Screenreading Lab 1:
Annotation Interfaces

ENGL W3986 | Spring 2014 | Columbia University

February 19, 2014

> *In the semiotic engineering theory of human-computer interaction, "interactive computer systems are viewed as one-shot messages sent from designers to users. Through the system's interface, in many direct and indirect ways, designers are telling the users how they can, should, or must interact with the system in order to achieve a particular range of goals anticipated at design time."*[1]
>
> *"A semiotic perspective is particularly attractive for HCI because it underlies the fact that every computer artifact necessarily introduces new signs or sign systems in its users' universe."*
>
> *"... designer-user communication at interaction time is a new type of human communication, constrained by formal computational factors."*

Description

Following the **semiotic engineering** approach to interface design, this lab provides the opportunity to assess the approach taken by various digital platforms to textual annotation.

Platforms

- SocialBook (Institute for the Future of the Book)
- Annotation Studio (MIT Hyperstudio)
- Rap Genius
- Prism (University of Virginia Scholar's Lab)

[1] Clarisse Sieckenius de Souza, "Semiotic Engineering: Bringing Designers and Users Together at Interaction Time," *Interacting With Computers*, 18, (2005): 317–341.

Goals

- Learn to write scholarly annotations on a digital text
- Work collaboratively to annotate a text
- Work collaboratively to decide on a rationale for annotation
- Learn about the relevance of audience and readership to textual presentation
- Gain critical awareness of annotation processes as part of scholarly textual study

Process

We will use these four different platforms to annotate the same text. Pay particular attention to the choices made by the designers of these annotation interfaces. As you read through the text and mark it up, think through some of the questions below. You can use Neilsen's ten usability heuristics (attached) to guide your questioning of the interface.

Questions

- What reviewing tools are available? What is the lexicon for these tools (note, highlight, comment) and how does this terminology influence our understanding of their functions?
- What form of reading does this interface encourage? What forms of reading does it exclude?
- Is there a note review/summary available? In what order are these notes presented – page order, or the order in which they were marked?
- Are concrete steps to carry out various tasks communicated clearly enough? Does the interface allow for and value innovative problem-solving strategies? That is, does the interface allow us to negotiate meaning in the text we edit?
- Who does this interface assume we are?
- What pronouns are used for various interlocutors in this interface: author, reader, editor, designer, admin, etc. You? Me? I?
- How should your annotations be presented to be most effective? (As footnotes, endnotes, marginal notes, some other format?) What difference does this make to the reader's experience of the text?

Exchange

When the annotation process is complete, look back over the newly marked-up text as a group. Compare the annotated version with the original, and then in

discussion consider the following questions:

- How much difference did the annotations make to the comprehensibility of the text? What insights were possible with the annotated version that were not possible with the original?
- What kinds of annotations were most helpful? Which ones were least helpful?
- What is the overall effect of the annotation on the text? How does it alter your impression of the text?
- How did the annotations address you as a reader? What knowledge did they assume you had? Did you feel comfortable in that role?

Jakob Nielsen's usability Heuristics[2]

1. **Visibility of system status:** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

2. **Match between system and the real world:** The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

3. **User control and freedom:** Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

4. **Consistency and standards:** Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

5. **Error prevention:** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

6. **Recognition rather than recall:** Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7. **Flexibility and efficiency of use:** Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8. **Aesthetic and minimalist design:** Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9. **Help users recognize, diagnose, and recover from errors:** Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

[2]Jakob Nielsen, "Heuristic Evaluation," in *Usability Inspection Methods*, ed. Jakob Nielsen and R. L. Mack, (New York: Wiley, 1994), http://www.nngroup.com/articles/ten-usability-heuristics/.

10. **Help and documentation:** Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Acknowledgements

Portions of this assignment are taken from the Brown Women Writer's Project Collaborative Annotation Assignment.[3]

[3] http://www.wwp.brown.edu/wwo/teaching/assignments/annotation.html