E-Study

Specifications Document

author: Mark Humphrey <msh>

date: February 7, 2003

1. Introduction

1.1 Problem Statement

While the world gets more electronic-based with every day, students still typically lug around massive textbooks when they could easily browse some sort of e-book. This is becoming more and more reasonable with the emergence of wireless connectivity and the dropping prices of laptops and tablet-pcs. One feature which students currently lack, however, is the ability to effectively mark up (take notes in, highlight) their electronic books/lectures. This is a necessity to most students and creates a reliance on the traditional "textbook".

1.2 Project Description

The general idea behind this project is to create an electronic study environment in which students can mark up books/lectures/presentations in several different ways and then provide an organized interface at which students can get at these notes as well as additional study tools to aid in the study process. This involves a convenient gui which provides the students with easy navigation through their notes and access to all the additional features this project may have. These features include a task master, designed to keep students focused, and possible quizzing and note card tools. User controls would be the mouse and keyboard, but the ideal situation of having a tablet pc with a "pen" would be kept in mind.

1.3 Similar Existing Products

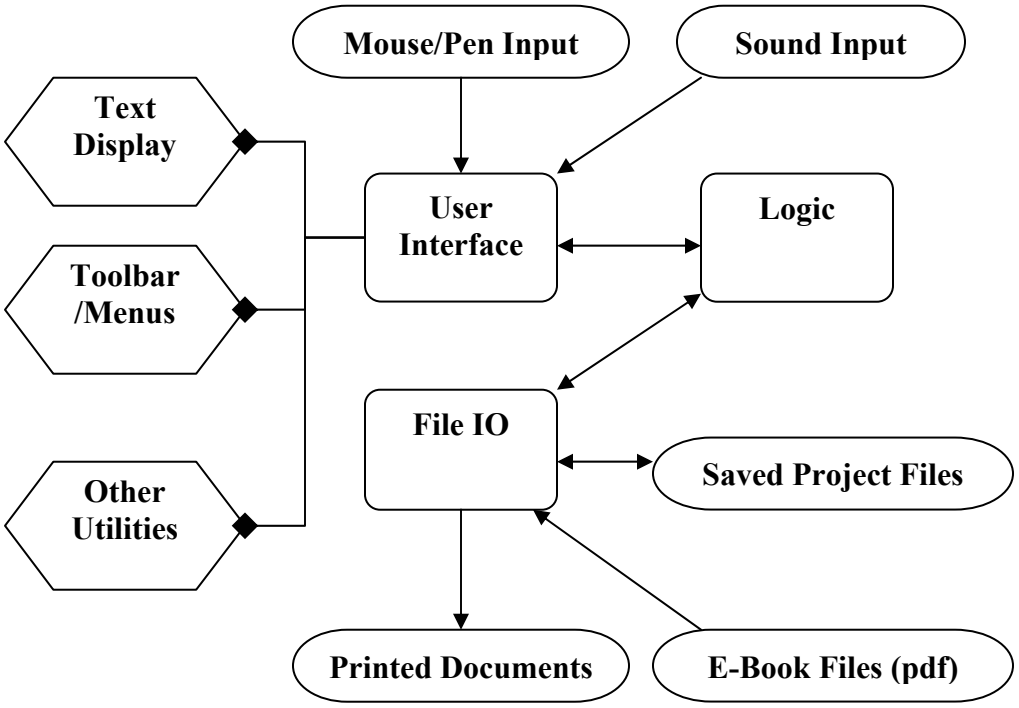
Several companies have e-book readers, most which simply allow the reading of e-books. Adobe's "ebook Reader" actually allows for highlighting of text. However, most of these products allow limited interaction with the text and are primarily designed for "ebooks" as opposed to the variety of different types of input that this program can expect. An example is Microsoft's "PowerPoint" which allows one to write with the mouse on a presentation, but does not save the marks. Adobe “Acrobat” does allow for lots of annotation, but is a full featured document creation tool and has too many features for the typical user.

1.4 Target Users

Everyone who wishes that they could do more than just read text on a computer would find this project useful. It is ideally suited for students, probably applying to high school and college students the most as the integrated study environment requires a certain level of sophistication. The variety of options which are provided should suit both the ultra-studious individual and casual studier. Students at Brown University would provide a good pool of users, in particular those who major in the humanities, although this project applies to science concentrators to a large degree as well.

2. System Details

2.1 System Model Diagram



2.2 System Component Description

- **Mouse/Pen Input**
  This is how the user interacts with the application, through clicking and mouse movement, although more ideally pen movement.

- **Sound Input**
  Although this is not a standard feature, this would be used for the possible voice note extension.

- **User Interface**
  - ***Text Display***
    Main display window where text is viewed
  - ***Toolbars/Menus***
    Consists of various buttons and popup-menus
  - ***Other Utilities***
    Displays optional features such as note-cards/quizzes

- **Logic**
  Delegates tasks to the various system components handles decisions making

- **FileIO**
  Handles all file operations from loading and saving of project files, to importing of e-books. Printing operations will also pass though here.

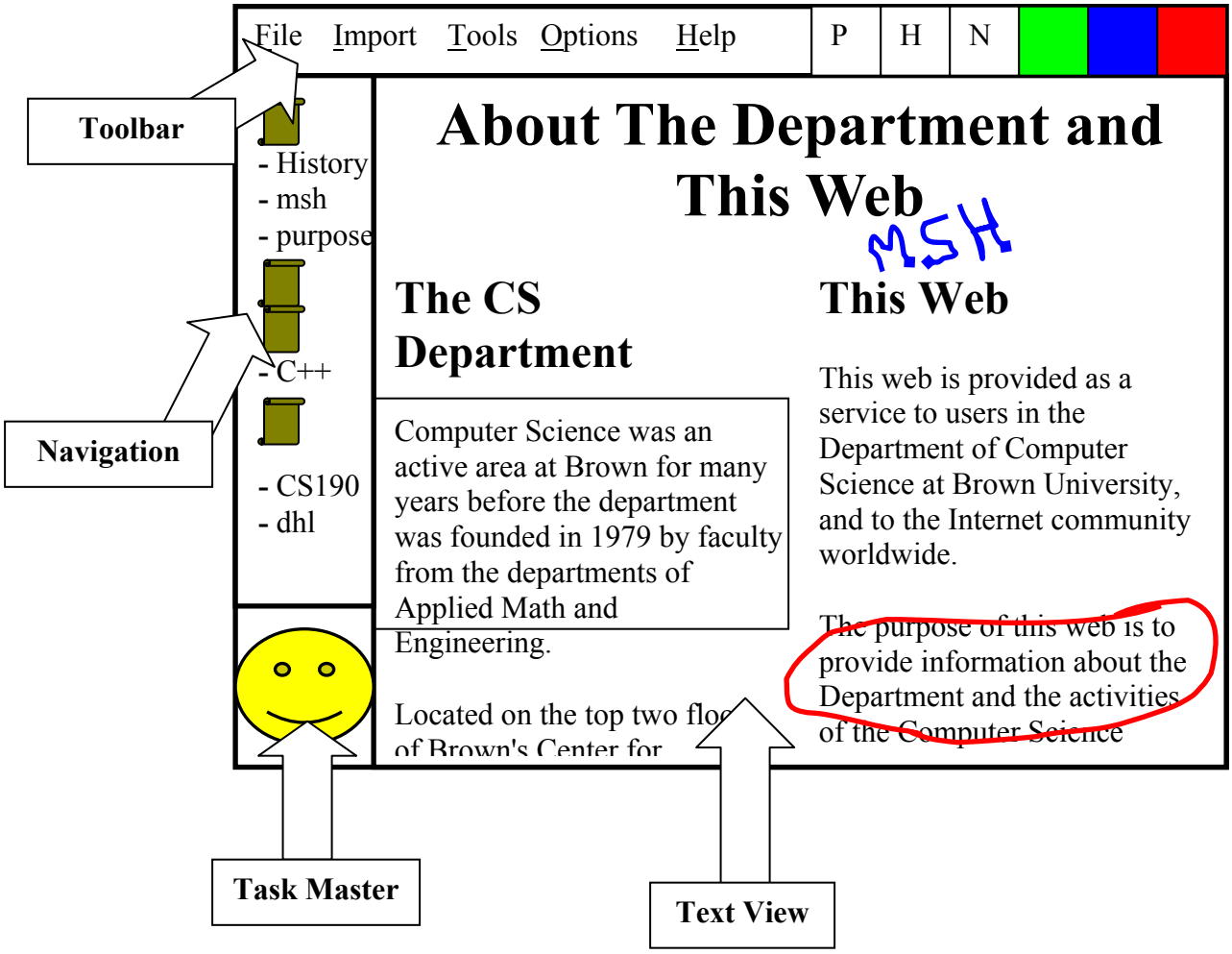
- **E-Book Files (pdf)**
  Input files which are impoted by the FileIO

- **Saved Project Files**
  Stored collection of annotated “e-books”, links, notecards, etc.

- **Printed Documents**
  Printed selection from currently open project file

3. User Interface

3.1 User Interface Diagram



3.2 User Interface Component Description

- **Text View**
  Where "e-books" are displayed and annotated.

- **Navigation**
  Allows visual and easy jumping between books, links, etc.

- **Task Master**
  Keeps user on task; tracks application idle time and gives visual reminders

- **Toolbar**
  Standard save/load options as well as numerous miscellaneous functions

- **Pop-up Menu**
  Circular menu appears when mouse held down with intelligent options depending on where mouse is

4. Non-Functional Requirements

4.1 Performance

This software must be fairly lightweight as it is intended to be a study tool and companion, and it can be expected that users will often have word processors, web browsers and other utilities open. Quick switching of pages and documents is also a necessity.

4.2 Testing Criteria

Once the basic skeleton of a gui is constructed, each element of the project is fairly independent and can be tested separately. The only difficulty could be that most pieces are dependent in some way on the gui, so it is difficult to test in isolation from the gui.

4.3 Reliability

As this software is intended as a note-taking tool which will organized and record all of an individuals studies, it is crucial that losing work is avoided at all cost. Backups should probably be made periodically and some sort of recovery option included.

4.4 Ease of use

Since this software is designed for people with a wide range of computing skills its use must be very intuitive. This is very important as any sort of difficulty that users have with this software will discourage them from moving from to this new form of electronic text.

4.5 Portability

This software will be written to run on linux, however it would be ideal if it were written with a graphics tool kit which could run on Windows, as this would be the eventual destination of future versions.

4.6 Hardware/Software Requirements

This program should be able to be written in C++ on the linux machines which are readily available. Ideally it would be run on windows, but that can be done at a future date.

4.7 Documentation

Documentation should be thorough and easy to use, with the ability to search by content, topic or keyword. Some sort of html form of documentation is preferable as it familiar to most users and easy to navigate.

4.8 Dependencies on Other Systems

The eventual format which e-books are stored in is one external dependency as well as the particular format which this study tool works with.

5. Functionality Requirements

5.1 Basic Features

- Ability to import and view pdf format files
- Simple organization and compilation of all files
  - drag and drop organizational hierarchy
  - easy point and click navigation
- Intuitive graphical user interface
  - no more than two level of menus
  - designed with the ideal situation of using a "pen" kept in mind
- Several types of marking
  - Highlighting (selected with mouse)
  - Freehand (drawn with mouse or pen)
  - Sticky Note (textbox with typed text)
- Linking/Bookmarking
  - Ability to jump through all the different type of markings
  - Relevant sets of notes can be linked together
  - Arbitrary bookmarks can also be set
- Task Master
  - Generates a study schedule given goals and available time

- Tracks idle time and gives visual and audio reminders to stay on task
- Schedules break time to optimize studying

5.2 Optional Features

- Voice notes
  - Can be linked to a particular section of text
  - Recorded via microphone

- Index card factory
  - Compiles index cards from existing notes
  - Produces a printable format

- Quizzes
  - Can tabulate primitive questions based on notes
  - Keeps track of students results on quizzes
  - Analyzes strengths/weakness and suggests which sections to study

6. Uncertainties

It is unlikely that this application will come into immediate use, as most text books have not migrated into electronic form yet. Also questionable is whether or not there is a need for this as Adobe has several tools which do much of what this is intended to do, although with a broader focus. There are also technological barriers, as this would really work best in a work where people all have slim, lightweight, crystal clear pen-tablets, a reality which is several years away. In addition, linux is not the idea platform for distribution to the majority of users. Finally, it may prove difficult to design and implement this in a way such that it can be easily scaled up.