

Action

Part I: Cooper & Reinmann

MAKING INTERFACES INVISIBLE

Flow and the Interface

- Flow (Csikszentmihalyi)

- When people are able to concentrate wholeheartedly on an activity, they lose awareness of peripheral problems and distractions.
- Software interaction should promote and enhance flow, rather than potentially breaking it and including flow-disturbing behavior.
- The interface is the greatest threat!



Trouble with Interface

- **An interface is**
 - ...an artifact, not directly related to the goals of the user.
- **The best interface is**
 - ...no interface at all!
- **Interfaces must be**
 - ...at the service of the user, providing what is needed at every turn.



Making Interfaces Invisible

- **Four important guidelines:**
 1. Follow mental **models**
 2. Direct, **don't discuss**.
 3. Keep tools **close at hand**.
 4. Provide **modeless** feedback.

1. Follow mental models

- The mind looks for a **pattern of cause and effect** to understand the machine's behavior.
- Provide what is **most likely the users will look for** first, based their background.
- Yet, don't forget to improve on "mechanical-age" representations to **move things forward**.



2. Direct, don't discuss

- The ideal interaction is not a dialog – it's more **like using a tool** such as a hammer.
- **Direct feedback** is expected from the tool and the environment – not a dialog box.
- **Direct manipulation** idioms provide better and more flow-inducing interfaces.



3. Keep tools close at hand

- Most programs are **too complex for one mode** of direct manipulation to cover all features.
- **Tools** (effectors, manipulators) offer different modes.
- Make tool manipulation and **switching easy** to prevent flow disturbance (provide shortcuts).
- Users should **not have to divert attention** from application to find a tool.



4. Modeless feedback

- Presented information and feedback should be **built into the main interface** and shouldn't stop the normal flow of activities.
- The **heads-up display** (HUD) is typically used for this purpose.



Part II: Brenda Laurel

COMPUTER AS THEATRE

Dramatic Techniques

- **Dramatic Theory**

- Used to design interesting, engaging and satisfying human-computer activities.



From "Ivanov" in Þjóðleikhúsið, 2008

Drama vs. Narrative

- Sometimes emphasis on **narrative**, but...
- Human-Computer Activities are **more like drama** than narrative.

Narrative	Drama
Description [storytelling]	Action (Enactment)
Detail [expansion]	Intensity (Intensification)
Thematic Links [episodic]	Causal Links (Unity of action)

Drama and Time



"...I'd design games that were meant to be played in 4-5 hours. The games would be of the same scope that I currently design, I'd just remove the silly time-wasting puzzles and take the player for an intense ride. The experience they would leave with would be much more entertaining and a lot less frustrating." (Gilbert, "Monkey Island", 1989)

- Drama imposes time limits
 - So does an interactive system.

Part III: Brenda Laurel

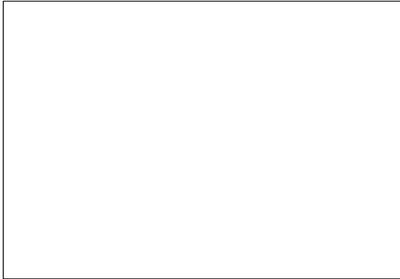
CONSTRAINTS

Interaction Constraints

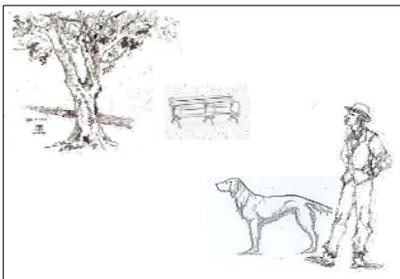
- Two kinds of imposed constraints:
 - "Real world" or hardware related.
 - "**Mimetic world**" or software related.



Create a new story...

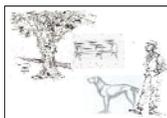


Create a new story...



Creativity and Constraints

- **No limits**
 - Can produce a sense of powerlessness or even complete **paralysis** of imagination.
- **Limitations**
 - Paradoxically increase **imaginative power** by reducing open possibilities.



Creativity and Constraints

- **Closed Mimetic Worlds**

- Provide a creative security net.
- People respect this.
- Increased potential for **effective agency** where causal relations are clear and not left open.



Eve-Online by CCP Games

Giving Constraints

- How should mimetic constraints be given?

- **Explicitly**

- **Undisguised** constraint
 - Directly available (e.g. menus)
- Can be used **before** action.

- **Implicitly**

- **Disguised** constraint
 - Simply inferred from behavior of system (e.g. failing)
- Can be used **during** action.

The Power of Context

- **Constraints should limit...**

- ...**not** what we can do,
- ...but what we are **likely to think of doing**.

- **Context**

- Is the most effective way to do this.

The Power of Context

- **Mimetic Context**

- Can provide a way to make constraints appear to be **within the scope of the activity**.



In-Story Tutorial in Star-Trek Elite Forces by Raven Software

The Power of Context

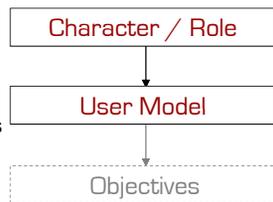
- **Character as Mimetic Context**

- A role template giving rise to action.
- Implicitly constrains choices.
- Inspires creativity in fulfilling that role.



User as Character

- Think of the user of your environment as someone taking a specific **role**.
- A **user model** maps roles to specific interaction objectives.
- **Support these objectives** with the available actions.



Using Plan Recognition



- **Discover**
 - What action is being performed (observation).
 - What process has started (inference).
 - What objectives are being pursued (user model).
- **Intervene**
 - To help user fulfil their role.
 - To guide user to a different role.

References

- Cooper, A., Reimann, R., Reimann, R. M., and Dubberly, H. 2003 *About Face 2.0: the Essentials of Interaction Design*. John Wiley & Sons, Inc.
- Laurel, B. 1993 *Computers As Theatre*. 2nd. Addison-Wesley Longman Publishing Co., Inc.
