ANNOTATION: A STEP TOWARDS THE READ/WRITE DOCUMENT

Peter and Heather Brown

Department of Computer Science

University of Exeter

Exeter EX4 4QF, UK

P.J.Brown@ex.ac.uk,

H.Brown@ex.ac.uk

http://www.dcs.ex.ac.uk/~ pjbrown/

PAPER/ELECTRONIC DOCUMENTS

Consider a pencil: the electronic document world is not remotely close to emulating its HCI characteristics...

... and a pencil can annotate any paper document.

DOCUMENT SYSTEMS

1) reading tools, e.g. browsers.

2) writing tools, e.g. editors or word processors.

Reading tools typically have rudimentary facilities for writing, and vice versa.

There are specialised systems for critiquing and conferencing.

THE DREAM

- all software tools for documents allow reading and writing in an integrated way: every document is a *read/write* document.
- writing while reading: annotation, either personal or shared.
- reading while writing: pop-up suggestions. These can be regarded as pro-active, and continually changing, system-generated annotations.
- thus a uniform and flexible approach to annotation is a key to the dream of a read/write document.

Annotations are main topic of this talk; a fundamental issue is the relationship of the concept of annotation to the concept of hypertext linking.

ANNOTATIONS

- can be an enhancement to the original document or an edit.
- should be applicable to ‘mark-up’ as well as content, e.g. to use of fonts or to names of hypertext links.
- should be persistent.

DEFINITIONS

- Informal definition of a simple annotation:

  An annotation is where the author causes an existing document, which may be owned by someone else, to link to a document of their own.

- this mirrors a definition for a simple hypertext link:

  A hypertext link is where the author causes a document of their own to link to an existing document, which may be owned by someone else.

- The only difference is ownership: whether the document that the author creates is a source or a destination of a link. Similarities are particularly strong when (a) the author is annotating their own document, or (b) all links are two-way.

PRESENTATION

The default way of presenting annotations differs from hypertext links:

- annotations are typically presented in-line: the reader wants to see both the annotation and the original text.
- annotations may pop up automatically without the need for the user to select them (but beware infinite recursion of annotations of annotations).
- edit-annotations need special treatment as they modify the image of the original document.

... but differences are not fundamental.

APPLICATIONS

A key potential application is in scientific research, e.g. bioinformatics.

Discovery may, for example, involve, bringing together knowledge from many disparate web pages. A simple yet powerful annotation facility can be a key to doing this well.

A further application is adaptive systems, e.g. for retrieval: by making an annotation, the user is making a statement about what interests them, and this can be exploited to tailor systems to meet the user's needs.

DATA TYPES FOR ANNOTATIONS

- data types in programming languages add discipline, functionality and expressive power. Ditto for data types of annotations.
- sample data types for annotations: Vital, Suggested, Supporting-argument, Counter-argument, Cross-reference.
- when lots of people are adding different annotations to the same original document, data types may be used to help create order out of chaos: e.g. by organising the discussion into threads.

ANCHORS

- in hypertext a simple link has two anchors, the source and destination; each is typically a piece of text, or just a position, in a document. More sophisticated hypertext systems allow an anchor to be any occurrence of a certain word or concept.
- annotations have the same needs for anchors, plus the additional requirement for multiple source anchors, e.g. for the annotation ‘I suggest you exchange these two paragraphs’.

WHAT IS ANNOTATED

An annotation may be attached to:

- an image on the screen.
- the notation used to create the image, e.g. the HTML source.
- an underlying data structure from which the source was created, e.g. a database entry.
- a physical object, e.g. in virtual reality.

Generally the higher the level of the abstraction, the more useful it is, but the harder it is to provide. (A problem in computer science in general.)

AUTOMATIC ANNOTATIONS

- sample usage: annotations popping up on a map used by a tourist.
- may be created by autonomous agents.

CHANGE

Change is a fundamental problem in most areas of computer science. (Three problems: change, complexity, levels of abstraction.)

- change is a fundamental problem with hypertext links; 10-20% of links are typically dangling. (Dangling link is like orphan annotation.)
- change of underlying document is a big problem for annotations. There are clever systems for adjusting the position of annotations to cater for a changed document. However user experience suggests that some are too clever by half: in many cases it is better to give up and mark an annotation as an 'orphan'.
- change will always be a limitation to the ideal of a read/write document.

UNIVERSALITY

Universality is a pipe-dream: the same mechanism works with all documents. However cut-and-paste comes close to achieving this.

Ideally annotation should not only apply to web pages but to e-mail, a database field, a video, or more generally to any physical object.

In practice universality often comes because one system knocks out all the competition, e.g. the web.

CONCLUSIONS

Annotations are a key to the read/write document; facilities we need are:

- data types.
- multi-way anchors.
- different abstraction levels.
- automatic (and intelligent) creation.
- new theoretical models, probably combining annotation and hypertext.
- some robustness over change.
- capabilities not only to do what we have done before, but fundamentally new practices.

... but a pencil still works.