
    3. go to correct shelf and retrieve book  
    4. take book to checkout counter

Example HTA (Plans):

* plan 0: do 1-3-4. If book isn’t on the shelf expected, do 2-3-4.
* plan 2: do 2.1-2.4-2.5. If book not identified do 2.2-2.3-2.4.

Example HTA (Graphical):



Summary:

•

* Getting requirements right is crucial
* There are different kinds of requirements, each is significant for interaction design
* The most commonly-used techniques for data gathering are: questionnaires, interviews, focus groups and workshops, naturalistic observation, studying documentation
* Scenarios, use cases and essential use cases can be used to articulate existing and envisioned work practices.
* Task analysis techniques such as HTA help to investigate existing systems and practices

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 8:  Design, Prototyping and Construction

This chapter will cover:  prototyping and construction (low and high fidelity prototyping, vertical and horizontal compromises); conceptual design (conceptual model, using scenarios and prototypes in conceptual design); and physical design (guidelines and widgets).

Flow of Interaction Design:  
Identify Needs / Requirements (Ch. 7) --> Prototype cycles / Design (Ch. 8) --> Construction

What is a Prototype?

* in other fields, it's a small scale model (miniature car, building, etc)
* in Interaction Design it can be a series of screen sketches, a storyboard, a PowerPoint, a video simulating use of the system, a lump of wood (e.g. PalmPilot), a cardboard mock-up, or a piece of software with limited functionality

Why Prototype?

* Evaluation and feedback are central to interaction design
* stakeholders can see, hold, interact with a prototype more easily than documents / drawings
* team members can communicate effectively
* can test out ideas immediately
* it encourages reflection, a very important aspect of design
* prototypes answer questions and support designers in choosing between alternatives

What gets prototyped:  technical issues, work flow, task design, screen layouts / information displays, and any difficult / controversial / critical areas

Low-fidelity Prototyping:

* uses a medium unlike the final product (paper, cardboard)
* quick, cheap and easily changed
* can be screen sketches (drawing ability not important, practices simple symbols), task sequences, post-it notes, storyboards (often used with scenarios, and consists of a series of sketches [such as 3x5 index cards] showing how a user might progress through a task using the device)
* low fidelity prototypes have limited functionality / utility, but are helpful for identifying requirements and evaluating multiple design concepts

High-fidelity Prototyping:

* uses materials you would expect in the final product
* prototype looks more like final version than a low-fidelity version
* include software environments like Macromedia Director, Visual Basic, Smalltalk, etc.
* one drawback / compromise is that users might think they have a full system.  High fidelity prototypes are time consuming / expensive to make, and are not effective in requirements gathering

Compromises associated with Prototyping:

* every prototype has a compromise - for software this may be slow response time, sketchy icons, limited functionality, etc.
* Two types of compromise:  horizontal and vertical
* 'horizontal' compromise:  provide a wide range of functions, but with little detail
* 'vertical' compromise:  provide a lot of detail for only a few functions
* Compromises must not be ignored:  *products need to be engineered*

Construction:  taking a prototype and making it whole by engineering a complete product (focus on quality:  usability, reliability, robustness, maintainability, integrity, portability, efficiency, etc.)

**Conceptual Design:  From requirements to design**

* transforms user requirements / needs into a conceptual model, "a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave and look like, that will be understandable by the users in the manner intended"
* should ITERATE, ITERATE, ITERATE:  don't move to a solution too quickly
* should consider alternatives (prototyping helps this) Fudd's first law of creativity:  "To get a good idea, get lots of ideas" (Rettig, 1994)

Three Perspectives for a Conceptual Model:

1. *Which interaction mode?*  
     
   - how user invokes actions  
   - Activity-based:  instructing, conversing, manipulating and navigating, exploring and browsing  
   - Object-based:  structured around real-world objects
2. *Which interaction paradigm?*  
     
   - desktop paradigm with WIMP interface  
   - ubiquitous computing  
   - pervasive computing  
   - wearable computing  
   - mobile devices, etc.
3. *Is there a suitable metaphor?*  
     
   - Interface metaphors combine familiar knowledge with new knowledge in a  way that will help the user understand the product  
   - 3 steps:  understand functionality, identify potential problem areas, generate metaphors  
   - evaluate metaphors:  How much structure does it provide?  How much is relevant to the problem?  Is it easy to represent?  Will the audience understand it?  How extensible is it?

Expanding the Conceptual Model:

* What functions will the product perform?  What will the product do and what will the human do?  (task allocation)
* How are the functions related to each other?  Sequential or Parallel?  How are they categorized?
* What information needs to be available?  What data is required to perform the task?  How is this data to be transformed by the system?

Scenarios:  Using them for Conceptual Design

- scenarios can be used to explicate existing work situations, but are more commonly used for expressing proposed or imagined situations to help in conceptual design.

- they can be used through design in various ways:  as scripts for user evaluation of prototypes, as a means of co-operation across professional boundaries

- in extreme cases, *plus and minus* scenarios can be used, which attempt to capture the most positive and the most negative consequences of a proposed design solution

Prototypes:  Using them for Conceptual Design

- prototypes allow for evaluation of emerging ideas

- low-fidelity prototypes are used early on, while high-fidelity prototypes are used later

**Physical Design: Getting Concrete**

* physical design considers more concrete & detailed issues of designing the interface
* can be things like screen or keypad design, which icons to use, how to structure menus
* Some guidelines for physical design:
  + Nielsen's heuristics (see [Chapter 1](http://www.sharritt.com/CISHCIExam/preece.html#ch1))
  + **Shneiderman's eight golden rules:**

1.  Be consistent  
2.  Enable frequent users to use shortcuts  
3.  Offer informative feedback (meaningful error messages)  
4.  Design dialogs to yield closure (like when you complete a task)  
5.  Offer error prevention and simple error handling (to err is human, so figure that in to your design)  
6.  Permit easy reversal of actions ('undo' button)  
7.  Support internal locus of control (user feels in control)  
8.  Reduce short-term memory load (less info to remember between screens)

* style guides (commercial, corporate, etc. - decide the 'look and feel', along with widgets [icons, menus, toolbars, dialog boxes, etc])
  + **menu design:**  How long will menu be?  In what order?  How will they be structured (sub-menus / dialog boxes)?  What categories will group menu items?  How will division of items be denoted?  How many menus?  What terminology will be used?  What physical constraints (mobile phone) must be accommodated?
  + **icon design:**  can be difficult, as icons can be cultural / context sensitive - so draw on traditions / standard, use concrete objects
  + **screen design:**  Split screen?  How much white space?  How to group things (boxes / lines / colors)?  Draw attention to the focus point, using color, motion, possibly animation, and use good organization.  Balance the tradeoff between overcrowded / sparse displays
  + **Information display:**  show only relevant information, make different mediums (computer / paper) consistent

*There is no rigid border between conceptual and physical design... they are all iterative processes.  Often in conceptual design some detailed issues come up in the iterations.  The important part is that in the conceptual design that we don't get tied to physical constraints early as they will inhibit creativity and limit our options.*

Summary:

* Different kinds of prototyping are used for different purposes and at different stages
* Prototypes answer questions, so prototype appropriately
* Construction: the final product must be engineered appropriately
* Conceptual design (the first step of design)
* Physical design: e.g. menus, icons, screen design, information display
* Prototypes and scenarios are used throughout design

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 9:  User-Centered Approaches to Interaction Design

***The main aims of this chapter are to:***

* *Explain some advantages of involving users in development.*
* *Explain the main principles of a user­centered approach.*
* *Describe some ethnographic­based methods aimed at understanding users' work.*
* *Describe some participative design techniques that help users take an active part in design decisions.  (users as co-designers will raise acceptance of product)*

***Why involve users at all?***

1. To manage their *expectations:*  no surprises, communicate their expectations, get a common set of realistic expectations
2. So users feel *ownership:*  by making users active stakeholders, they are more likely to forgive / accept problems, and are more likely to accept the final product

***Degrees of user involvement:***

* **member of the design team**(*part time vs. full time:*  degree of input / time and contact; *short term vs. long term:* degree of consistency across project life.  Long term members might loose contact with users)
* **newsletters / e-mail / etc**(disseminate information to a large selection of users, but requires 2-way communication, not 1-way)
* Actual user involvement may be a combination of the above two ways.
* Microsoft involves users by 'activity based planning' (studying users doing tasks), usability tests, internal developer usage of products, and customer support lines.

***What is a user-centered approach?***User-centered approach is based on:

* *Early focus on users and tasks:* directly studying cognitive, behavioral, anthropomorphic & attitudinal characteristics
  + users' tasks and goals are the driving force behind the development
  + users' behavior and context of use are studied and the product is designed to support them
  + users' characteristics are captured and designed for
  + users are consulted throughout development, from earliest phases to the latest, and their input is seriously taken into account
  + all design decisions are taken within the context of the user, their work, and their environment
* *Empirical measurement:*users’ reactions and performance to scenarios, manuals, simulations & prototypes are observed, recorded and analyzed
* *Iterative design:* when problems are found in user testing, fix them and carry out more tests

***Understanding Users' Work:  Ethnography***

* ***Ethnography*** stems from anthropology, and literally means 'writing the culture' - a form of participant observation.  However, it is difficult to use the output of ethnography in design.  Design is concerned with abstraction and rationalization, while ethnography is concerned with minute details, so it is difficult to harness the data gathered from ethnography so that it can be used in design.
* *Framework for using ethnography in design:*
  + *distributed coordination:*  distributed nature of the tasks / activities and the means / mechanisms by which they are coordinated
  + *plans and procedures:*  organizational support for the work, such as workflow models and organizational charts, and how these are used to support the work
  + *awareness of work:*  how people keep themselves aware of others' work
* ***Coherence:***  a method offering questions to address these dimensions (above) by presenting the ethnographic study data as a set of *"viewpoints"* and *"concerns"*

**Examples:***Distributed coordination:*  How is the division of labor manifested through the work of individuals and its coordination with others?   *Plans and procedures:*  How do plans and procedures function in the workplace?   *Awareness of work:*  How does the spatial organization of the workplace facilitate interaction between workers and with the objects they use?

***Contextual Design:***  developed to handle data collection and analysis from fieldwork for developing a software-based product (used commercially quite widely)  There are seven parts to Contextual Design:

**1.  Contextual Inquiry  
2.  Work Modeling   
3.  Consolidation  
4.  Work Re-design  
5.  User Environment Design  
6.  Mock-up and test with customers  
7.  Putting it into Practice**

***1.  Contextual Inquiry:***  an approach to ethnographic study where the user is an expert, and the designer is an apprentice.  It is a form of interviewing, but takes place at the users' workplace / workstation, and is often 2-3 hours long.  Four main principles of contextual inquiry are:

1.  *Context:*  see workplace and what happens  
2.  *Partnership:*  user and developer collaborate  
3.  *Interpretation:*  observations interpreted by user and developer together  
4.  *Focus:*  project focus to help understand what to look for

***2.  Work Modeling:***  In interpretation sessions, models are drawn from the observations.  Five models are:

* **Work flow model:** the people, communication and coordination
* **Sequence model:**  detailed work steps to achieve a goal
* **Artifact model:**  the physical 'things' created to do the work
* **Cultural model:**  constraints on the system from organizational culture
* **Physical model:**  physical structure of the work, e.g. office layout

***3.  Consolidation:***  each contextual inquiry (one for each user / developer pair) results in a set of models, which need to be consolidated into one view of 'the work'

* **Affinity Diagram:**  organizes interpretation session notes into common structures and themes
  + Categories arise from the data
  + Diagram is built through induction
* Work models consolidated into one of each type

***Participatory Design:***(Scandinavian background) emphasizes social and organizational aspects - based on study, model-building and analysis of new and potential future systems.  Aspects to user involvement include:

* Who will represent the user community?  Interaction may need to be assisted by a facilitator
* Shared representations
* Co-design using simple tools such as paper or video scenarios
* Designers and users communicate about proposed designs
* Cooperative evaluation such as assessment of prototypes

Benefits of Participatory Design:  “Computer-based systems that are poorly suited to how people actually work impose cost not only on the organization in terms of low productivity but also on the people who work with them. Studies of work in computer-intensive workplaces have pointed to a host of serious problems that can be caused by job design that is insensitive to the nature of the work being performed, or to the needs of human beings in an automated workplace.”  [Kuhn, S. in Bringing Design to Software, 1996]

PICTIVE:  Plastic Interface for Collaborative Technology Initiatives through Video Exploration:  Intended to empower users to act as full participants in design

Materials used are:

* Low-fidelity office items such as pens, paper, sticky notes
* Collection of (plastic) design objects for screen and window layouts

Equipment required:

* Shared design surface, e.g. table
* Video recording equipment

Before a PICTIVE session:

* Users generate scenarios of use
* Developers produce design elements for the design session

A PICTIVE session has four parts:

* Stakeholders all introduce themselves
* Brief tutorials about areas represented in the session (optional
* Brainstorming of ideas for the design
* Walkthrough of the design and summary of decisions made

CARD:  Collaborative Analysis of Requirements and Design - Similar to PICTIVE but at a higher level of abstraction: explores work flow not detailed screen design

* Uses playing cards with pictures of computers and screen dumps  
  Similar structure to the session as for PICTIVE
* PICTIVE and CARD can be used together to give complementary views of a design

Summary:

* User involvement helps manage users’ expectations & feelings of ownership
* A user-centered approach has three main elements: early focus on users, empirical measurement and iterative design
* Ethnography is useful for understanding work, but can be difficult to use in design
* Coherence and Contextual Design support the use of ethnographic data in design
* Participative design involves users taking an active part in design decisions  
  CARD and PICTIVE are example techniques

Exercise:

This exercise is to be done in pairs.

Consider a website application for booking theatre or cinema tickets online

(a) Think about how you would design such a site, and sketch out some ideas  
(b) Run a CARD session with a colleague acting as a ‘user’ to map out the functional flow of the website  
(c) Ask your colleague to produce some scenarios of how the system may be used. Meanwhile, prepare some ‘empty’ templates for a PICTIVE session for this system, using paper, sticky notes and pens  
(d) Run a PICTIVE session to develop the online booking system collaboratively, using PICTIVE.

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 10:  Introducing Evaluation

What, Why and When to Evaluate...

Goals of this chapter:

* discuss how developers cope with real-world constraints
* explain the concepts and terms used to discuss evaluation
* examine how different techniques are used at different stages of development

What to evaluate:

iterative design and evaluation is a continuous process that examines:  early ideas for a conceptual model; early prototypes of the new system; later, more complete prototypes

Designers need to check that they understand users' requirements

Why evaluate:

- because user experience can be extremely important for a product's success

- because the cycle of **design** and **testing** is the only validated methodology in existence that will consistently produce successful results

***Five good reasons for investing in user testing (Tognazzini):***  
- fixed problems before the product is shipped  
- the team can concentrate on real problems  
- Engineers code instead of debating  
- Reduced time to market  
- sell without bothering to release patches

When to evaluate:

- when it's brand new:  develop markups of the product to elicit reactions from the potential users  
- upgrades:  compare user performance & attitude w/ previous versions

Two main types of Evaluation:  ***Formative Evaluation and Summative Evaluation***

1. ***Formative Evaluation:****done at different stages of development to check that the product meets users needs*
2. ***Summative Evaluation:****assess the quality of a finished product*

it is best to evaluate throughout design- from the first descriptions / sketches, all the way to the final product

Design proceeds through iterative cycles of design / test / re-design - where evaluation is a key ingredient for a successful design

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 11:  An Evaluation Framework

Goals of this chapter:

* Explain key evaluation concepts and terms
* Describe the evaluation paradigms & techniques used in interaction design
* Discuss the conceptual, practical and ethical issues that must be considered when planning evaluations
* Introduce the DECIDE framework

User studies focus on how people behave, *in their natural environments*, or *in the laboratory,* with *old technologies and with new ones*.

Evaluation Paradigms:

* any kind of evaluation is guided by a set of beliefs, implicitly or explicitly.  These beliefs are often supported by a theory.  The beliefs and the methods associated with them are called ***evaluation paradigms***.
* There are four main evaluation paradigms discussed:
  + **Quick & Dirty:**
    - this is the common practice by which designers get feedback from users, to confirm that they are in line with users' needs and are liked.  Emphasis is on fast input to the design process rather than carefully documented findings.
  + **Usability Testing:**
    - records typical users' performance on typical tasks in controlled settings.  As the users perform the tasks they are watched and recorded on video, with keypresses logged.  Data is used to calculate performance times, identify errors and help explain why the users did what they did.
    - User satisfaction questionnaires & interviews are used to gather users' opinions.
  + **Field Studies:**
    - done in natural settings- aims to understand what users do naturally and how technology impacts them.  This can help to:
      * identify opportunities for new technology
      * determine design requirements
      * decide how best to introduce new technology
      * evaluate technology in use
  + **Predictive Evaluation:**
    - experts apply their knowledge of typical users to predict usability problems
    - can involve heuristics, and theoretically based models
    - users do not need to be present, and this method is cheap and quick to do
    - GOMS / KLM / Fitt's Law???

Evaluation Techniques:

Observing users ([ch. 12](http://www.sharritt.com/CISHCIExam/preece.html" \l "ch12))

* can help identify needs, which can lead to new products
* can help evaluate prototypes
* ways in which to record observations:  notes, audio, video, interaction logs
* challenges:  how can we observe others without disturbing them?  how will we analyze the data gathered?

Asking users their opinions ([ch. 13](http://www.sharritt.com/CISHCIExam/preece.html" \l "ch13))

* Techniques:  Interviews and Questionnaires
* Asking experts their opinions is inexpensive and quick

Testing users' performance ([ch. 14](http://www.sharritt.com/CISHCIExam/preece.html" \l "ch14))

* ways to measure user performance to compare 2 or more designs
* modeling users' task performance to predict the efficacy and problems of a user interface
* some of these techniques:  GOMS and the keystroke model

DECIDE:  A framework to guide evaluation

**D**etermine the goals the evaluation addresses (what, who, why...)

**E**xplore the specific questions to be addressed (break down into subquestions- e.g. consumers' attitudes, security, interface, reputation of system, trust, adequate access)

**C**hoose the evaluation paradigm and techniques to answer the questions (paradigm selected determines the techniques used.  Practical issues, ethical issues, and tradeoffs must be considered)

**I**dentify the practical issues (select users, stay on budget, stay on schedule, find evaluators, select equipment)

**D**ecide how to deal with the ethical issues (informed consent form, privacy / confidentiality, and let the participants know the goals of the study, what will happen to the findings, privacy of information, etc..)

**E**valuate, interpret, and present the data (what data to collect, how to analyze and present depends on the paradigm used.  Need to consider reliability, validity, biases, scope and ecological validity)

Pilot Studies:

* these are a small trial of the main study.  They can help make sure the study is viable.
* pilot studies check that you can conduct the procedure, and that your interview skills, questionnaire questions, and experiment procedure works properly
* they can identify potential problems and is useful for ironing out problems before doing the main study
* can use colleagues if you can't spare real users

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 12:  Observing Users

The goals of this chapter:

* how to select appropriate observation techniques
* how to do observation
* how to collect, analyze the data, and present findings from it

Goals and questions:  provide a focus for observation

Paradigms:  guide all evaluation studies

What & When to observe:

observing is useful at any time during product development  
- early on in design:  helps designers understand users' needs  
- later on:  to examine whether the developing prototype meets users' needs

evaluators can be an onlooker, a participant or an ethnographer

Approaches to Observation:

* ***Quick & Dirty observation:***  can occur anywhere, anytime.  Good for immediate feedback.  Evaluators can temporarily join a group to observe.
* ***Observation in Usability Testing:***  video and interaction logs (keystrokes, mouse clicks, conversations).  Observers can watch through one-way mirror or on remote TV screen.
* ***Observation in Field Studies:***  can observe on any of four levels:  complete participation, marginal participation, observers who also participate, observers who do not participate

How to observe:

* *In controlled environments:*  decide where to set up equipment, test the equipment, provide an informed consent form.  Cons:  observer doesn't know what users are thinking
  + to overcome this:  the ***think-aloud technique*** allows us to observe the users' problem solving strategies by requiring them to say out loud everything they are thinking and trying to do
* *In the field:*  different framework to structure and focus observation and data collection activity (the person, the place, the thing; who, what, where, why, how; space, actors, activities, objectives, acts, events, goals, feelings)
* *Participant observation and ethnography:*  follows the same guidelines as above, but the observer must be accepted into the group (gain their trust by offering them the results, etc).  ***Ethnographic study allows multiple interpretations of reality:  it is interpretivist.****Often data collection and analysis happen simultaneously in ethnographic study, through participant observation and interviews by immersing the observer in the users' culture.*

Data collection:

* Notes + still camera
* Audio recording + still camera
  + helps provide a visual record, but transcribing the data can be a pain in the butt
* Video
  + gets audio and visual data, but can be intrusive.  Also, it's easy to miss stuff (outside the camera's viewpoint).  This too can be very time consuming to process the data

Indirect Observation:  Tracking users' activities

Techniques include:

* Diaries:  good when users are scattered and unreachable in person  
  - pros:  inexpensive, require no special equipment or expertise  
  - are good for long term studies
* Interaction logging (key presses, mouse / device movements)  
  - used in usability testing  
  - typically synchronized with video and audio logs  
  - good for being unobtrusive and for logging large amounts of data automatically

Analyzing, Interpreting and Presenting the Data

*There are three types of data:*

1. **qualitative analysis to tell a story**
   1. review data and identify key themes
   2. record themes in a coherent / flexible form
   3. recode the data and time of data analysis session
   4. check your understanding with the people you observe
   5. iterate process until your story represents what you observed
   6. report findings to development team (oral or written report)

analyzing and reporting ethnographic data (from participant observation, interviews, artifacts:

1. look for key events
2. look for patterns of behavior (in various situations and players)
3. compare sources of data
4. report findings
5. **qualitative analysis for categorization**
   1. look for incident patterns
   2. analyze data into categories (content analysis)
   3. determine the content categories reliability and inter-research reliability ratings
   4. analyze discourse (conversation analysis)
6. **quantitative data analysis**
   1. video data collected in usability labs
   2. annotated
   3. recording to calculate performance times
   4. analyze statistically

**Finally, feed the findings back into design!**

Summary

* it is very valuable to be able to observe users in the field to see how technology is used in context
* this observation can confirm ideas and offer possibilities to explore new design ideas
* the way that observational data is collected and analyzed depends on the paradigm in which it is used:  quick & dirty, user testing, or field studies.
* various amounts of control, intervention, and involvement are possible when observing:  
    ***lab studies / usability testing    <------------>    participant observation / ethnography***  
  - on one end, lab studies offer a strongly controlled environment with little evaluator involvement  
  - on the other end participant observation and ethnography require deeper involvement with users and understanding of context
* Diaries and data-logging techniques provide a way to track user activity without intruding

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 13:  Asking Users and Experts

Interviews and Questionnaires are used in "quick and dirty" evaluation, in usability testing, and in field studies to ask about *facts, behavior, beliefs and attitudes.*

Interviews

* 4 types:  can be **open-ended (unstructured), structured, semi-structured, or group interviews**
  + ***unstructured:***  no script, not replicable
  + ***structured:***  tightly scripted like a questionnaire
  + ***semi-structured:***  guided by script but open for deeper exploration if desired
  + ***group:***  often a 'focus group' of 3-10 people, consensus reached on questions
  + *data analysis on structured interviews is like a questionnaire, while unstructured like participant observation*
* 2 types of questions:  open and closed.  Closed require interviewee to choose between options, open allow for a free-range response.

Questionnaires

* can use:  yes / no; Likert scale; semantic scale; open-ended responses on questions
* to make a good questionnaire, provide a clear purpose statement, plan the questions, decide if phrases have a positive or negative connotation, pilot test your questions, and decide how the data will be analyzed
* to reach a large amount of people:  guarantee anonymity, offer online (large base / instant results & often instant data analysis)
* Be careful- online questionnaires don't prevent people from answering multiple times, and can have a low response rate
* *data analysis includes identifying trends, using simple statistics, making use of percentages & bar graphs*

Heuristic Evaluation

* when cost of accessing users is too much, use expert inspections or heuristics to analyze usability, etc.
* For example:  ***Nielsen's heuristics***
  + *Visibility of system status*
  + *Match between system and real world*
  + *User control and freedom*
  + *Consistency and standards*
  + *Help users recognize, diagnose and recover from errors*
  + *Error prevention*
  + *Recognition rather than recall*
  + *Flexibility and efficiency of use*
  + *Aesthetic an minimalist design*
  + *Help and documentation*
* ***Heuristic evaluation is referred to as 'discount evaluation' because evidence supports the fact that 5 evaluators can detect 75% of the usability problems!  It is inexpensive and quick.***
* Three stages of heuristic evaluation:  
  1.  briefing session telling experts what to do  
  2.  evaluation period of 1-2 hours where each expert works separately to get a feel for the product, and then to focus on specific features  
  3.  a debriefing session where the experts work together to prioritize problems
* Pros:  quick and cheap evaluation, diagnosis of problems
* Cons:  can be hard to find experts, important problems can get missed, often trivial problems get identified
* Note:  different combinations and types of heuristics are needed to evaluate different types of applications and products.

Walkthroughs

* walkthroughs are an alternative to heuristic evaluation and involve 'walking through' a task with the system an noting problematic usability features.  often these don't involve users.
* Two main types:  **Cognitive Walkthroughs** and **Pluralistic Walkthroughs**
* **Cognitive Walkthroughs** simulate a users' problem solving process at each step in the human-computer dialog, checking to see if the users' goals and memory for actions can be assumed to lead to the next correct action.
  + focus on ease of learning
  + designer presents an aspect of the design and usage scenarios
  + one of many experts walk through the design prototype with the scenario
  + expert is told the assumptions about user population, context of use, task details
  + Expert is guided by 3 questions:  
    - Will the correct action be sufficiently evident to the user?  
    - Will the user notice that the correct action is available?  
    - Will the user associate and interpret the response from the action correctly?
  + As the experts work through the scenario they note problems (focus on users' problem in details)
* **Pluralistic Walkthroughs** are where developers and usability experts work together to step through scenarios and discussing usability issues associated with dialog elements involved in the scenario steps.  Each group of experts are asked to assume the role of typical users.

1.  Scenarios are developed as a series of hard-copy screens representing a single path through the interface  
2.  The scenarios are presented to the group of evaluators  
3.  When everyone writes down actions, the panelists discuss the actions for the review  
4.  The panel moves on to the next round of screens

* + Pluralistic Walkthroughs produce a set of quantitative data which lends itself well to participatory design practices by involving a multidisciplinary team in which users / users' tasks play a key role
* Walkthroughs are very focused and are therefore suitable for evaluating small parts of systems

Summary

- Interviews can be structured, semi-structured, or unstructured.  Structured and semi-structured are designed to be replicated.  Questions can be open or closed (format)

- Focus groups are a type of group interview.

- Questionnaires are a cheap and easy way to reach large numbers of people.

- typically 5 experts can find 75% of the usability problems.

- heuristic evaluation is cheaper and more flexible than user testing.

- User testing and heuristic evaluation often reveal different usability problems.

- pluralistic and cognitive walkthroughs are focused and good for evaluating a small part of the interface.

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 14:  Testing and Modeling Users

A central part of Interaction design is user testing.

Usability testing uses a combination of techniques, including user testing and user satisfaction questionnaires.  User testing is of central concern.

The end of this chapter talks about GOMS (Goals, Operators, Methods, Selection rules), KLM (Keystroke Level Model) and Fitt's Law.

User Testing

* User testing is applied experimentation in which developers check that the system being developed is usable by the intended user population for their tasks.
* User testing tests typical users, measuring their typical task time, and the number and type of errors are recorded
* Can consist of completion time, observational data, answers to questionnaires, answers from Interviews, and keystroke logs.
* User testing is a systematic approach to evaluating user performance to inform or improve usability design
* Usually there are few participants (5-10), but can be 1-2 in "quick and dirty" for quick feedback
* Typically we record:  task completion time; task completion after being away from the product; number / types of errors; errors per unit of time; number of navigations to help manuals; number of users making a particular error; number of users completing a task successfully

Using the DECIDE Framework for User Testing

* Determine the goals & Explore the questions
* Choose the paradigm and techniques
* Identify the practical issues - design typical tasks (typical tasks, typical users, testing conditions, how to run the test, contingency plans if users take too long) and record what's mentioned above
* Deal with ethical issues (make an informed consent form)
* Evaluate, analyze and present the data

Using Experiments for Usability Evaluation

* can have one (control condition vs. experimental condition) or 2+ (test multiple conditions, so break users into groups) independent variables.
* Random participant allocation not always best.  Can attempt to group users by expertise, then balance them across conditions.  However, this can cause problems if users are not assessed properly / exactly equal (ordering effects)

|  |  |  |
| --- | --- | --- |
| ***Design*** | **Advantages** | **Disadvantages** |
| *Different Participants* | No order effects | Many participants needed. Individual differences between participants is a problem.  Can be offset to some extent by randomly assigning to groups. |
| *Same participants* | Eliminates individual differences between experimental conditions. | Need to counterbalance to avoid ordering effects. |
| *Matched participants* | Same as different participants, but the effects of individual differences are reduced. | Can never be sure that subjects are matched across variables. |

Predictive Models

* Predictive models provide a way of evaluating products or designs **without directly involving users.**
* The usefulness of predictive models is limited to **systems with predictable tasks** (answering machines, etc.)
* Predictive models are based on **expert behavior**

**1.  GOMS:  Goals, Operators, Methods, Selection rules (Card, et. al 1983)  
-**[**see Carroll Ch. 4 for more information**](http://www.sharritt.com/CISHCIExam/carroll.html#ch4)

* GOMS aims to model knowledge and cognitive processes when users interact with a system.
* It is good for evaluating efficiency between two ways of doing things.
* This is the most well-known predictive modeling technique
* See example, p. 450

**Goals:**  the state the user wants to achieve  
**Operators:**  the cognitive processes and physical actions performed to attain those goals (decide which search engine to use, think up a keyword.  The goal is obtained; an OPERATOR is executed)  
**Methods:**  learned procedures for accomplishing the goals, steps to do so (drag mouse over field, type in keywords, press 'Go' button)  
**Selection rules:**  determine which method to select when there is more than one available

**2.  The Keystroke Level Model (KLM)  (Card et. al, 1983)**

* differs from GOMS in that it **provides actual numerical predictions of user performance**
* **KLM**is good for **comparing task times between two different strategies**
* can come up with average times to do something from empirical studies of actual user performance
* this model allows for predictions to be made about how long it takes an expert user to perform a task.  The predicted time is computed by describing the sequence of actions involved in the task and summing their approximate times (looked up from empirical data):  
      T(execute) = Tk + Tp + + Th + Td + Tm + Tr  
      Operators:  K (keystroke); P (pointing); H (homing); D (drawing); M (mental preparation); R (system response time)
* for more explanation, see [summary from Carroll book (Ch. 4)](http://www.sharritt.com/CISHCIExam/carroll.html#ch4)

Pros and Cons of GOMS:

Pros:  allows for comparative analysis for different interfaces or computer systems relatively easily  
    outcome:  counter-intuitive, help make decisions about the effectiveness of new products

Cons:  not often used for evaluation purposes because of its **highly limited scope**    only good for predicting **expert performance**, and **error is not modeled** (average users not predicted)  
    many unpredictable factors come into play

A Con of Predictive Models:  they can make predictions about predictable behavior, but it is difficult to use them as a way of evaluating how systems will be used in the real world.  They are only useful for comparing the efficiency of different methods in completing a short, simple task.

**3.  Fitt's Law  (Paul Fitts, 1954)**

* used for *evaluating systems*where the **time to physically locate an object** is critical to the task at hand
* the law predicts that the time to point to an object is a function of the distance from the target object and the object's size (derived from Shannon:  amplitude and noise)
* originally the law spoke of the speed and accuracy when moving towards an object on a display.  In Interaction Design, it is used to describe the **time it takes to point at a target**, based on the **size of the object and the distance to the object**.
* The further away the object, and the smaller the size, the longer it takes to locate it and point to it.
* Fitt's law predicts that **the most quickly accessed targets on any computer display are the four corners of the screen**
* Fitt's law is ***useful for evaluating systems for which the time to locate an object is critical*** - such as handheld devices like mobile phones.

Summary

* user testing is at the CORE of usability testing
* GOMS, Keystroke level model & Fitt's law are **used to predict expert, error free performance**
* ***Predictive models are used to evaluate systems with limited, clearly-defined functionality & predictable tasks***

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

Chapter 15:  Design and Evaluation in the Real World:  Communicators and Advisory Systems

Key Issues:

* user-centered approaches to interaction design involve iterative cycles of **design-evaluate-redesign** as development progresses from initial ideas through various prototypes to the final product
* a multitude of questions, concerns and decisions come up throughout system design projects (to be good at system design:  must be good at working through the cycles and be good at multitasking, decision-making, team work and firefighting)

From Requirements to Design:

* which design cycle to use?
* which combination of methods to use when designing and evaluating your product?
* what happens when the product being developed is confidential and there are no users available to test it?
* how many users should be involved in tests?
* what should we do with the evaluation findings?
* how much should we expect from users?

Case study:  Nokia's mobile communicator

- used ethnographic research and did scenarios and task models to get requirements  
- (method) followed participatory design- involved users throughout  
- (method) used interface metaphors  
- (method) followed with frequent low fidelity prototypes based on alternative designs / immediate evaluation (yielded invaluable insight to designers)  
- wrote usage scenarios (high level descriptions of device in use)  
- user testing (usability tests):  did summative testing before release (at end, not throughout [formative]) and questionnaires after release

Case study:  redesign of a telephone response information system (TRIS)

- current system was very hard to use- a deep menu system over the phone- users can't remember it without cues  
- GOMS / KLM used to show how interface supported users' tasks  
- heuristic evaluation used as an alternative method for showing usability problems (expert review of system)  
- methods complemented each other and showed benefit of doing a re-design

Key points:

* **Design involves trade-offs that can limit choices but can also result in exciting design challenges**
* Prototypes can be used for a variety of purposes throughout development, including for marketing presentations and evaluations
* The design space for making upgrades to existing systems is limited by the design decisions previous system.  The design space for new products is much greater.
* Cycles of rapid prototyping and evaluation allow designers to examine alternatives in a short time
* Simulations are useful when evaluating systems used by large numbers of people when it is not feasible for them to work on the system directly
* **Piecing together evidence from data from a variety of sources can provide a rich picture of usability problems, why they occur, and possible ways of fixing them.**

**(**[**top**](http://www.sharritt.com/CISHCIExam/preece.html#top)**)**

**Jump To:** [1](http://www.sharritt.com/CISHCIExam/preece.html#ch1)  [2](http://www.sharritt.com/CISHCIExam/preece.html#ch2)  [3](http://www.sharritt.com/CISHCIExam/preece.html#ch3)  [4](http://www.sharritt.com/CISHCIExam/preece.html#ch4)  [5](http://www.sharritt.com/CISHCIExam/preece.html#ch5)  [6](http://www.sharritt.com/CISHCIExam/preece.html#ch6)  [7](http://www.sharritt.com/CISHCIExam/preece.html#ch7)  [8](http://www.sharritt.com/CISHCIExam/preece.html#ch8)  [9](http://www.sharritt.com/CISHCIExam/preece.html#ch9)  [10](http://www.sharritt.com/CISHCIExam/preece.html#ch10)  [11](http://www.sharritt.com/CISHCIExam/preece.html#ch11)  [12](http://www.sharritt.com/CISHCIExam/preece.html#ch12)  [13](http://www.sharritt.com/CISHCIExam/preece.html#ch13)  [14](http://www.sharritt.com/CISHCIExam/preece.html#ch14)  [15](http://www.sharritt.com/CISHCIExam/preece.html#ch15)

HCI Models, Theories and Frameworks: Toward a Multidisciplinary Science  
*by John Carroll*

**Carroll, J. M. (Ed.) (2003). HCI Models, Theories and Frameworks: Toward a Multidisciplinary Science.** San Francisco: Morgan Kaufmann publishers.   *ISBN*: 1-55860-808-7.   
This collection of tutorial articles is an appropriate survey for the graduate level student. [DS]

**Jump To:** [1](http://www.sharritt.com/CISHCIExam/carroll.html#ch1)  [2](http://www.sharritt.com/CISHCIExam/carroll.html#ch2)  [3](http://www.sharritt.com/CISHCIExam/carroll.html#ch3)  [4](http://www.sharritt.com/CISHCIExam/carroll.html#ch4)  [5](http://www.sharritt.com/CISHCIExam/carroll.html#ch5)  [6](http://www.sharritt.com/CISHCIExam/carroll.html#ch6)  [7](http://www.sharritt.com/CISHCIExam/carroll.html#ch7)  [8](http://www.sharritt.com/CISHCIExam/carroll.html#ch8)  [9](http://www.sharritt.com/CISHCIExam/carroll.html#ch9)  [10](http://www.sharritt.com/CISHCIExam/carroll.html#ch10)  [11](http://www.sharritt.com/CISHCIExam/carroll.html#ch11)  [12](http://www.sharritt.com/CISHCIExam/carroll.html#ch12)  [13](http://www.sharritt.com/CISHCIExam/carroll.html#ch13)  [14](http://www.sharritt.com/CISHCIExam/carroll.html#ch14)  [15](http://www.sharritt.com/CISHCIExam/carroll.html#ch15)

[Ch. 1:  Introduction](http://www.sharritt.com/CISHCIExam/carroll.html#ch1)  
[Ch. 2:  Design as Applied Perception (visual senses)](http://www.sharritt.com/CISHCIExam/carroll.html#ch2)  
[Ch. 3:  Predictive vs. Descriptive Models, Fitt's Law](http://www.sharritt.com/CISHCIExam/carroll.html#ch3)  
[Ch. 4:  GOMS, KLM, Advanced / Modified GOMS](http://www.sharritt.com/CISHCIExam/carroll.html#ch4)  
[Ch. 5:  Cognitive Dimensions of Notations Framework](http://www.sharritt.com/CISHCIExam/carroll.html#ch5)  
[Ch. 6:  Users' Mental Models](http://www.sharritt.com/CISHCIExam/carroll.html#ch6)  
[Ch. 7:  Information Foraging Theory, Optimal Foraging Theory, Scatter/Gather (evolutionary)](http://www.sharritt.com/CISHCIExam/carroll.html#ch7)  
[Ch. 8:  Collaborative Technologies / Distributed Cognition](http://www.sharritt.com/CISHCIExam/carroll.html#ch8)  
[Ch. 9:  Cognitive Work Analysis (CWA)](http://www.sharritt.com/CISHCIExam/carroll.html#ch9)  
[Ch. 10:  Clark's Common Ground Theory (as used in CMC)](http://www.sharritt.com/CISHCIExam/carroll.html#ch10)  
[Ch. 11:  Activity Theory](http://www.sharritt.com/CISHCIExam/carroll.html#ch11)  
[Ch. 12:  CSCW Research (Psychological foundations)](http://www.sharritt.com/CISHCIExam/carroll.html#ch12)  
[Ch. 13:  Ethnography, Situated Action, Ethnomethodology](http://www.sharritt.com/CISHCIExam/carroll.html#ch13)  
[Ch. 14:  Computational Formalisms](http://www.sharritt.com/CISHCIExam/carroll.html#ch14)  
[Ch. 15:  Design Rationale as a Theory](http://www.sharritt.com/CISHCIExam/carroll.html#ch15)

Chapter 1:  Introduction

HCI lies between social and behavioral sciences; computer and information technologies.  It is the fastest growing CS field, and needs a multidisciplinary approach to keep flourishing.

HCI practitioners:  analyze and design user interfaces, integrate technology to support human activity

* in the past:  WIMP interface, voice / video interfaces
* today:  better input / display devices (mobile computing), information visualization (digital libraries), navigation techniques (virtual environments)

Initially, HCI brought cognitive science theories to focus on software development

Activity theory brought into HCI (Marxist foundations), complemented cog. sci.

While a multidisciplinary approach helps HCI theories, a tradeoff is the fragmentation of HCI (it's very difficult to make sense of the vast, diverse science of HCI; to synthesize a comprehensive, coherent methodological framework)

*Goal of the book:  To 'survey' the many approaches to HCI; compare / contrast them  
 - each chapter is a different approach / method*

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 2:  Design as Applied Perception

Much of human intelligence can be characterized as our ability to recognize patterns

Vision dominates as the primary sense:  engaging 50% of the cortex and more than 70% of all our sensory receptors

Why we care:  we have a *fundamental assumption* that there is such an entity as ***the human visual system***

*display design -*map data into a visual form that is matched to our perceptual capabilities

* the same perceptual mechanisms that enable us to perceive the world enable us to perceive information patterns in computer displays

In Cognition, humans share a similar muscular / skeletal structure, but have a highly adaptable neural structure, which allows for a high degree of adaptation (humans are highly adaptive machines)

***Information Psychophysics*** - concerned with how we can see simple elementary information patterns (paths in graphs, clusters of items, correlated variables, etc.)

* *Theory of Affordances (J. J. Gibson) -*a highly relevant theory, saying that ***affordances*** are physical properties of the environment that are *directly perceived*(before HCI / computers)
* Since a computer screen doesn't have a physical environment, we use a ***metaphor*** - which says that a good user interface is *similar* to a real-world situation
* Information Psychophysics can be seen as a component of a larger set of cognitive models, such as ACT-R (Adaptive Control of Thought - Rational [Anderson]) and EPIC (Executive-Process/Interactive Control [Kieras]) which focus on the **cognitive processing of sensory information**

**3 Stage Model of the Visual System:**  this happens very fast - our brain's process when we see stuff

* ***Stage 1:  Early Vision:  our visual image is analyzed into color, motion and shape***
  + involves eye optics, photoreceptors
  + use *trichromatic system* to see colors- segment into Red / Green / Blue
  + use *luminance* to see detail / contrast
  + the theory of color vision is that we separate color into red-green, blue-yellow, and black-white *opponent color channels*
  + *Opponent colors* explain why red, green, yellow, blue, white and black are special colors in all societies, and why we should use these colors first if we need to color-code things
  + Since these colors can't convey patterns, we use *luminance contrast* to show detail
  + ***preattentive processing theory*** - uses early stage processing, and tells us how to design things so that they POP-OUT of a cluttered display*.  (Color, motion, texture, stereoscopic depth can be preattentively perceived, and understanding these limitations is critical in making displays that can be rapidly interpreted)*
* ***Stage 2:  Pattern Perception:  we split what we see into segments:  a 2-D image, pick up edges, etc.***
  + we can display data using patterns that are easy to perceive, which facilitates problem solving
  + GESTALT psychologists (Wesheimer, Koffka, Kohler) first theorized of pattern perception:
    - *Proximity* - entities that are close together are perceptually grouped
    - *Good Continuity -*smooth continuous lines are more readily perceived
    - *Symmetry -*symmetric objects are more readily perceived
    - *Similarity -*similar objects are perceptually grouped
    - *Common Fate -*objects that move together are perceptually grouped
    - *Common Region -*(added later) - objects in enclosed spaces are grouped
    - *Connectedness* - (added later) - objects connected by continuous contours are perceived as related
  + In HCI, visual displays can support creative thinking and problem solving
* ***Stage 3:  we perceive objects (infer based on stages 1 & 2) out of what we see***
  + ***working memory*** - core of modern models of cognitive processing
    - there are several components of working memory:  visual, verbal, etc...
  + ***Structured object perception (Biederman)*** - we perceive objects as composed of simple linked 3-D solid shape primitives, called GEONs.  There is considerable difference between high-level (object perception, color is secondary) and low-level vision (color perception, etc)

In the above system, **feedback loops** can modify what we see.  Higher stages can feedback to lower stages to modify what we see.  Lower stages are more robust, while higher stages involve more inference making.

In the above system, culture makes it difficult to design by visual patterning since aspects of displays owe their value to cultural factors (R, G, Y, B are different colors in every culture).  Some cultural aspects are so hard-wired that there's no way to design around them.

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 3:  Motor Behavior Models for HCI

Why model motor behaviors?  Because we need to match movement limits, capabilities, and potentials of humans - to input devices and interaction techniques - on computing systems

A model is a simplification of reality that we can use to design, evaluate and provide a basis of understanding a complex behavior of complex artifacts.

Models lie somewhere in a continuum:    ***analogy / metaphor  <-- ? --> mathematical equations***

* ***Predictive Models:***  on the mathematical end   
  (a.k.a. *Engineering Models* or *Performance Models*, such as **Fitt's Law**)
* **Descriptive Models:**on the metaphorical end of the spectrum   
  (*Descriptive*, such as **Guiard's model** - *provide a framework / context to think about a problem*)

***More on Predictive Models:***

allow human performance to be analyzed analytically, avoiding time-consuming experiments

predictions are a-priori, allowing for hypothetical exploration of a design scenario

Hick-Hyman Law:  an example predictive model, predicts human response time by an equation

**Fitt's Law:**  a predictive model for human movement (measures accuracy and amplitude of human movement; computes an index of performance to compare efficiency of different devices / interfaces)

* drawn from Shannon's theory - uses electronic noise (accuracy) and electronic signal (amplitude) to quantify a movement task's *index of difficulty (ID)* and predict *movement time (MT)* to complete a task
* *ID / MT = Index of Performance (IP)*; now known as Throughput (TP)
* **The goal of Fitt's law:***determine which devices / interaction techniques are most efficient by comparing performance measures*

*An example:*  keystroking on mobile phones:  ones with multi-tap vs. ones with one-key disambiguation (guesses most common word based on key presses):  second is faster based on Fitt's law, although more error prone

* Fitt's law isn't appropriate for complex tasks, like measuring typing speed on a qwerty keyboard (complex task with 2 hands)

**Keystroke-Level Model (KLM):**  a predictive model for predicting time to do a task by dividing into sub-tasks:

K = Keystroking  
P = Pointing  
H = Homing  
D = Drawing  
M = plus 1 Mental operator  
R = plus 1 system Response operator  
  
*TExecutive = tK + tP + tH + tD + tM – tR*

Other Predictive Models:  GOMS (see Ch. 4), Programmable User Model (PUM:  Young, Green, Simon 1989)

**More on Descriptive Models:**

Descriptive Models provide a framework / context for thinking about a problem

Example:  **Key-Action Model (KAM):**  divides keys on a keyboard into 3 categories:  *symbol* keys, *modifier* keys, and *executive* keys (simple model, organizes keys into categories)

**Three-State Model of Graphical Input (Buxton):**  Descriptive model, says computer pointing devices follow a state transition diagram of 3 states:  *out of range, tracking, and dragging*(good for analyzing different types of pointing devices [touchpad, trackpad, mouse, etc]) - resulted in redesign of touchpad's to be pressure sensitive, so drag state could be induced easily

**Guiard's Model of Bimanual Skill:**  studies *between-hand division of labor in everyday tasks* - see chart of descriptive model on p. 41

* Features of non-preferred hand:  leads the preferred hand, sets spatial frame of reference for preferred hand, performs coarse movements
* Features of preferred hand:  follows non-preferred hand, works within spatial frame of reference (set by non-preferred hand), performs fine movements

Buxton & Myers found that people prefer to use two hands when not instructed, and this resulted in lower times in task completion.  This had little insight into hand use, but was one of the first papers in HCI.

Another study (Gibson) of keyboards found that they are right-hand biased (power keys on right side), but the Desktop as a whole is left-handed bias:  lefties can cash in on a time savings, because having the mouse to the left of the keyboard allows for the right hand to use the keypad / power keys WHILE the left hand operated the mouse (don't have to stop, switch between the two)

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 4:  Information Processing and Skilled Behavior

**GOMS** is the primary focus of this chapter (*Goals, Operators, Methods, and Selection* rules:  a way to describe a task and calculate the time to complete it)

* **Goals:**  The users' goals:  what does the user want to accomplish with the software, and by when?
* **Operators:**  The actions that the software allows the user to take.  Generally it is a command such as a button press, menu selection, or a direct-manipulation action.
* **Methods:**  the sequence of sub-goals and operators that can be used to accomplish a goal:  such as the sequence of moves necessary to cut and paste text to move it from one location to another.
* **Selection Rules:**  Giving the user the personal "choice" (selection) to accomplish the goal in those times when there is more than one method available to accomplish the goal or certain task

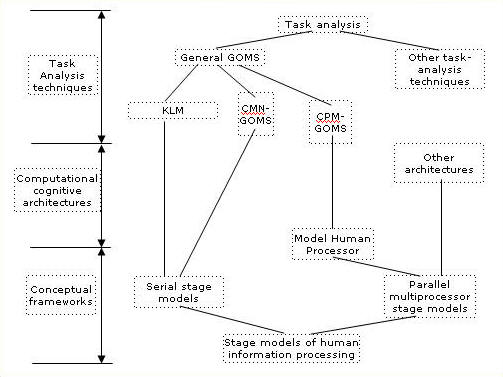
GOMS applies to situations in which users will be expected to perform tasks that they already have mastered.  It does not work for tasks being done by novices that are trying out a new interface design.  The knowledge gathered by GOMS reflects what a skilled person (expert) will do in a seemingly unpredictable situation.

GOMS works for single-user, active systems where the system changes in unexpected ways or other people -participate in accomplishing the task.  It has been shown to work very well in analyzing user-paced, passive systems.

GOMS can be used quantitatively and qualitatively.

* Quantitatively:  gives good predictions of performance time and relative learning time
* Qualitatively:  can help design training programs, help systems and inefficient systems following an analysis of the results of study

GOMS as a task analysis technique:  stemming from conceptual frameworks on human information processing (Carroll, p. 65):



*Conceptual frameworks* - informally stated assumptions about the structure of human cognition

serial stage models - we process information in a serial sequence  
parallel multiprocessor stage models - we process information like parallel computing

*Computational Cognitive Architectures* - also called unified theories of cognition - are proposals to make cognition explicit enough to run as a computer program

**The Model Human Processor (Card et. al, 1983):**  ties cognitive science and engineering models:  represents human cognition as a parallel computer.  The parallel processing being done:

* Cognitive (thinking / thought process)
* Perceptual (seeing / what we observe)
* Motor (doing / what physical movements)

In the diagram above, the bottom is more theoretical while the top is more practical.  GOMS is a way designed to make human information processing something that can be measured practically.

*Task Analysis Techniques* - these map out the system in terms of goals, operators, methods and selection rules.  There are 3 restrictions to what GOMS models can be used for:

1. there must be a way to do the task in question
2. the task must be able to be 'routinely done' by a skilled expert
3. the GOMS analyst must start with a list of top-level tasks or user goals (provided from external sources outside of GOMS)

**GOMS Models:**

1. **KLM:  Keystroke Level Model (Card, et. al)**
   * this is the simplest GOMS:  turns a simple action into a sequence of steps for analysis
   * Time of execution = T(execute) = Tk + Tp + Td + Tm + Tr  
     Operators:  K (keystroke); P (pointing); H (homing); D (drawing); M (mental preparation); R (system response time)
   * The main advantage is this allows for a very quick estimate of execution time with very little theoretical / conceptual baggage.  It is the most practical GOMS technique.
2. **CMN-GOMS:  Card-Moran-Newell GOMS**
   * named this to differentiate itself from other GOMS models as they began to appear
   * based on the serial-stage model (see diagram above), not the Model Human Processor (MHP)
   * slightly more specified than general GOMS
   * Quantitatively:  predicts the operator sequence and execution time
   * Qualitatively:  focuses attention on methods to accomplish goals.  Similar methods are easy to see.  Unusually short or long methods jump out and can spur design ideas.
   * While the KLM has no explicit goals, CMN-GOMS does.  The output of CMN-GOMS is in program form, so it is executable.
3. **CPM-GOMS:  Cognitive, Perceptual, Motor GOMS (Bonnie John)**
   * based on the MHP (Model Human Processor) - Cognitive, Perceptual and Motor Operators can run in PARALLEL (subject to resource and information dependencies)
   * based on parallel multiprocessor stage model of human information processing
   * CPM also stands for Critical Path Method, since the critical path in a schedule (of parallel processes) determines the overall execution time.
   * Quantitatively:  predictions of performance times can be read off the chart of CPM-GOMS
   * Qualitatively:  analysis of what portions of a design lead to aspects of the performance are easy once the models are built
   * as with KLM, selection rules are not explicitly represented in the chart because the chart is just a trace of the predicted behavior
   * This is the only model representing parallelism, so several goals can be achieved at one time, which is characteristic of expert performance of a task

Human Information Processing

* this approach can be used to model more complex human behaviors like problem solving, learning and group interaction, all of which are critical to designing complex systems
* some HCI practitioners think of GOMS as a good tool, many think it is not (too cumbersome / time consuming)

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 5:  Notational Systems- The Cognitive Dimensions of Notations Framework

<http://homepage.ntlworld.com/greenery/workStuff/Papers/introCogDims/index.html>

Think of the point of view of the system creators (practice side, not so theoretical).

System designers try to incorporate new HCI Theories into their designs, but have learned that some are more practical than others.  They want to use theoretical knowledge, but prefer things like checklists which often aren't too theoretical.  The authors believe that **it will not be possible to deal with these new notational systems by creating new checklists.**

Especially with user testing- many of the methods that have been suggested such as **user testing** (laborious, expensive, in artificial labs)and **Predictive models** (expert-done calculations that predict time to complete a task on a finished system)are **very expensive / time consuming to use**

**HCI has generated several approaches that offer suggestions for redesign, but they focus on representation and require extensive, detailed modeling.  There is no ONE approach that addresses all types of activity that can lead to constructive suggestions for improving the system or device, that avoids details and allows for the identification of similar problems down the road.  In HCI, no one model is perfect- each one has its own limitations, which is why we should know about the cognitive dimensions framework.**

**Alan Blackwell and Thomas Green**propose a set of **cognitive dimensions framework** that will allow researchers and designers to 'talk together' (DISCOURSE) about evaluation.  It is much lighter and easier to use than the above two main types of evaluation.

**Cognitive Dimensions Framework**

**- not a method; it is a set of discussion tools; used by the designers themselves**

**- attempts to IMPROVE DISCUSSION through a SHARED VOCABULARY**

**- main process is to DISCUSS TRADEOFFS of design decisions among designers by using a SHARED VOCABULARY**

**- Cognitive Dimensions Framework suggests a handful of basic 'Notational Dimensions' in the shared vocabulary:  VISCOSITY, HIDDEN DEPENDENCIES , ABSTRACTION LEVEL, PREMATURE COMMITMENT**

**Notational Dimensions (the SHARED VOCABULARY)**- each dimension describes an aspect of an information structure that is reasonably general

* Viscosity:  resistance to change
* Visibility:  ability to view the components easily
* Premature commitment:  constraints on the order of doing things
* Hidden dependencies:  important links between entities that are not visible
* Role-expressiveness:  the purpose of an entity is readily inferred
* Error Proneness:  the notation invites mistakes and the system gives little protection
* Abstractions:  types and availability of abstraction mechanism
* Secondary notation:  extra information in means other than formal syntax
* Closeness of mapping:  closeness of representation to domain
* Consistency:  similar semantics are expressed in similar syntactic forms
* Diffuseness:  verbosity of language
* Hard Mental Operation:  high demand on cognitive resources
* Provisionality:  degree of commitment to actions or marks
* Progressive Evaluation:  work-to-date can be checked at any time

**Evaluation using the CD Framework**

There are two steps to evaluation using the cognitive dimensions framework:

1. Decide what generic activities a system is desired to support.  Each generic activity has its own requirements in terms of cognitive dimensions.
2. Scrutinize the system and determine how it lies on each dimension.  If the two profiles match, all is well.

**Tradeoffs**

What is nice about the CD framework is that it eliminates design maneuvers in which one dimension is 'traded-off' against another.  However, there are certain relationships:  such as a way to reduce viscosity is to introduce abstractions.  Abstractions might reduce viscosity and increase visibility...

**Questionnaires**

Because CD Framework is so general, it has been used to structure questionnaires to get user-feedback on certain aspects (notational dimensions) of a system.  Because the notational dimensions are very general, feedback can be very useful and may be unanticipated by the designer (a good thing).

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 6:  Users' Mental Models  
- stems from cognitive psychology  
- why?  because understanding users' mental models can enrich our understanding of the use of cognitive artifacts

Cognitive psychologists think the UMM is at the heart of understanding HCI- reasoning behind what we do  
  
However, the term is over used in so many different ways that it has lost its usefulness  
  
Still, mental models is a very important and useful construct, with many areas that can still be researched

Cognitive psychology foundations to mental models:

Idea 1:  Mental content vs. cognitive architecture:  mental models as theories

* we use 'bounded rationality' - a process of using readily-available knowledge to make a decision.  (Note not ALL available information, because we make a quick decision and don't dig too deep for info)
  + cognitive psychologists focus on the 'structure' (human information-processing architecture) of the mind more than the 'content' (focus on what an individual knows / believes) of the mind to explain bounded rationality because 'structure' is more generalizable than content (individual / incidental).
  + while most cognitive psychology theories focus on structure of the mind rather than content of the mind (to achieve generalizability).  However, in HCI, we need to focus on singular behaviors (content) as they are a staple of usability studies, etc.  For example, a software 'walkthrough' will lead to diagnoses about misleading features of an interface design.
* users use mental models of familiar things to construct mental models of unfamiliar things (predicting the behavior of an ATM is based on other, familiar computer usage)

Idea 2:  Models vs. Methods:  mental models as problem spaces

* a characterization of skill:  a collection of methods for achieving tasks
* in cog. psych, solving a problem involves searching through a mentally constructed problem space of possible states (a.k.a. mental simulation)
* as we gain skill, we know what situation calls for what method (less searching / problem solving, more routine)

Idea 3:  Models vs. Descriptions:  Mental models as Homomorphisms

* mental models are 'analog representations' (share the structure of the world it represents)
* similar to pictures, 'simple static models' represent the world, with objects / relations
* 'dynamic models' are homomorphisms (one to many relationships rather than one to one), and can resemble state-transition networks

Idea 4:  Models of Representations:  Mental models can be derived from language, perception or imagination

* mental model can result from processing language, from a perception or from the imagination

Idea 5:  Mental representations of representational artifacts

* 'goal space' is the domain of the users' goals (such as spaces and lines on a paper when we are using a word processor)
* 'device space' is the problem space that the user searches for solutions to the goal (such as characters / operations in a word processor)

Idea 6:  Mental models as computationally equivalent to external representations

* it is possible to have two or more representations of the same information / task, but these are 'informationally equivalent' if one is inferable from the other and vice versa
* two or more 'informationally equivalent' representations may or may not be 'computationally equivalent', which is the cost of accessing the information / doing the task
* a 'cognitive map' is a mental representation that allows users to navigate through an environment (computationally equivalent to an external map) through process of *internalization*

Case studies:

What are 'yoked' state spaces?

'yoked state spaces' can motivate an informal analysis of the fit between the representational capacities of a device and the purposes of a user (calendar / appointment diary example)

support the process of *internalization*- computationally equivalent mental versions of external representations

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 7:  Exploring and Finding Information

How do people forage for information?  People are ***informavores*** (George Miller):  we are organisms hungry for information about the world and themselves.  This chapter draws much from evolutionary theory.

Two main concepts of this chapter:

* ***Information-Foraging Theory***
* ***Information Scent***

Information-Foraging Theory

* a metaphor from evolutionary theory:  deals with understanding how user strategies and technologies for information seeking, gathering, and consumption are adapted to the flux of information in the environment
* built on notation of hunter / gathering

Information Scent

* concerns the user's use of environmental cues in judging information sources and navigating through information spaces

**Both ideas evolved in reaction to the technological developments associated with the growth of globally distributed, easily accessible information, and the theoretical developments associated with the growing influence of evolutionary theory in the behavioral and social sciences.**

Following the metaphor:  What do we do when we hit a**roadblock**(metaphorically like hitting a stream, where we loose our 'information scent')?  We need to search around to pick up the scent again- much of the probability of success depends on the interface design (whether it's good or not- and can lead us in the right direction:  provide clues as to which choice to choose next, eliminate bad paths, etc.)  *If we loose track of our 'scent' and accidentally pick up the wrong scent, we loose trust in the system.*  Also, how far down the wrong path (scent) will we go before we realize we are on the wrong path?

Adaptation vs. Exaptation

this chapter's approach sees users as being adaptable.  ***Users are complex adaptive agents who shape their strategies and actions to be more efficient and functional with respect to their information ecology.***

Adaptationist approaches became mainstream in the 1980s, as a reaction to ***ad-hoc*** models on human cognition (cognitive & perceptual tasks).  This is in contrast to ***mechanistic*** approaches of the time, such as the MHP (Model Human Processor, Card, et. al 1983)

Adaptationist approaches reverse-engineer the problem- examining **what** environmental problems are being solved and **why**cognitive and perceptual systems are well adapted to solving them

***Exaptation*** looks at how we as humans are capable of adapting to solve similar problems (a way of generalizing our knowledge:  we can solve similar but different problems by observing themes in the problems)

Extending the Information Foraging Theory Metaphor:

* ***exaptation of food-foraging mechanisms and strategies of information foraging:***  natural selection favored our ancestors- those with better foraging strategies had better success
* ***the economics of attention and the cost structure of information:***  a wealth of information creates a poverty of attention- we need to efficiently allocate our attention to RELEVANT information.  Information systems should evolve to provide more valuable information per unit cost.
* ***relevance of optimal-foraging theory and models:***  **optimal foraging** from study of animals- who try to get the most nutritious food with the least amount of effort.  With information, we want the richest, most relevant information in the least amount of effort

Optimal-Foraging Theory

Goal is to optimize the information we receive to be the most rich, relevant to our information needs.  Optimization models include three major components:

* Design assumptions:  specify the decision problem to be analyzed (how much time to spend processing a collection of information)
* Currency assumptions:  identify how choices are evaluated (information value = currency)
* Constraint assumptions:  limit / define the relationships among decision / currency variables (rise out of task structure / interface / knowledge of population)

In general, all tasks can be analyzed according to the value of the resource currency returned and costs incurred.  Classified as **Resource costs (cost to get it) and Opportunity costs (potential benefit of it)**

Scatter / Gather

* an interaction technique for browsing large collections of documents
* we cluster documents, then make a cluster hierarchy
* very messy looking interface on p. 170 using this- bad HCI!
* gather resources, scatter into categories

**As Designers:**  we want to design systems that provide the richest information, with the least cost to access it (easy information retrieval) - can be very useful in web design and search engines

**Current Status of IFT:**  *Internet Ecology* is being studied- looks at complex global phenomena that yield predictions on Internet usage and distribution of users over web sites

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 8:  Distributed Cognition

Almost every work situation requires someone working with other people

HCI hasn't had a way into this problem because cognitive theory tells us little about social behavior, and social science is hard to apply directly to design

Ways in which people use tools (artifacts) to support their goals is poorly understood

***Distributed Cognition*** grew out of a need to understand how information processing and problem solving could be understood as being performed across units LARGER than the individual  
- cognitive scientists (psychology) don't have to abandon their background to understand this:  we just shift from focusing on **'information in the head'** to **'information in the world'** - examine things **'in the wild'  
-**how to understand how *intelligence is manifested at the systems level* rather than the individual level

Designing Collaborative Technologies

**CSCW (Computer Supported Collaborative Work)** shifts the focus of HCI from the user to the group (multiple, codependent users and their social network)

we focus on the *problem solving* of the group as a *cognitive*problem

we focus on distributed cognition **within a context** - drawing on *actors* and other features within the environment that allow problem solving (*socially distributed cognition*)

***the goal of analysis:  describe how distributed units are coordinated by analyzing interactions among individuals, the representational media used, and the environment that the activity takes place***

shift from traditional HCI (micro-structure level) to macro-structure level (**ecological design** - a turn to the social)

this has led **ethnography** - an anthropological approach to collecting data about the problem domain - to become a central feature of CSCW (see [Ch. 13](http://www.sharritt.com/CISHCIExam/carroll.html#ch13))

**Cognition**

* *definition:  referring to all of the processes by which the sensory input is transformed, reduced, stored, recovered, and used*
* *cognitive science:  the problem solving and the organization of knowledge about the problem domain*
* The dominant paradigm for cognition, ***Information-Processing Theory***, proposes that problem spaces (the representation of the operations required for a given task) are abstract representations.  There is no reason this theory needs to be restricted to an individual - any unit performing these activities can be described as a cognitive entity
* *artifact*:  any man-made or modified objects (Cole, 1990)
* *cognitive artifact*:  a subset of artifacts - those that aid cognition (*'knowledge in the world'* - Norman; memory aids, etc.).  These can increase human potential by extending our abilities, and can also transform the task itself- which can allow the user to re-allocate resources for more efficient task completion
* with distributed cognition, this re-allocation of resources can take place across the group of individuals, for even more efficient task completion
* this has led to new theories on ***"systems perspective"*** that focus on describing the features of a system:  the people, artifacts, and the means of organizing these into a productive unit
* *external cognition*:  described by Preece as a way of externalizing information (creating and using information in the world, without performing logic operations in the head), in order to *reduce memory load*, to *simplify cognitive effort* by computational offloading onto external media, and by allowing us to *trace changes* through annotation or by using a new representational form.  This allows humans to logically process information without performing logic operations in their heads.
* *cognitive ethnography*:  analyzes the workplace to determine the role of technology and work practice in system behavior

**Distributed Cognition and Computing**

* looks to examine the role between the computer and the social group
* How is information about a domain stored (represented) in a system?
* *functional system*:  the actors and artifacts of the system (complex cognitive system / functional unit) - derived from the activity system (activity theory)
* make use of **abstraction** to describe a cognitive system - away from detailed design- focus on the general functions of the system (nonspecific:  what it DOES rather than what it IS)
* A distributed cognitive system consists of:  (see Fig. 8.2 - p. 207)  (note:  somewhat resembles MHP)
  + *sensory mechanism* - input from outside the cognitive system - passes to information processing unit
  + *action generator* - allows production of output from the cognitive system, can provide feedback to the information processor
  + *memory* - a stored representational state to order subsequent activities, receives representations from the information processor and passes back to it
  + *information processor* - receives representations from the sensory system and acts upon them:  transforms them, combines them or destroys them
* when a cognitive system is distributed over a number of individuals, cooperation is required among the individuals in order to bring their problem-solving resources into conjunction with each other.  This is crucial to their cooperative action
* a problem faced by groups in performing distributed cognition is the ability to organize a task into component parts that can be performed by individuals- and then re-assembled at the end (distribute the load fairly evenly across group members)
* we need to proactively **structure** distributed group labor so that effective work can be done
* there needs to be **effective communication and coordination** between group members (artifacts with universal meanings, etc)
* DCog vs. Ethnomethodology:  Ethnomethodology informed ethnography focuses on the ordering of the activity (coordination), but not on the work itself

**Doing DCog:  Cognitive Ethnography**

**Unit of analysis:  the functional system (individuals, artifacts, and their relations)**

**Look for information-representation transitions that result in the coordination of activity and computations:**

* examine the way that the work environment structures work practice,
* changes within the representational media,
* the interactions of the individuals with each other,
* the interactions of the individuals with system artifacts

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 9:  Cognitive Work Analysis *(sometimes called sociotechnical work analysis)*

Cognitive engineering - the analysis, modeling, design and evaluation of complex sociotechnical systems  
- first coined by Norman after Three-Mile Island crisis, helped launch the field (how to design human-machine systems so they are safer / more reliable)  
- goal was to design systems with good user interfaces, so that in *unanticipated situations*, the user would be able to safely handle the physical system

Cognitive Work Analysis (CWA) - way of analyzing human cognitive work - describes the forces that shape human cognitive work in complex real-time systems (where humans control complex physical processes).  The goal is to lead to systems that better support human-operator adaptation when operators are confronted with unanticipated variability.  CWA is a multidisciplinary approach to cognitive engineering.

Often, the human operator is very separated from the actual process he is managing, due to a poor interface.  CWA is an approach to cognitive engineering that aims to help find better ways of connecting the operator with what he is managing (physical system behind the technology) - so in unexpected situations, adaptive behavior (by the operator) will be successful.

Ecological Interface Design (EID) - subset of CWA - a set of principles to guide interface design (influenced by ecological psychology).  Said to be one of the most useful products of CWA.  Makes use of WDA and WCA (see below), as well as some ecological concepts.

formative model - an approach that describes the requirements that must be satisfied so that a system can behave in a new, desired way.  CWA is a formative approach to work analysis.

*Phases of Analysis in CWA:  Based on Figure 9.3, p.230  
 - each narrows down the action possibilities further from the previous phase*

1.  Work Domain Analysis (WDA):  find purpose and structure of work domain; often represented in abstraction-decomposition diagrams and abstraction hierarchies  
 - constraints (physical & purposive) within which activity takes place (NOT the activity itself, though)  
 - often 5 levels of abstraction:  functional purpose, abstract function, generalized function, physical function, and physical form (see Figure 9.7 on p. 242 for an example)

2.  Control Task Analysis (CTA):  find what needs to be done in the work domain so that it can be effectively controlled; often represented in maps of control task coordination, decision ladder templates  
 - continues to narrow down the 'dynamic space of functional action possibilities' by defining constraints that must be satisfied when work functions are coordinated over time and when effective control is exercised over the work domain

3.  Strategies Analysis (SA):  find ways that control tasks can be carried out; often represented in information flow maps  
 - '*HOW* control tasks can be done' is analyzed (we don't care by whom)  
 - focus on general classes of strategies and their intrinsic demands

4.  Social-Organizational Analysis (SOA):  find who carries out work and how it is structured; often represented by annotating the information flow maps of #3  
 - focus on the division and coordination of work (the *content (information passed between actors)*), and the social organization of the workplace (the *form (behavioral protocol of communications)*)

5.  Worker competencies analysis (WCA):  find the kinds of mental processing supported; often represented by skill-based, rule-based, and knowledge-based behavior models

CWA was influenced by *systems thinking* and *ecological psychology.*  Both emphasize that the human-environment system needs to be the unit of analysis, with the environment being a primary unit of analysis in an actor's goal-oriented behavior.

*Systems Theory:*- the whole is more than the sum of its parts (study the whole environment)  
- study the relation between parts, not the properties of the parts  
- study *cybernetics - open vs. closed systems* - and how outside disturbances affect the system

*Ecological Psychology:*- CWA wants to build 'ecological information systems' that can be operated closer to the ease which the natural world is navigated  
- EID (ecological interface design) is an approach to building interfaces using principles of CWA  
- "*ecological science* rests on the principle that systems in the natural and social world have evolved to exploit environmental regularities" - Rosson & Carroll  
- *ecological psychology* is the study of the way organisms perceive and respond to regularities in information  
- **key concept #1:**  environments and information should be described in terms that reveal their functional significance for an actor rather than being described in objective actor-free terms  
- **key concept #2:**the affordances of an object are the possibilities for action with that object, from the point of view of a particular actor  
- **key concept #3:**the actor uses direct perception, which proposes that certain information meaningful to an actor is automatically picked up from the visual array

*The CWA System Life Cycle (SLC):*  requirements definition, modeling and simulation, tender evaluation, operator training, system upgrade, and system decommissioning.  This area may see much development in the next decade.

At the end of the chapter, many case studies are presented.  They fall into 2 categories:  display design, and evaluations of human-system integration.

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 10:  Common Ground in CMC:  Clark's Theory of Language Use

HCI has come to encompass technologies that mediate human-human communication (chat, etc)

Production + Comprehension = Communication

**Clark's theory of common ground:  we use existing common ground to develop further common ground and hence to communicate effectively**

* ***Different types of common ground:***
  + a proposition is common ground if:  all the people conversing know the proposition, and they all know that everyone else knows the proposition.
  + *communal common ground:*  "we both speak English.  we are professionals.  we live in the same town."
  + *personal common ground achieved before the conversation:*  "our joint purpose is to sign off on the plan."
  + *personal common ground developed during the conversation:*  "the door on the bedroom that faces south has to be moved.  We can see the bedroom facing south on the plan.  The plan can go to the builder."
* ***Formalizes collaborative activity as "joint action"***
  + *language is a joint action involving two or more people*
* ***Describes the process by which common ground is developed through joint action***
  + *face to face conversation uses common ground to reduce the effort required to communicate*
  + *face to face conversation develops common ground*
  + *face to face conversation involves more than just words (non-verbal comm, etc.)*
  + *face to face conversation is a joint action*
  + *face to face conversation is "basic" (basis for understanding language behavior)*

**Grounding:**  process of making sure that another person sufficiently understands you.  If not- use grounding.  Often facial expressions (non-verbal behavior) or questioning serves to notify us if other person needs more information (grounding)

Summary:  Clark's theory of language use is applicable to CMC (computer mediated communication).  The usage / application of this theory to designing systems will be more evident as time goes on- and will likely be very useful to new systems supporting video conferencing, asynchronous and synchronous communication, etc.  However, a very useful set of guidelines based on this has yet to be developed.

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 11:  Activity Theory

**History / Foundation for Activity Theory:**

A basic perspective on HCI that came from Scandinavian roots in the early 1980's

an 'Action Research Approach' that focused on active cooperation between researchers and 'those being researched' - researchers entered into an active commitment to improve the situation of those they researched

roots in social psychology, industrial sociology and critical psychology.  Influence by the introduction of the personal computer, moving away from the mainframe.

There were many problems with existing HCI and system design at the time.  Much of the system design being done at the time consisted of heavy task analysis, with consideration of a generic novice user working alone.  A new theoretical foundation was needed!

**Foundations of Activity Theory:**

* analysis and design for a particular work practice with concern for qualifications, work environment, division of work, and so on;
* analysis and design with focus on actual use and the complexity of multi-user activity.  In particular, the notion of the artifact as mediator of human activity is essential;
* focus on the development of expertise and of use in general;
* active user participation in design, and focus on use as part of design.

Activity theory shares with ecological psychology the attempt to move away from the separation between human cognition and human action, and an interest in actual material conditions of human acting.  Activity theory differs by adding a notion of motivation.  Activity theory values hierarchical analysis, task analysis, etc.  Activity theory takes place on all levels at the same time, not in sequence (see below).  Activity theory moves away from the generic user.

**Unit of analysis in Activity Theory:**  motivated activity.  This is mediated by socially produced artifacts (tools, language, representations, etc.)

Activity is mediated through a computer or other tool

Activity theory does not assume a boundary between internal representations and external representations, like cognitive science.  Activity theory has a basic feature of unifying consciousness and activity.

**Development:**  The most distinct feature of Activity Theory is development.  When compared to other materialist accounts in computer science, the focus is on development.

***Nardi's Five Principles of Activity Theory:***

* object-orientedness
* hierarchical structure of activity
* internalization and externalization
* mediation
* development

*Mediation is folded in with each of the other four principles, resulting in four categories of concerns:*

* means / ends
* environment
* learning / cognition / articulation
* development

**Focus and Focus Shifts:**

Activity theory starts with a perspective / point of view - which yields the objects to work with.  However, the focus shifts that indicate the dynamics of the situation are the main point of concern in the analysis.

([see Index page for more on Focus Shifts](http://www.sharritt.com/CISHCIExam/index.html))

**Summary:**

**Nardi suggests that activity theory is a powerful descriptive tool rather than a predictive theory.**  Carroll somewhat disagrees- because in this chapter concrete techniques were presented that show how it can be used to focus on computer-mediated activity (a.k.a. HCI)

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 12:  Applying Social Psychology Theory to Group Work

              Study \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
                   /                      /|  
        **Approach**/                      / |  
                 /                      /  |  
         Build  /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/   |  
                |                      |   |  
     Technical  |                      |   |  
Infrastructure  |                      |   |  
**F**|      ***Variations***      |   |  
  Architecture**O**|       ***in CSCW***       |   |  
**C**|      ***research***      |   |  
   Application**U**|                      |   |  
**S**|                      |   |  
          Task  |                      |   /  
                |                      |  /    
        People  |                      | /     
                |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|/  
                    **Group Size**Individual ... Group ... Society

CSCW (Computer Supported Collaborative Work) is a subfield of HCI that aims to build tools to help group work, learning and playing.  It examines how groups incorporate tools into their routines and the impact of CMC on group processes and outcomes.

Two main goals of CSCW:

* to support distributed groups (distributed groups are typically not as effective as collocated members)
* to make collocated groups more effective (CSCW systems are designed to ameliorate some of Steiner's process loss

Elements of an input-process-output model of groups:



Production Outcomes:  multidimensional

* groups do better than individuals because of **aggregation** (different people contribute to the groups with unique resources
* groups do better than individuals because of **synergy** (the increase in effectiveness that comes about from joint action and cooperation
* group maintenance and member support:  groups need to have the capability to work together in the future (group maintenance) and support the needs of its individual members (member support)

Inputs:  both inputs and processes that group members use will determine the success of the group

* *personnel:*  diversity can be a mixed blessing - more viewpoints, but more arguments
* *task:*  McGrath states that tasks can be:  generative (e.g. brainstorming); intellective (e.g. correct answers); problem-solving (e.g. open ended ?'s).  Groups in knowledge work tend to be as good as the second best person and follow a 'trust supported ' heuristic... a process of aggregation and synergy
* *technologies:*  groups will be more effective if they have more qualified personnel and appropriate technology (applies to collocated and distributed groups)

Interaction Process:  the way group members interact can directly influence group outcomes and mediate the impact of inputs on the group.

* focus is on communication (takes time away from group production) and can be characterized according to volume, content, structure and interactive features
* the right volume, content, structure and interactive features depends on the particular task
* uncertainty is a key feature that determines whether a particular technology will be appropriate
* technologies restricting communication are less acceptable if tasks are more uncertain

Process Losses

* Steiner:  being in a group degrades performance from what the members could be producing individually
* can be caused by **mis-coordination** (production blocking, schedule conflicts, misaligned goals, etc.)
* can be caused by **reductions in motivation** (slackers, not present for goal setting, being around other slackers, social loafing)
* **production blocking** is when people cannot get work done because they are busy listening and participating with other group members
* **social loafing** is when people think their efforts are being pooled with the efforts of other group members.  People tend to work harder individually or when they realize their contributions to the group are unique, or if they like the group (it's attractive / valuable)
* **social pressure**(such as evaluation apprehension) can cause production losses.  A way to lighten this is to introduce anonymity, but this must be considered carefully (a set of trade-offs)

Anonymity:  a way to offset social pressures, but can cause social loafing (trade-off)

CSCW researchers turn to the social-science literature outside their own field, and often consult this- such as ethnographic research- than experimental social psychology

Carroll says that CSCW research has been underexploited- a lot due to mismatching goals and values between HCI/CSCW research and social psychology research

social psychology has provided a rich body of research and theory about principles of human behavior, which should be applied to the design of HCI applications- especially those supporting multiple individuals who are communicating or performing a collaborative task

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 13:  Studies of Work in HCI

Jonathan Grudin argued in 1990 that HCI was passing from the 4th stage to 5th stage:  from "a dialog with the user" to a "focus on the work setting"

This leads into the topics of this chapter, which get into ethnography, situated action, and ethnomethodology.

Ethnomethodology

* predominated by the emergent HCI concerns with the workplace
* utilizes an ethnographic / fieldwork approach
* is concerned with the analysis of work and the workplace
* ethnomethodology & conversation analysis

Many factors precipitated the adoption of CSCW:

* Lucy Suchman's "Plans and Situated Action" - move from the individual user to a computer placed in a social context.  Attacked many cognitive scientists that failed to take into account the social and cultural world.  She undermined the idea that we 'plan' what we're going to do- replacing it with an idea that action takes place within sociocultural contingencies that cannot be covered by a plan (owes much to ethnomethodology)
* Suchman demonstrated that work could be studied as part of the process of designing systems for the workplace (draws from ethnomethodology within sociology and conversation analysis)
* the Scandinavian Participatory Design movement:  led to the involvement of the workers in the design process, and emphasized:  the flexibility of work activities, and that work is 'accomplished' rather than a 'mechanical process' (Activity Theory?)

Overview:  A Paradigmatic Case

* there are a number of ways in which ethnomethodologically grounded studies of work have been applied to HCI.  One way is to analyze the impact that a system has on the work that is done in the setting into which it is introduced
* the study of work analyzes the methods through which a domain of work is organized by those who are party to it from the inside, and makes clear that modeling a workflow using formal processes as a resource only partially grasps the work that a system is meant to support.
* studying work reveals a domain of work practices and methods that are crucial for the efficient running of an organization (characterized as "hidden" work of organizations)
* often, this 'hidden' work needs to be revealed for a successful analysis and design

**Scientific Foundations**

Ethnography

* theorize work
* empirical
* labor process theory (describe work as the reproduction of capital labor- not looking at the content of work.  serves as a structural orientation)
* an Interactionist approach:  attempts to develop an understanding of work from the inside (the actual workplace)
* Studies of work within HCI and CSCW tend to employ an ethnographic approach.  (Anderson:  ethnography in HCI has really stood for "fieldwork" grounded in ethnomethodology)

Situated Action

* Suchman:  human action takes place within contingencies.  Contingencies are not able to project in detail how any one conversation may unfold before participants engage in the course of social interaction.  Social actions are situated within arrangements and interaction.
* Social action should be analyzed within the situations in which it occurs in order to find how a course of action has been put together

Ethnomethodology

* primary concern is with social order
* social order is essential:  people display their orientation to the ordered properties of actions and interaction, and use the orderly properties when carrying out their social conduct
* members' phenomena:  order is to be found; order is a matter for ordinary members of the society
* members' accomplishment:  order is the product of consensus brought about by reciprocally shared normative values and rules, where Marxist sociology has showed that order is the product of constraint where the people are subject to the power if a coercive state exercised over them by it's institutions
* **Ethnomethodology has explored the in situ production of social order through two broad domains of interest:  Conversation Analysis and Ethnomethodological Studies of work.  Both are work methodologies.**

Conversation Analysis

Turn taking in organization is organized by participants on a moment-by-moment basis.  Turn taking is an interactional phenomenon, relating to multiple participants and organized as a collaborative matter.

Conversation analysis has developed as the study of turn taking in conversation.  It underpins some of the ways to study work, such as in the development of natural language interfaces.

Ethnomethodological Studies of Work

The second major preoccupation of ethnomethodology is its interest in work.  Ethnomethodological studies of work attempt to examine domains of work in order to understand what are the particular constitutive features of work, and how in their actions with one another, people are recognizably engaged in doing their work.

Critique:

* work is not organized as 'rule following' but rather in the contingencies and improvisations of applying rules
* Suchman critiqued a workflow system based on *speech acts*(Searle)- said that conversational analysis can underscore the situated and unfolding character of conversational exchanges

Summary:

In the past 10 years, there has been a shift in HCI from the user to the social world (work setting) in systems design.  The door has been opened to study the work setting, and information being gathered from the work place will allow us to design better systems.

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 14:  Upside-Down Algorithms - Computational Formalisms & Theory

This chapter is about how to gain insight into the context of HCI design and evaluation by using theoretical concepts (computational theory) and methods (formal methods).  We must understand our raw material (the computer itself) and that drawings, sketches, etc. are part of the design process.

Formalism is about being able to represent things in such a way that the representation can be analyzed and manipulated without regard to the meaning.  An example of this is State Transition Networks.

Key Features of Formal Descriptions:

* formal analysis:  possible to ask questions about the system from the formal description
* early analysis:  analyze early by using rapid prototypes, etc
* lack of bias:  formal analysis helps to break biases
* alternative perspective:  different representations allow us to see different things about a design, providing more views of the artifact during design
* forcing design decisions:  using formal representation, the designer is forced to make user-interface decisions explicitly and communicate those decisions to the implementer

Detailed Description:

* Fitt's Law shows that no matter what the size or number of the screen buttons - that a reasonable typing speed is always faster (3x faster than mouse clicking)
* Touch screens:  large targets are better
* Per screen cost:  more items per screen the better... ???
* for small targets, small numbers of well explained items may be better

Reasons for using Formal methods in HCI:

1. to analyze the system to assess potential usability measures or problems
2. to describe the system in sufficient detail so that the implemented system is what the designer intends
3. the process of specification forces the designer to think about the system clearly and to consider issues that would be missed

Formal Modeling can be done for single user systems or for cooperative work

Case study:  flowcharts of the human-computer dialogue:  work well because they are simple formal methods.  Why:

* *useful:* addresses a real problem
* *appropriate:* no more detail than needed
* *communication:*  mini pics and clear flow are easy to walk through with client
* *complementary:*  different paradigm than implementation
* *fast pay-back:*  quicker to produce application
* *responsive:*  rapid turnaround of changes
* *reliable:*  clear code is less prone to errors
* *quality:*  easy to establish test cycle
* *maintenance:*  easy to relate bug / enhancement reports to specification and code

Summary:

There once was a time when every computer method had to be formal to be respectable.  However, formal models bred problems over time, and grew to the point where they were written off.  However, we should not write off formal methods - look at how successful UML diagramming has been.  Often using a formal model can be beneficial- especially those that projects simple in nature.

The web is an example of a rapidly growing distributed system.  Things like bookmarks and the 'back' button are stupid- it restarts the application in the middle of a process!

The problem and challenge of formal methods:  whenever we capture the complexity of the real world in formal structures, whether language, social structures, or computer systems, we are creating discrete tokens for continuous and fluid phenomena.  In doing so, we are bound to have difficulty (it is impossible to capture everything, in minute detail to a program, process, etc.).  However, it is only in doing these things that we can come to understand, to have valid discourse, and to design.  (*think of Formal Models as 'naive physics' - as simple models that can work quickly / easily, but do not accurately represent reality*)

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Chapter 15:  Design Rationale as Theory

This chapter aims to show how reflective HCI design practices (involving design-rationale documentation and analysis) can be used to:

1. closely couple theoretical concepts and methods with the designed artifacts that instantiate them
2. to more closely integrate theory application and theory development in design work
3. to more broadly integrate the insights of different technical theories

Design rationale contributes to theory development in HCI in three ways:

1. it provides a foundation for ecological science in HCI by describing the decisions and implicit causal relationships embodied in HCI artifacts
2. it provides a foundation for action science in HCI by integrating activities directed at description and understanding with those directed at design and development
3. it provides a framework for a synthetic science of HCI in which the insights and predictions of diverse technical theories can be integrated

Design Rationale

* is documentation and analysis of specific designs in use
* describes the features of a design, the intended and possible use of those features, and the potential consequences of the use for people and their tasks
* this involves observing or hypothesizing scenarios of user interaction, and describing their underlying design tradeoffs
* helps to make theory more applicable by codifying the terms and relations of the application domain, and grounding them in design tradeoffs and decisions

Applying theory in HCI design involves mapping concepts across domain boundaries, and directing descriptions and analysis to prescriptions for intervention.

TAF:  Task-Artifact Framework

* design rationale is created to guide and understand the impacts of computing technologies on human behavior
* Tasks analysis is expressed as *scenarios of use*.  The scenarios include the activity context, the actors' motivations, actions and reactions during the episode
* *Claims analysis* produces a causal analysis of the actors' experience, enumerating the features of a system-in-use that are hypothesized to have upsides and downsides for the actors.
* The design rationale for a system is built up through analysis of multiple scenarios, often over incremental versions, leading to a network of overlapping claims.
* Book exemplifies these through MOOsburg example - p. 434

Design Rationale:  Three scientific foundations:

1. Ecological science:  rests on the principle that systems in the natural and social world have evolved to exploit environmental regularities.
   * HCI can be developed as an ecological science at three levels:  
     1.  taxonomic science  
     2.  design science  
     3.  evolutionary science
2. Action science:  a principle to research that closely couples the development of knowledge and the application of that knowledge.  Integrates the traditional scientific objectives of analysis and explanation with the engineering objective of melioration.
3. Synthetic science:  the design rationale that surfaced during a design project can be grounded in existing scientific theory, or can instantiate predictions that would extend existing theory

Detailed Description:

* the most important step in constructing a design rationale is to identify a collection of typical and critical scenarios of user interaction
* Methods for identifying scenarios:
  + carry out the work that a system is intended to support
  + carry out the work that collaborative workshops with users envisioning alternative ways
  + draw on prior work or case-study reports of other design work
  + instantiate general types of interaction scenarios in the current domain
  + transform existing scenarios
* Methods to recognize claims (tradeoffs) in scenarios
  + Text analysis:  raise questions about the actions, events, goals, and experiences depicted in the scenario, and jointly analyze scenarios with users

Case Study:  MOOsburg

The design rationale associated with an interactive system can be evaluated and refined.  In the case of MOOsburg, prototypes are still being developed, analyzed, used and refined.

When a design rationale is generalized, the hypothesized causal regularities contribute to theory building.  Design rationale supports ecological science at three levels:

* supports taxonomic science by surveying and documenting causal regularities in the usage situation
* supports design-based science through abstractions that enable knowledge accumulation and application
* supports the development of evolutionary science by promoting insights and development of new features and new situations

When the generalized rationale is grounded in established scientific theory, it serves as an integrative frame within which to understand and further investigate competing or complementary concerns (synthetic science)

In general, scenarios and design rationale specify a shareable design space that can be used to raise, discuss, and arbitrate widely varying theoretical prediction and concerns.

**(**[**top**](http://www.sharritt.com/CISHCIExam/carroll.html#top)**)**

Jump to Chapter:  [1](http://www.sharritt.com/CISHCIExam/carroll.html#ch1) [2](http://www.sharritt.com/CISHCIExam/carroll.html#ch2) [3](http://www.sharritt.com/CISHCIExam/carroll.html#ch3) [4](http://www.sharritt.com/CISHCIExam/carroll.html#ch4) [5](http://www.sharritt.com/CISHCIExam/carroll.html#ch5) [6](http://www.sharritt.com/CISHCIExam/carroll.html#ch6) [7](http://www.sharritt.com/CISHCIExam/carroll.html#ch7) [8](http://www.sharritt.com/CISHCIExam/carroll.html#ch8) [9](http://www.sharritt.com/CISHCIExam/carroll.html#ch9) [10](http://www.sharritt.com/CISHCIExam/carroll.html#ch10) [11](http://www.sharritt.com/CISHCIExam/carroll.html#ch11) [12](http://www.sharritt.com/CISHCIExam/carroll.html#ch12) [13](http://www.sharritt.com/CISHCIExam/carroll.html#ch13) [14](http://www.sharritt.com/CISHCIExam/carroll.html#ch14) [15](http://www.sharritt.com/CISHCIExam/carroll.html#ch15)