



## Activity Design

---

*Goal: work from problems and opportunities of problem domain to envision new activities*



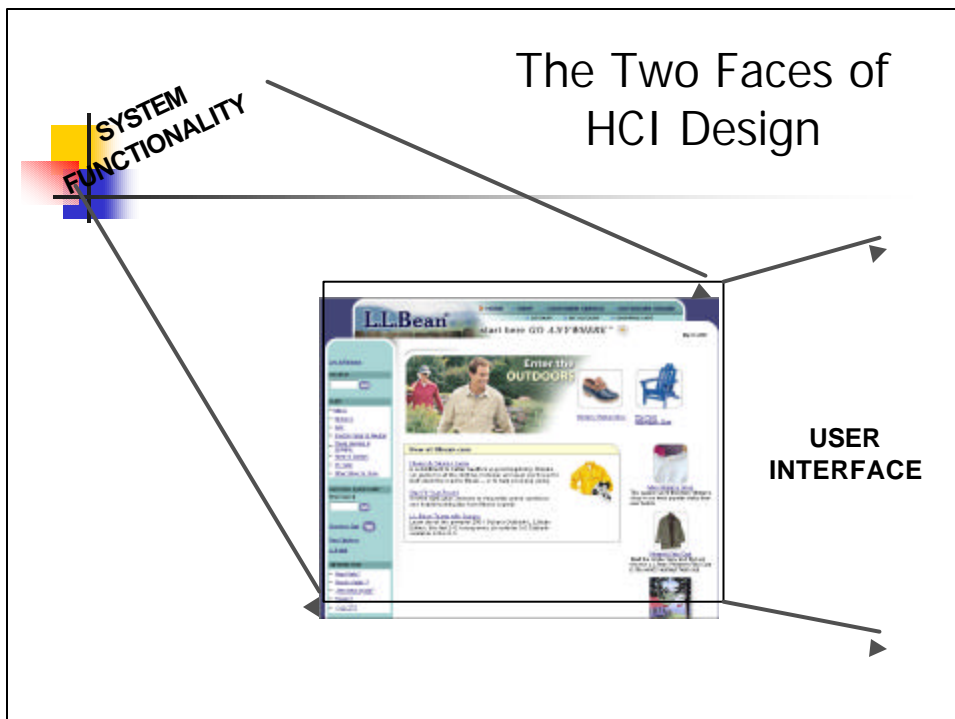
## An HCIC Example

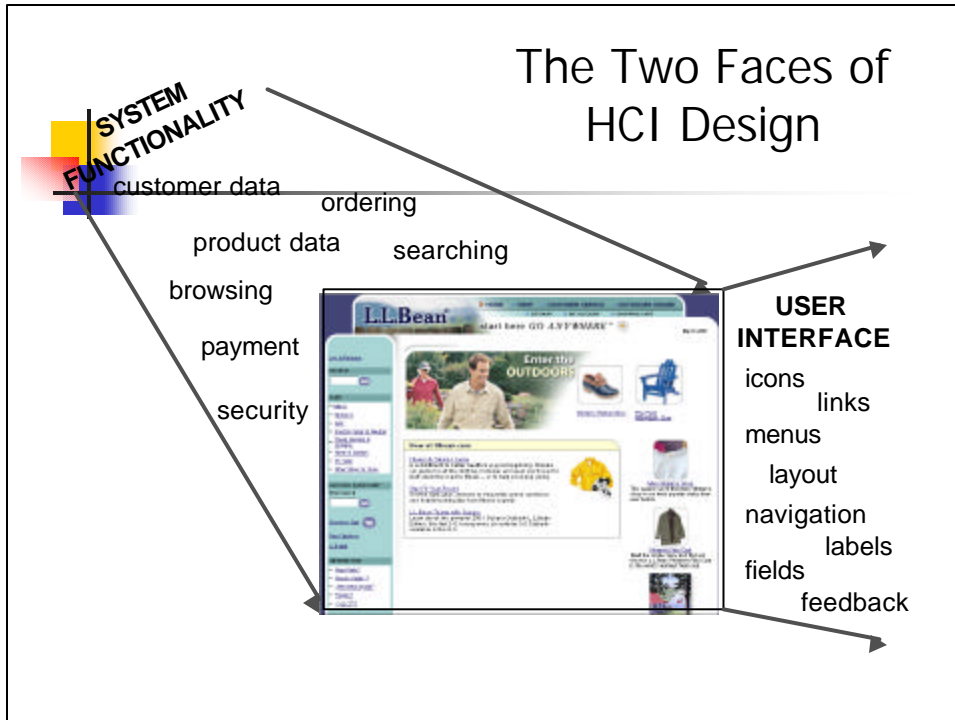
---

- Goal: design a universal remote control
- Measure use of remote controls and functionalities
- Mathematically determine which remote and which function might be next
- Automatically perform it with a touch of a button

# From Requirements to Design

- Requirements analysis sets the scene
- Design transforms people's activities
  - New technology, new tasks, new experiences
  - And the cycle continues...





- ## Activity Design
- Emphasizes broad scope of what is being designed
  - Establishes and maintains *usage context*
  - Also referred to as conceptual design or task-level design



## Why System Functionality First?

---

- Designers can focus on what a system will do
- Postpone the “how” question until later
- Hard to analyze UI needs without knowing what a system will do
- UI difficulties can destroy a system’s usability



## Envisioning New Activities

---

*Three activity design concerns to keep in mind:*

- Effectiveness: Designing tasks that meet real needs
- Comprehension: Designing concepts and services that your users can predict, understand
- Satisfaction: Designing tasks that are motivating and lead to feelings of accomplishment, satisfaction



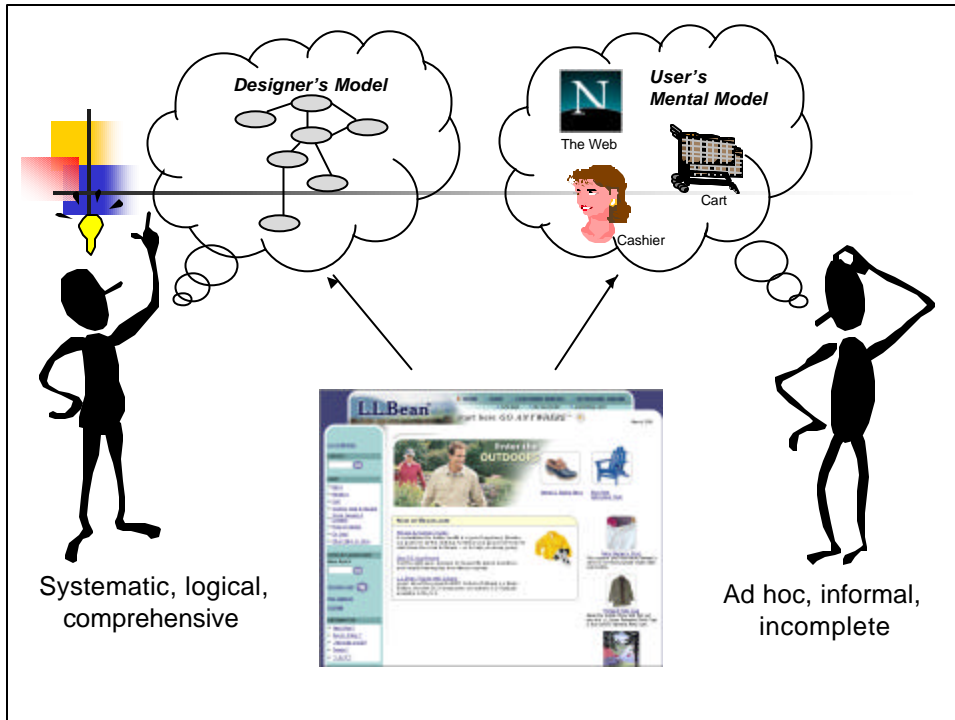
## Designing for Effectiveness

- Innovation is good, but how much is too much?
  - Build on what is already working well
  - Engage stakeholders in *cooperative design*
- What parts of a task to support via technology?
  - Leverage other aspects of the work context, both people and things (*distributed cognition*)
- Balance tendency toward general solutions with the needs of specific tasks
  - Predict and support exceptions, provide special cases for common or critical tasks



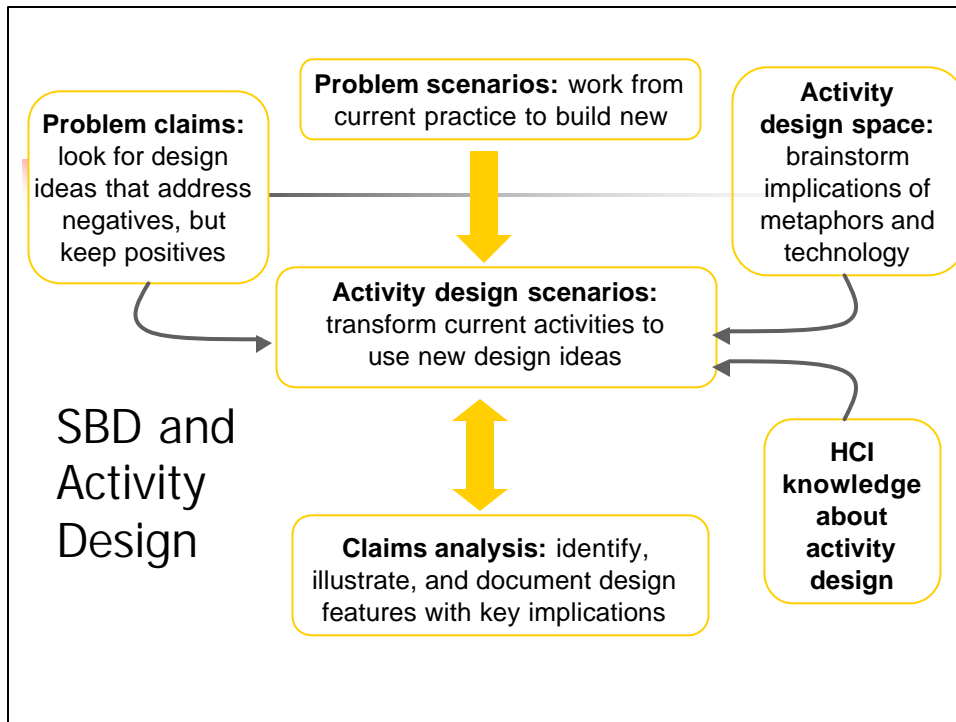
## Designing for Comprehension

- Cannot directly observe comprehension
  - Must rely on users' behaviors, reactions, comments
  - Make inferences about their *mental models*
- Metaphors play a crucial role in this
  - Designers explore metaphors to get new ideas
  - Users evoke metaphors to understand new concepts
- Try to leverage users' existing knowledge
  - Anticipate and support analogical reasoning
  - But look for ways to "break" current understandings



## Designing for Satisfaction

- Automate tedious tasks, but try not to remove sources of reward or accomplishment
  - Carefully examine sources of reward, maintain or enhance opportunities for feelings of achievement
  - Use the computer to make tasks more personal, more stimulating, more "fun"
- Balance the needs of individuals with those of the groups they work with
  - The people who do the most "work" when using a system may not be those who get the most "benefit"



## Refining an Activity Design

- Ongoing claims analysis of activity scenarios
  - Capture key ideas, begin to build *design rationale*
  - Document problems to address during UI design
- Participatory design
  - Brainstorming sessions with stakeholders
  - Share rough ideas, get them to elaborate (metaphors can be very useful here as well)
- Consistency and coherence
  - Reuse actors and objects to increase coherence
  - Complement with ongoing “what if?” reasoning to expand and test the overall design