**CLASH Product Description CS411w Lab1**

**Charles Stafford- Blue/Purple Team**

**Old Dominion University**

**Table of Contents**

**1 Intro…………………………………………………….……3**

**2 Product Description…………………………………...……4**

**2.1 Key Product Features and Capabilities……………….…...4**

**2.2 Major Hardware and Software Components……………..5**

**3 Case Study…………………………………………………...6**

**4 Prototype Description……………………………………….7**

**4.1 Hardware and Software Prototype Architecture……….....7**

**4.2 Prototype Features and Capabilities………………….……8**

**4.3 Prototype Development Challenges………………….…….9**

**Glossary……………………………………………………………..10**

**References…………………………………………………………..11**

**1 Intro**

CLASH is a web based application designed to increase the reading speed and comprehension of ESL, or English as a second language, students. The application will consist of two modules connected by a website. The first module, COLRS, or Colored Organized Lexical Recognition Software, will take user inputted text and parse out the different parts of speech found in the text. The module will then color all the parts of speech in the text to help with reading comprehension. The second module, Slash, will read in user inputted text, and turn the text into a series of lexical bundles (a group of words that we subconsciously associate with each other) to help ESL students increase their reading speeds.

Reading speed and comprehension for ESL students is a major issue at schools all across the nation. In 2004, there were nearly 5 million ESL students enrolled in public schools in the United States. Of these students, only 18.7 percent of them were either performing at or above an average academic level. On top of this, ESL students have up to a 400% higher chance to drop out compared to English speaking students.

The current teaching process for ESL is very basic. The professor writes a sentence on a white board, and manually parses it, pointing out all the parts of speech. Lexical Bundles are rarely even used. Instead, professors use programs like Spreeder, a speed reading program that puts a single word on screen for a user defined amount of time. Spreeder does help users increase reading speeds, but at the loss of comprehension.

This is where CLASH comes in. CLASH intends to increase both reading comprehension and reading speed by creating a speed reading program that uses Lexical Bundles. Through the use of Lexical Bundles, it has been demonstrated that students increased their reading comprehension and reading. It is hoped that CLASH can easily and efficiently bring this to the classroom.

**2 Product Description**

CLASH is designed to help ESL professors in their goal of preparing ESL students for their college courses. CLASH will consist of two modules, COLRS, and Slash, which will be accessed through a web page. COLRS will focus on parsing sentences, and coloring parts of speech, while Slash will focus on increasing reading speed through the use of Lexical Bundles.

COLRS will help students with the difficult task of identifying the various parts of speech in a sentence. Many languages have defined sentence structure where verbs and nouns can be easily identified either by their placement a sentence, or by their endings. Through the color coding the traditional 8 parts of speech, ESL students will find it easier to locate and understand English grammar.

Slash will focus more on reading speed and comprehension. Slash will read through documents like a normal speed reader, but instead of going one word at a time, it will show entire Lexical Bundles at once. By showing entire thought groups at once, users will have a higher comprehension of the text, than if they were using a traditional reader.

## 2.1 Key Product Features and Capabilities

CLASH will be a web-based application that will allow users to improve their reading comprehension by breaking up sentences into lexical bundles and showing the parts of speech. CLASH features can be accessed using a standard web browser and internet connection. Through the application’s user interface, users can control the speed at which lexical bundles are displayed. Parts of speech can be viewed in a separate part of the user interface.

The product features individual password-controlled user logins with three different types of user roles. These roles include Administrator, Instructor, and Student. The Student will be able to control their reading speed, type of view, and which available document to view. The Instructor will include the Student user capabilities plus more. The Instructor will be able to add/remove Students as users and select the documents available to be viewed by the Students. The application allows for the Instructor to view activity data for the Student users. This activity data will include the student’s current reading speed, and the amount of time spent on the site. The Instructor will be able to upload documents to the server to be parsed, edit files, and delete documents currently on the server. The Administrator has all the capabilities of the Instructor plus the ability to add/remove Instructors as users.

One of the features that make CLASH unique is that it both displays parts of speech and lexical bundles. This allows students to improve their reading comprehension by having a speed reading application that does not break up the individual lexical bundle. There is the ability to pause and change the display speed. CLASH will include a text parser that is able to automatically identify parts of speech, and color them accordingly, allowing for easy identification. It will also allow instructor review of the parsed text, to ensure accuracy(because no program is perfect). CLASH will be able to save usage data, allowing instructors to review student progress, and fine tune their lectures accordingly. CLASH is the first web-based speed reader specifically designed for use in ESL instruction. Instructors can have documents available to the students at their appropriate level. By focusing on ESL students, CLASH will give students and instructors a powerful tool to increase reading speed and comprehension.

**2.2 Major Hardware and Software Components**

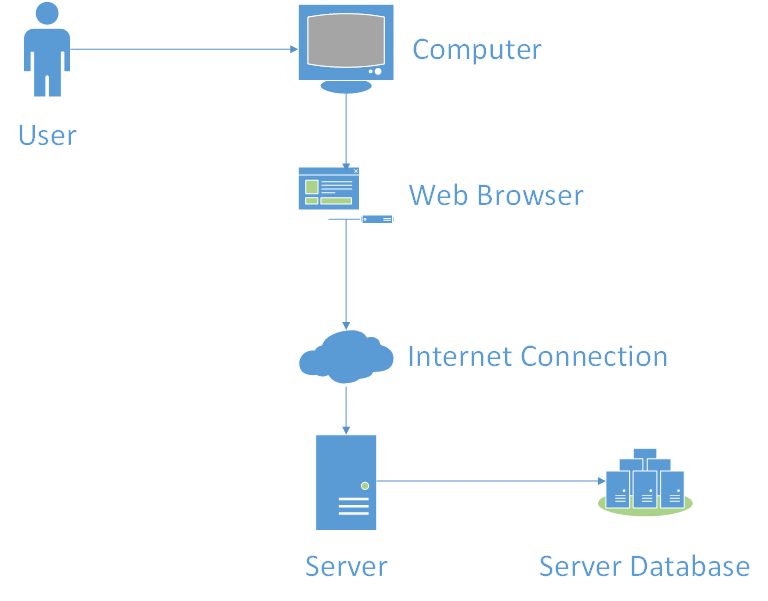
CLASH will be a web based program hosted from a central server, and accessed from any of the major web browsers. CLASH has only a few hardware requirements. Namely, a server with a connected database for the server component of the program, and a device connected to the internet capable of opening a browser for the client side.

The COLRS module will use an open source python script called NLTK(or Natural Language Tool Kit) to parse the user’s input. COLRS will take the output of this script, and will convert it to a more readable and color coded format. The final output will be two files, consisting of a text stream and a set of tags pointing to all the parts of speech in the document.

The Slash module will take the two files from COLRS, and will read through the tags and text stream, marking the Lexical Bundles. Slash will add a new tag to the tag list showing where the Lexical Bundles or “slashes” are. This module will be supplemented by a exception list with the most common or recurring Lexical Bundles.

The Web Page portion of the program will take the streaming and tagging files from the server, and present it to the reader. There will be 3 rendering modes: Colored, Slashed, and Speed Reader. Colored will render up the parts of speech. Slashed will mark all of the lexical bundles in the text. Speed Reader will take the lexical bundles, and run through the document at a user defined speed, showing each lexical bundle in turn.

**Hardware Requirements**



**3 Case Study**

This project was started at the behest of Professor Raver-Lampman to help him teach ESL students. He wants to use this program as a supplement to his lectures. Students could start CLASH at home, and study on their own time.

Professor Raver-Lampman approached the CS department, and managed to recruit two teams to his project, creating the combined Blue/Purple team.

Raver-Lampman has expressed interest in getting other Virginia schools to try out this program, and is also interested in potentially commercializing the final program.

**4 Prototype Description**

CLASH’s prototype will be built around the SPA template. SPA, or Single Page Application, is a web based application that has a single page, where all the functionality of the application is accessible from menus without reloading. SPA’s are traditionally built using Javascript, as are all of its functional components. The web page will be an SPA, built around Javascript, and it will communicate with the CLASH server to render the needed information. CLASH will be using MySQL for the Database, and will be using Node.js as the server application.

The reason CLASH will be using SPA is to create as accessible of a website as possible. Placing the entire application on a single page keeps the user interface relatively clean and easy to use. Non-tech savvy users should be able to navigate the site as well as anyone else, and that means a concerted effort to keep the user interface as simple and easy to understand as possible.

Sadly, the prototype simply cannot have every feature of the real world product. There is simply too little time to do everything on the wishlist. Instead of hap-hazardly introducing half-finished features, the prototype will be a scaled down version of the final product.

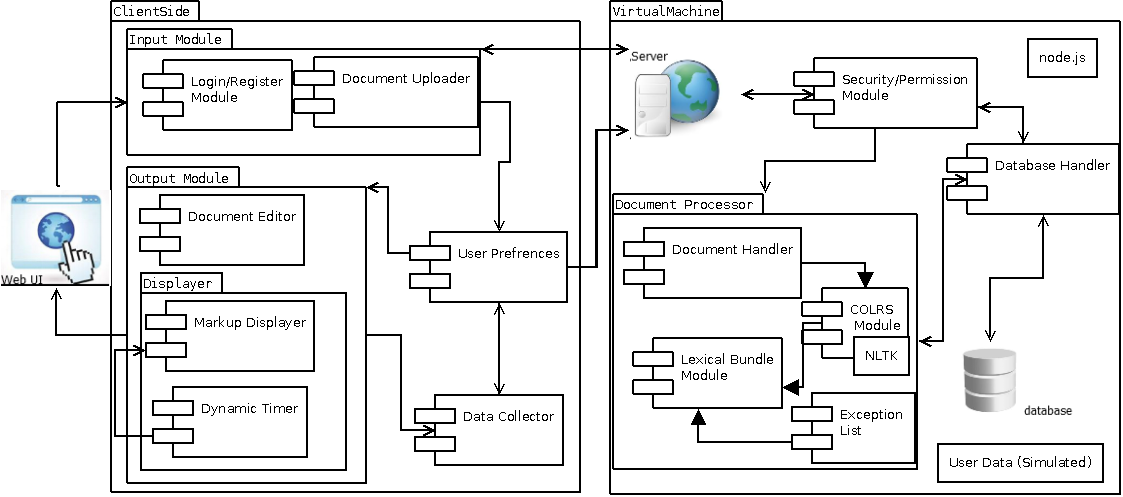
The main differences between the final product and the prototype will be the lack of a more advanced POS tagging system, lack of integration with ODU servers (and the lack of real enrollment data that implies), limited student metrics, and the lack of a true “homework” mode.

As stated above, one of the major hurdles for the project will be the lack of access to ODU enrollment data. CLASH will avoid this hurdle by implementing a user management system for instructors. This management system will allow instructors to add and remove users from the database, and can be used en-masse through the use of .csv files. The admins will be able to go one step further, and be able to add and remove instructors, and instructors’ files depending on the situation.

**4.1 Hardware and Software Prototype Architecture**

CLASH’s prototype hardware architecture will be a Virtual Machine hosted by an ODU server. MySQL and Node.js will be installed on the VM, to allow the server to communicate with the software components. The server’s operating system will be Ubuntu. The backend of the server will require NLTK, which will interface with Node.js, which in turn will communicate with the MySQL database. Node.js will also run the SPA that will be seen on the client’s browser, and will receive raw text, and send back parsed text to the client.

**Prototype MFCD**



**4.2 Prototype Features and Capabilities**

The CLASH prototype will be able to parse sentences, identifying both the Parts of Speech, and the Lexical Bundles of the sentence. The program shall render Parts of Speech as colorized words, each part of speech being a different color. The program will render Lexical Bundles in a speed reading format. Bundles will be shown on the screen at a user determined pace, starting from the beginning of the document. There will be a basic reporting system in place, instructors will be able to query individual student’s current Bundle reading speed, as well as the number of files the student has accessed.

The goal of this prototype is for it to be used as a Proof of Concept. It will be tested by a class of ESL students, against a control, to determine if the project is feasible. If students using the program demonstrate an increased reading speed, and higher reading comprehension, then the project can be considered a success.

In order to the risk of failure, the scope of the prototype is very small. If a part of the prototype is not working as intended, it can be focused on in detail, without sacrificing quality.

**4.3 Prototype Development Challenges**

CLASH will have to pass several hurdles before it is ready to be released. First of all, NLTK needs to be linked in with Node.js with a custom program. Then the actual web application needs to be written.

CLASH also needs to easy to use. Creating a simple and accessible interface will require a lot of iterations. Often UI mock ups that look good on the drawing board will simply not work in the final product. Creating a pleasant looking, and useful interface will be critical.

The biggest challenge for the project will be producing accurate Lexical Bundles. If CLASH cannot accurately and reliably create Lexical Bundles, the project will be a complete failure.

**Glossary**

**CLASH** - Color Lexical Analysis algorithm and Slash Handler

**COLRS** – Colored Organized Lexical Recognition Software

**ELC** – English Learning Center

**ESL** – English as second language

**IBT** – International benchmark test

**JSON** – JavaScript Object Notation

**Lexical Bundle** – a group of words that occur repeatedly together within the same register

**MFCD –** Major Functional Component Diagram

**NLTK** – a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for the Python programming language.

**Node.js** – an open source, cross-platform runtime environment for server-side and networking applications.

**POS** – Parts of Speech

**SPA** – single page application, is a highly responsive web application that fits on a single page and does not reload as the web page changes states.

**Ubuntu** – a Debian-based Linux operating system.

**VM** – Virtual Machine

**References**

McKeon, D. (n.d.). Research Talking Points on English Language Learners. Retrieved December 11, 2014.

Tremblay, A., Derwing, B., Libben, G., & Westbury, C. (2011, January 15). Processing Advantages of Lexical Bundles: Evidence From Self-Paced Reading and Sentence Recall Tasks. Retrieved December 10, 2014.

Mikowski, M., & Powell, J. Single Page Applications. Manning Publications 2014.

CS 411w Blue/Purple Group. [www.cs.odu.edu/~411blue/](http://www.cs.odu.edu/~411blue/) .