Specifications

Louisa Rosenheck
(lrosenhe)

February 7, 2003

CS190

1. Introduction

1.1 Premise

There is a lot of teaching and learning software available, from teaching aids for the classroom to learn at home CD Roms, with a wide range of content such as arithmetic, foreign languages, or home improvement. This type of software is very effective because of the visual and audio components and most of all because people can learn on their own time and at their own pace. But- what about those creative people that want to spice up their syllabus with their own electronic lessons? They must resort to one-sided Power Point type presentations or struggle with html. What teachers need is a creative software with built in templates for the kinds of interactive lessons they need which can be customized to fit an individual's teaching style and philosophy.

1.2 Focus

This version of Teaching Tools is designed for language learning, and therefore the lesson templates, quizzes, and tests will utilize formats that facilitate teaching a foreign language.

1.3 Users

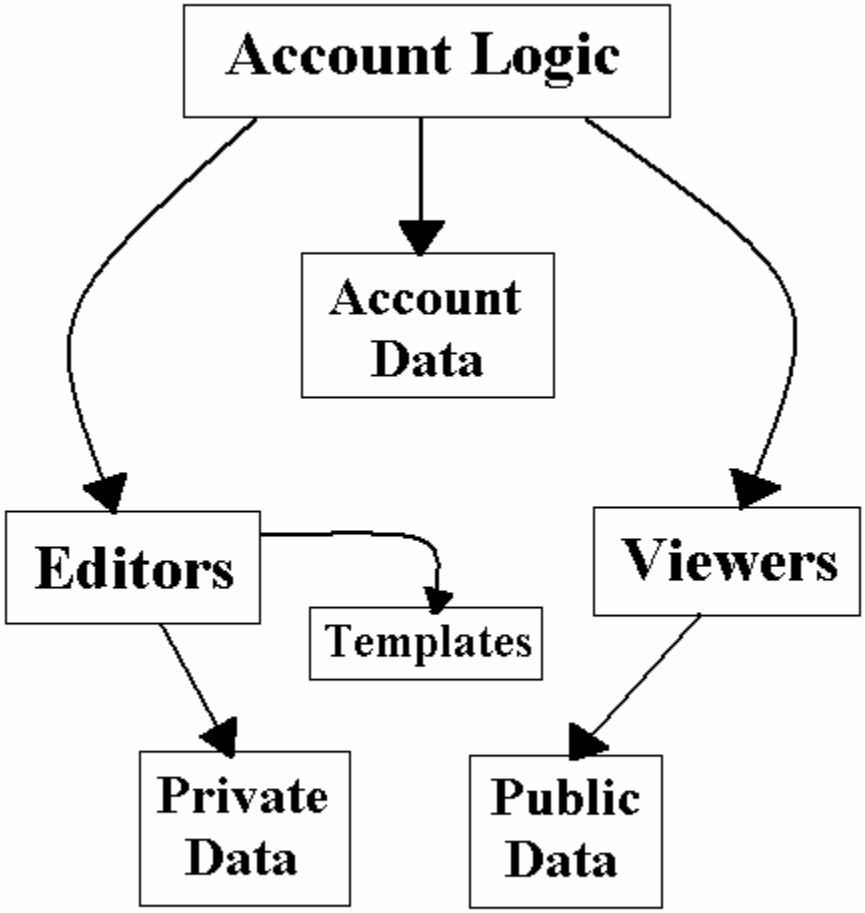
Local users of this software include language teachers and TA's here at Brown and possibly at nearby high schools or elementary schools. These users are accessible and presumably willing to try innovative teaching techniques and therefore would be both convenient for testing and interested in the final product.

1.4 Description

Each user will have their own account, with separate types of accounts for teachers and students, and capabilities to specify which students have access to which courses. A given user may be both a student and teacher, but not for the same course. The students will be able to study their current or previous lessons within a course, do practice exercises, and take quizzes and tests, as well as to see their own grades. Teachers' accounts have many more capabilities, including looking at each student's accomplishments, creating and editing new lessons, and reading the gradebook for their class. The lesson editor will have a GUI similar to common painting and drawing programs people are familiar with. They can specify whether lessons are continuous or separate, and the result will be something in between a Power Point presentation and an html document, that is, it will have direction, but the user also interacts with it and can decide when to go back and forth.

2. System Model

2.1 System Model Diagram



2.2 System Model Description

2.2.1 Account Logic

This module regulates the program's flow of control. It accesses all user identity information, gives the user access to his/her account, and provides the interface which the user interacts with to access the other parts of the program.

2.2.2 Account Data
This is where account data is stored, such as user names and passwords, what courses users are associated with, how far students have gone within a course, and other identity and permissions information such as what lessons the user has access to and whether he/she is a student or teacher for a specific course.

2.2.3 Public Data
This is where the program stores lessons that are finished and ready to be used by students. Teachers will have specified whether anyone can see the lessons or whether only students with permissions for the course have access to them. Other materials connected to the lessons are stored here as well such as tests and practice exercises. These elements will also have specific properties in regards to who can access them and when.

2.2.4 Private Data
This includes data not accessible to certain people, namely students, and includes grades, test and quiz answers, and lessons the teacher has not finished creating yet. Only the creator of these items can see and edit them, unless they have specified someone else, such as a TA, who should also have access.

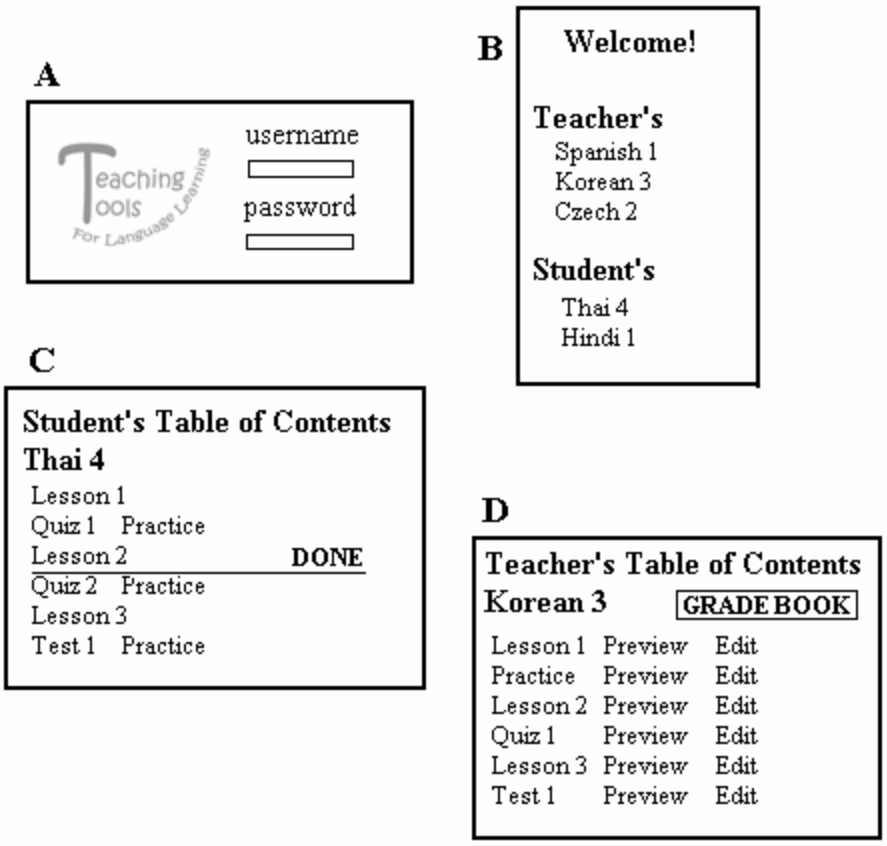
2.2.5 Template Data
This data consists of the templates accessed through the various editors when a teacher is creating new elements. These templates enable teachers to make up lesson slides, tests, practice exercises, and answer sheets quickly and clearly. Users can also create their own templates and save them, in which case they are also stored here.

2.2.6 Viewers
Viewers are used to look at and use various elements. Students will use the lesson viewer to go through lessons and interact with them, the test viewer to take tests and submit answers, and the exercise viewer to work on practice questions and see the answer key. Teachers use these viewers to preview their finished course material, as well as the answer key viewer to read their rubrics, the assignment viewer to read and grade assignments and tests, and the gradebook viewer to see students' grades and how far they've gone (i.e. how many lessons they have worked through).

2.2.7 Editors
The element editors are the most powerful module in the software since they are used to create the lessons and other course materials. Each element has a corresponding editor, including lessons, tests, practice exercises, rubrics, and gradebooks. The lesson, test, and exercise editors also contain a permissions editor so they can specify if each element can be seen at any time or only after having done certain other assignments.

3. User Interface

3.1 User Interface Diagrams



3.2 User Interface Description

3.2.1 Flow of Control & Organization

When Teaching Tools starts, the user will first be asked to log in or create a new account (diagram A). Upon login, the user's welcome screen will contain a list of the courses he/she is a teacher for and a list of the courses he/she is a student in (diagram B). The user will then choose which course he/she wants to work on, and the table of contents for that course will appear. The student's table of contents (diagram C) includes all the lessons, tests, and exercises the student has done and all those not yet done. He can then choose which lesson to work on or review. The teacher's table of contents (diagram D) contains all the elements created so far, in order designated by the teacher, and from there the teacher can choose what element to work on and whether to preview it with a viewer or continue creating it with an editor. This screen also gives a teacher the option to view or edit the course's gradebook from where he/she can see students' grades, monitor their progress, and grade tests.

3.2.2 Viewers

- The lesson viewer has back and forward buttons and the option to pause or stop working and save the progress made.
- The quiz and test viewer has buttons to go back and forth, a timer showing how much time the student has been working and/or how much time is left.
- The practice exercise viewer has buttons to go back and forth, optional timer functionality, and a show answer button.
- The gradebook viewer displays the grades of all students in a class, in a spreadsheet format, as well as how many lessons they have worked through.
- The rubric viewer displays test questions, their answers, and their point values.

3.2.3 Editors

- The lesson, quiz/test, and exercise editors are similar to basic drawing programs. The user controls the layout and can add text and import images. When new elements are added to a course, they are also added to the table of contents and the gradebook if applicable, and the teacher can designate their order and any other access restrictions.
- The gradebook editor is a basic spreadsheet showing students' grades and progress, and it gives the teacher the option to add more fields to the spreadsheet and grade assignments handed in by the students.
- The rubric editor allows teachers to create answer keys and rubrics, designating the answer and point value for each question.

4. Other Requirements

4.1 Preliminary Research

In order to further determine the needs of local users and specify the functions of each editor, more extensive research will need to be done. Users are readily available in the Brown community and presumably cooperative. First we will need to examine language teachers' current lesson plans, teaching materials, and course formats. Then, obtaining personal information will be done by questionnaires and interviews about which features would be most useful to whom.

4.2 User Testing

Once a semi-functional version of the program is ready, ideally more user testing would be done to find out what features are and are not user friendly, how to make the editors more intuitive, and what is and isn't useful to language teachers. This would require the program to not only be written very extensibly for easy accommodation, but also to have a working version done fairly early. This is not likely in one semester, so we will probably have to rely on the preliminary research.

4.3 Reliability

The security of the program is vital, in terms of keeping track of who should have access to what elements of a course's materials. This will be done primarily by password protection, because it is imperative that it be impossible for students to gain access to answer keys and gradebooks. Also, the submission of test answers and storage of answers and grades must never fail or the teacher's resulting actions might be unfair to the entire class.

4.4 Ease of Use

The software, especially the editors, must be extremely intuitive and user friendly because this program is geared towards teachers who want to incorporate personalized materials into their courses, but who might not have technical knowledge or extensive computer skills.