Center for LifeLong Learning & Design
University of Colorado at Boulder

Wisdom is not the product of schooling
but the lifelong attempt to acquire it.
- Albert Einstein

Critiquing

Gerhard Fischer

AI Course, Nov 13, 1996

Overview

- why critiquing
- examples
- critiquing process
- assessment

Why Critiquing?

- **Don Norman:**
  “Real Learning: The way we learn is trying something, doing it and getting stuck. In order to learn, we really have to be stuck, and when we’re stuck we are ready for the critical piece of information. The same piece of information that made no impact at a lecture makes a dramatic impact when we’re ready for it.”

- **support reflection-in-action**
  the designer shapes the situation in accordance with his initial appreciation of it construction

  the situation “talks back” with the help of the critics

  in answer to the situations “back-talk”, the designer reflects-in-action on the construction of the problem argumentation

- **humans settle on plateaus of suboptimal behavior**

Rationale for Critiquing Systems

- **claim**: as people take on more jobs that are more complex or more comprehensive, they need help accomplishing unfamiliar tasks that are part of an expanded job — e.g.: multi-media is a good example (charts, color, ...)

- **Kosslyn (in “Elements of Graph Design”, p 2):**
  - one reason for the abundance of bad graphs is the proliferation of low-cost microcomputers and “business graphics” packages, which often seduce the user into producing flashy, but muddled display

  - the ease of creating charts and graphs is a major selling point for personal computers, one rarely hears anything about the utility of the displays the machines produce

- **Travis (in “Effective Color Displays”):**
  - the standard IBM PC can now display 256 K colors and a Sun workstation can display 16.8 million — hardware is no longer a limiting factor to use color

  - *but*: when color is used inappropriately it can be very counter productive and few software designers have much experience with the use of color

Critiquing

- **critiquing** = presenting a reasoned opinion about a user’s product or action

- critics make the constructed artifact **“talk back”** to the users (beyond the “back-talk” provided by the materials)

- critics should be **embedded** into domain-oriented design environments

- **critiquing process:**
  - goal acquisition
  - product analysis
  - critiquing strategies (when, how, and where)

- **classes of critics:**
  - educational and/versus performance: primary objective is learning and/versus better product
  - negative and/versus positive

What is Critiquing?

- **exploiting the true power of computational media**
  - paper: passive — e.g.: style guides, design rationale systems (see Web Style manual: http://info.med.yale.edu/caim/StyleManual_Top.HTML)
  - computational media: active — critiquing, constraints, simulation, making argumentation serve design, contextualizing information to the task at hand, embedded critiquing

- **role distributions**
  - in our approach most of the time: human designs and computer critiques
  - examples of computer designs and human critiques: Unix directory trees (the computer “knows” or can compute the information structure)

- **increase the back-talk of a situation**
  - how is failure or inadequacy of the form perceived in a design?
  - Rittel: “Buildings to not speak for themselves”
  - critics volunteer information

Examples

- spelling and grammar

- Lisp-Critic
  - all Lisp program could be critiqued
  - no knowledge about the problem to be solved (the macro example; compare to technical editor)

- Voice Dialog Design:
  - critiquing from multiple perspective
  - end-user control over intrusiveness

- the Fischer-Technik trucks

- critiquing at
  - the tool level (Lisp-Critic, spelling checker)
  - critiquing at the domain level (kitchen, VDDE, lunar habitat design)

- embedded critiquing
  - specific critics (left-handed, very short person)
  - interpretive critics (resale versus personal)

“Cond”-Rules in the Lisp-Critic

- **notation:**

"?"	--->	exactly one s-expression
"?*"	--->	arbitrary s-expressions

- **examples:**

```
(rule cond-erase-t-nil
    (cond ?*x (t nil)) ==> (cond ?*x) safe (people machine))

(rule cond-drop-after-t-condition
    (cond ?*x (t ?*y) ?z ?*u) ==> (cond ?*x (t ?*y)) safe (people machine))

(rule cond-to-and-1
    (cond (?condition ?action)) ==> (and ?condition ?action)
    safe (machine people))

(rule cond-with-one-arg
    (cond (?a)) ==> ?a safe (machine people))
```

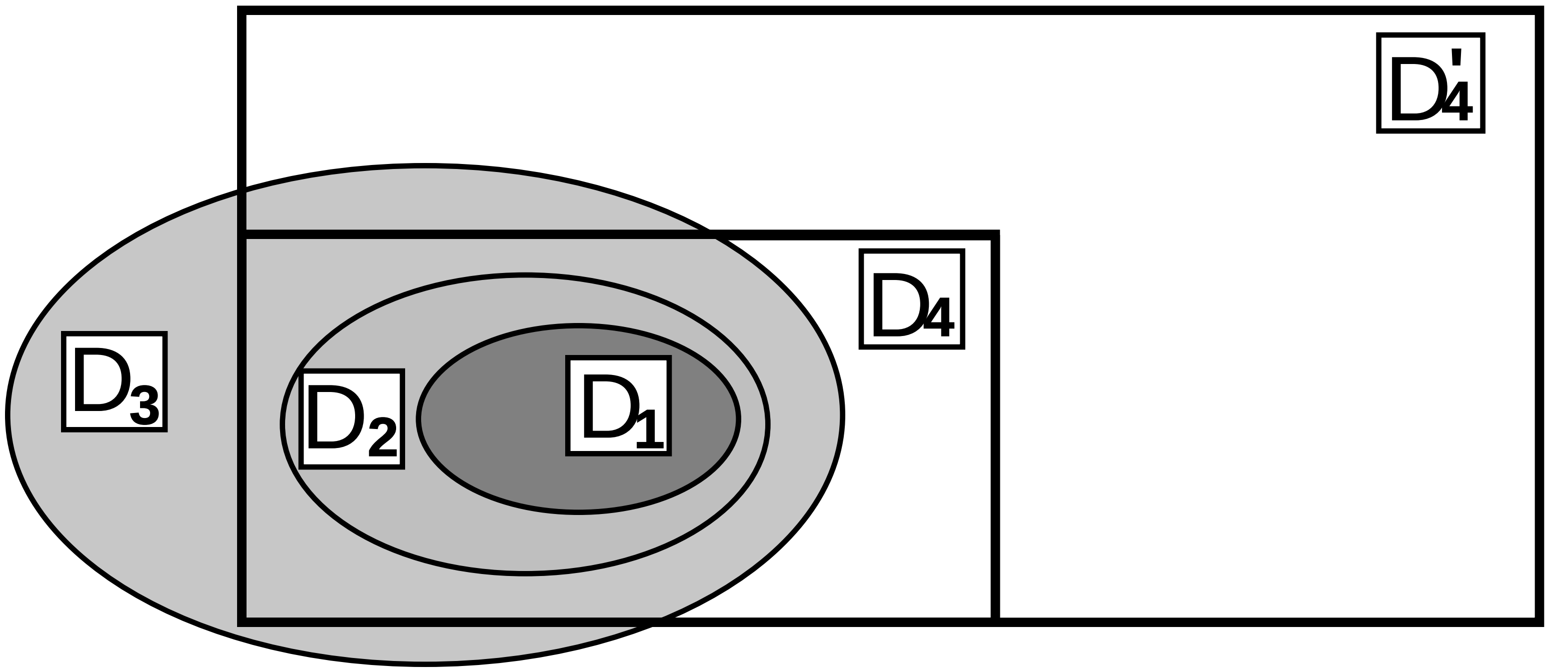
- **machine / people-flag:**

machine	---->	machine efficiency
people	---->	cognitive efficiency

Problems with Large Information Spaces and High Functionality Computer Systems

- users do not have **well-formed goals**
- users do not know about the **existence** of facilities provided by a system
- users do not know how to **access** the facilities
- users do not know **when** and **how** to use the facilities
- users do not understand the **results** that system facilities produce for them
- users cannot combine, adapt, and modify facilities according to their **specific** needs

Levels of Users' Knowledge About a System's Information Spaces



Taxonomy of Intelligent Support Systems

- **Type of Relationship:**
  - tutor
  - advisor
  - critic
  - assistant
  - consultant
  - agent

- **Type of Activity:**
  - teach
  - guide
  - visualize
  - explain
  - constrain
  - criticize
  - argue

Tutoring versus Critiquing

- **Tutoring:**
  - this is what I think you should do
  - problem of coverage is important
  - user has little control -- learning paths are determined by the tutor
  - a sequence of increasing complex microworlds can be constructed a priori

- **Critiquing:**
  - this is what I think of what you have done
  - users must be competent in the subject domain being critiqued
  - learning on demand / contextualized tutoring
  - critic system needs to infer to which microworld the user belongs

Goal Acquisition

- **domain knowledge**
  - generic — e.g., Lisp programs, kitchens, networks, voice dialog systems
  - what is a domain (residential versus commercial kitchens, kitchens for disabled persons)

- **goal knowledge**
  - specific about an individual product
  - inferred from partial specifications and partial constructions

Product Analysis

- **differential critiquing:**
  - system generates its own solution
  - compares it with the user's solution and points out the difference

- **analytical critiquing:**
  - system checks the product with respect to predefined features and effects

Critiquing Strategies

- control the presentation component of a critic
    - interrupt user’s work; separate window; flag
    - disable critics (in case where one disagrees)

- what aspects should be critiqued
    - educational critiques
    - performance critics

- negative versus positive critics

- how and when to intervene ---> intrusiveness
    - active critics: immediately, after one action, after a semantic unit
    - passive critics: critiquing is initiated by the users

- ground critiquing strategies in a user model (in addition to a task model)

Motivation and Associated Learning Strategies

- critiquing lets learners see for themselves the usefulness of new knowledge for actual problem situations; users are informed
  - when they are getting into trouble
  - when they are missing important information
  - when they come up with suboptimal solutions
- most of our critic rules state what one may not do; this makes for greater freedom of choice than if the rules were prescriptive
  - "You must not do X!" leaves open a whole range of possibilities in terms of what one may in fact do
  - "You must do X!" reduces the range of possibilities to the scope of X itself
- *unasked-for help* breeds incompetence and is often seen as an intrusion

Critiquing in the Context of Potential Benefits and Impact of Computer on Mistakes / Breakdowns

- allow us to make mistakes more quickly

- allow us to make mistakes in safe environments (undo commands, flight simulators)

- help us in recognizing mistakes / breakdowns
  - at the tool / medium level
  - at the content level

- contextualize information to the task at hand

- make the *important* invisible visible

Assessment

- **amateur designers:**
  - “breakdowns” provided learning opportunities
  - system helped improve their designs
  - learning on demand took place

- **professional designers:**
  - used it in a “spelling checker mode”
  - “breakdowns” provided opportunities to argue and to disagree
  - “there are no experts anymore” ---> e.g., kitchens for people in wheel chairs or blind people, multi-media design

The Most Frequently Asked Questions About Critiquing

- isn’t it annoying to be constantly criticized?
- if the system can critique it, why does it not do it itself?
- critic versus constraints (constraints for building codes and safety rules, critiques for functional and aesthetic principles)?
- consistency of a critiquing systems?
- does a critiquing system need to be complete?
- how much knowledge can be captured in critics?
- critics for whom: amateur and/or skilled designers?
- do critics hinder or enhance creativity?

Future Work — Issues for Investigation

- differences in performance if the system is used with and without critics, catalog, and simulation component?
- integrate constraints (e.g., for building codes)
- trade-offs between running the system in a mode
  - to prevent problems to occur (constraints)
  - to let designers get in trouble
- intervention strategies (displaying enough information versus disrupting the work process)?
- does “making information relevant to the task at hand” prevent *serendipity*?
- when are designers willing to suspend the construction process to access relevant information?
- when will designers/users challenge or extend the knowledge represented in the system? ---> end-user modifiability