LAB 1 - CLASH Product Description

James A. Ward

CS411

February 9,2015

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

A U.S. Immigration and Customs Enforcement report dated July 2014 stated, there are nearly 9000 American universities, seminaries, or other academic and vocational training institutions certified to enroll foreign nationals under the Student and Exchange Visitor Program. Distributed among those institutions there are 966,333 foreign students pursuing a program of study on F-1 or M-1 visas. This was an eight percent increase from 2013. Presence in the U.S. under a F-1 or M-1 visa requires the student to maintain full time student status in their program of study. There is a time limit on how long these student visas are valid. Although these students are prepared academically, many of them have limited proficiency in the English language. Colleges and Universities have developed English as a Second Language programs to rapidly teach these students the English language skills needed to succeed in a university setting. There are teaching techniques within these programs that can streamlined through custom software.

CLASH, or Color Lexical Analysis algorithm and Slash Handler is a software tool which automates two manual processes presently used in English as a Second Language (ESL) courses. The software interface is a web-page incorporating two components. The first component 'COLRS' examines a reading sample and identifies the part of speech for every word therein. The the reading sample is displayed with each word color coded by its part of speech. The colors assigned to each part of speech are fixed and of best available contrast to one another. The second component is a slash reading module. The slash reading module examines the structure of a sentence, and disassembles the sentence into individual thought groups. In this domain thought groups are referred to as lexical bundles. The user interface displays those lexical bundles either in original paragraph form with slash marks separating lexical bundles or in a video stream

format where each frame is a lexical bundle. Both of these processes are currently manual operations performed by the ESL course instructor. The present method, due to fixed time available in the classroom and being manually performed, limits the number of reading samples available to the students. The colorizing process is achieved with a different color pen or font color and cannot be quickly shifted to demonstrate separate subsets of the parts of speech. Other similar methods used include circling or underlining various parts of speech in an example sentence.

The slashing process is accomplished by marking a document with slash marks surrounding each lexical bundles within a sentence. Exempli gratia – “The students/ understood incrementally more/ of English grammar/ from day to day.” Old Dominion University ESL instructor Greg Raver-Lampman has stated the purpose of the slash reading exercises is to improve the comprehension of English sentence structure and accelerate reading speed. An English language learner's reading speed is retarded not only due to reading word by word, but also due to the time it takes to reassemble those words into a cohesive thought such as “from day to day”. By introducing lexical bundles non native speakers learn to read the thought group as a single entity instead of individual words.

2 PRODUCT DESCRIPTION

The CLASH software package assists ESL instructors in the preparation of reading samples for use in the classroom. By creating a selection of readings that can be reused and rapidly modified ESL instructors recover time lost to writing out examples during class. This recovered time can be used to cover more content or provide focused attention to students. The system concurrently, gives students greater opportunity for practice of the subject matter.

The target domain of the CLASH software package is English as a Second Language Programs. This domain requires consideration of design criterion of less importance in other software development arenas. The students are English language learners, therefore the interface must be overtly sparse and use intrinsically non-technical terms and phrases such as “Enter ESL Program” or “Access Your Assignments” rather than “Login”. The target domain's instructors and administrators being subject matter experts in the fields of multilingual education or the English language also benefit from these design considerations. The software is intended to provide expanded selection of reading samples. This requires the students be able to access the system outside of the classroom. One solution could be to distribute the software on a CD or USB drive. However, CD distribution would lock down the list of reading samples to those prepared prior to the creation of the disk. To accommodate continual content expansion and access outside of the classroom setting a web-based system was chosen. Using a web-based system also allows the use of JavaScript. All modern web browsers include JavaScript which eliminates any need for students to install additional software.

The subsections of the Product Description Section detail the design criteria of a completed product. Section 4 - Product Prototype Description, describes the functional components and limitations of the proof of concept prototype.

2.1 Key Product Features and Capabilities

The CLASH software package shall be accessed by all users through a web browser. The user interface presented shall alters according user role of the logged in account. User roles include Administrator, Instructor, and Student. Each user account shall be configured to only permit the use of system functions applicable for their user role or those of lower level accounts.

Administrator accounts are permitted to access all features of both lower level account types. Instructor accounts are likewise permitted to access all features of student accounts. The primary purpose of administrator accounts is to create other accounts and maintain the system.

Instructor accounts exist to load reading content to into the system, review the output for correctness and preserve the output for use in the classroom or reading assignments. Instructors shall have two methods available to add reading samples to the system. One method shall be to insert text into a form on the web-page. The second method shall be by file upload. Acceptable file formats are limited to Word (.doc or .docx), Portable Document Format (.pdf), Open Document Text (.odt), and plain text (.txt). It shall be the instructor's responsibility to ensure no copyright infringements or violations of fair use laws occur. Instructors may also input enrollment lists to generate new student accounts. Instructors may add student accounts to student account groups. Accounts may be archived en mass at the end of term or individually upon withdrawal from the ESL program. The instructor's view of the web-page shall include tracking statistics of a single student's or a defined student group's average reading speed improvement. The instructor screen provides an interface for corrections to automatically generated POS marking and Slash locations.

The student interface permits reading of pre-configured assignments in three modes: Slash, COLR, or Slash Playback. Slash and COLR mode display the reading sample as paragraphs either slashed or color coded respectively. The Slash Playback mode appears as a video stream with each lexical bundle as a singe frame. The student is given the ability to start, stop, and increase or reduce playback speed of the lexical bundle stream. In the Slash mode the student would see the reading samples as “The students/ understood incrementally more/ of English

grammar/ from day to day." The student interface shall also include a homework mode. In the homework mode the text is displayed in black font and the student assigns the correct part-of-speech color code to the words. The students assignations are then compared to the Instructors copy for grading.

2.2 Major Hardware and Software Components

As depicted in Figure 1 there is no special hardware requirement for the CLASH software package. A web server is required, it may be a virtual machine, a private server, or a rented server such as Amazon.com Incorporated's EC2 cloud based systems. The system is comprised of the Client Side user interface (UI) and the Server Side. The system is designed as a Single Page Application (SPA) hosted on an Ubuntu Server running MySQL, NGINX, and Node.js. The entire application is written in JavaScript with specific features delegated to other software packages.

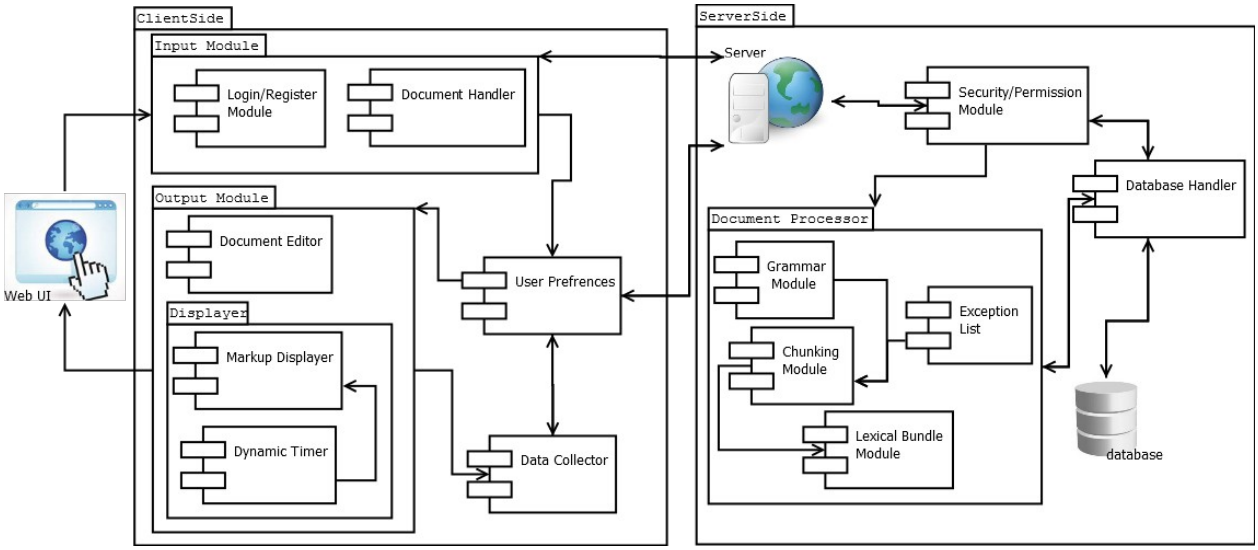

Figure 1. Major Functional Component Diagram

The NGINX software server is the initial server on the website it redirects web browser

requests to the Node.js server. The server.js file is run under Node.js and serves the application to the users web browser. Node.js maintains an open connection to the browser allowing the SPA to render an updated view as required by the interaction with the user. It is not the purview of the development team to perfect the complex task of natural language processing. Incoming reading samples are passed to the Document Processor. The Document Processor uses third party software package Natural Language Tool Kit (NLTK) to identify the parts of speech. The NLTK output is passed to custom code to parse the output into a JavaScript Object Notation (JSON) data structure format. The data is compared to an exceptions list for phases that the slashing algorithm could misinterpret. This list can be updated by the instructor when additional exceptions are observed to be repeatedly processed incorrectly. The Document Processor returns the JSON object to the server.js application. The application then relays the JSON data structure to the UI. This JSON data structure passes through re-parsers within the UI to incorporate the color coding values and slash locations depending on the viewing mode. The instructor reviews the automated output for correctness and preserves the sample for future use. Upon selecting “preserve sample” the SPA archives the instructors sample into the MySQL database running on the same server as the SPA. Figure 2 is a simplified diagram of the process flow described.

THIS SPACE INTENTIONALLY LEFT BLANK

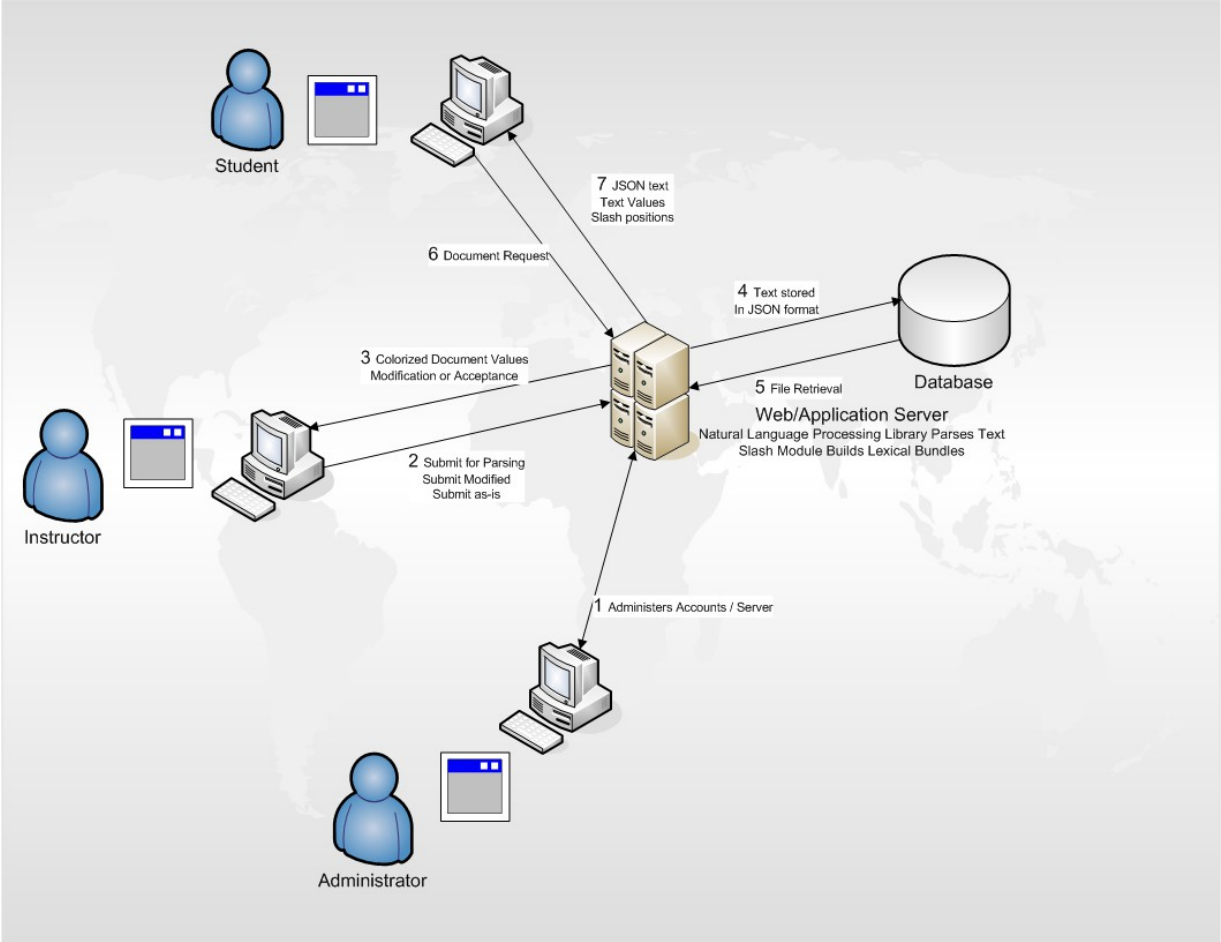

Figure 2. Process Flow

3 IDENTIFICATION OF CASE STUDY

Development of the CLASH software package stemmed from discussions with Mr. Greg Raver-Lampman of the English Language Center (ELC) at Old Dominion University. The ELC shall serve as the test bed for the continued development and enhancement of the software. The goal is to measurably increase reading speed of ELLs. The software's market potential however is incredibly diverse. According to a study performed in 2005 there were an estimated five million ESL students enrolled in American public schools grades K-12. The same study reports drop-out rates among this population to be four times that of their peers. (McKeon) Initial development is

for the English Language Center (ELC) at Old Dominion University at the students in the ESL program. However, expansion out of the university level ESL market into the primary and secondary education level ESL markets is entirely feasible. Further expansion into native English course work especially at the primary level is also possible.

4 CLASH PROTOTYPE PRODUCT DESCRIPTION

Due to time constraints on prototype development. The prototype CLASH software is designed as a proof of concept only. The prototype shall demonstrate the the core features of the real world product but with reduced functionality. The architecture remains the same to allow continued development into the full real world product as described in section 2 of this document. Elements of the user interface and component modules within the server relating to tracking student progress statistics are omitted. Table 1 explicitly lists the features present in the full product not included in the prototype. The risk of project failure has been mitigated by continuous dialog with Mr. Raver-Lampman to ensure requisite core features and minimum expectations were met or exceeded.

THIS SPACE INTENTIONALLY LEFT BLANK

Features	Real World Product	Prototype
Parsing Capabilities	Ability to Parse different kinds of documents	Ability to parse text copy and pasted in text block
Text Modification	Ability to modify and store previously parsed documents	Ability to modify and store previously parsed documents
Color Capabilities	Ability to color chosen parts of speech using a JSON format and JavaScript functions.	Ability to color chosen parts of speech using a JSON format and JavaScript functions.
Slashing Capabilities	Ability to identify Lexical Bundles through the inserting of slashes.	Ability to identify Lexical Bundles through the inserting of slashes.
Displaying Lexical Bundles in a single bundle form	Ability to speed up, slow down and pause Lexical Bundles being displayed.	Ability to speed up, slow down and pause Lexical Bundles being displayed.
Exception list	Lists of commonly used expressions that would otherwise be incorrectly handled by the SLASH Algorithm.	Lists of commonly used expressions that would otherwise be incorrectly handled by the SLASH Algorithm.
Login interface	User Authentication in a stand-alone environment	User Authentication in a stand-alone environment
Student Data Reporting	Tracks individual and collective student progress. To include words per minute, total time and total Lexical Bundles. Data to be stored in database. Displayed in graphs and statistics.	Limited basic student metrics shall be available such as Lexical Bundles per Minute.
Homework Mode	Instructors have the ability to remove coloring of words and have students correctly identify the part of speech.	Not Included.
Administrative Privileges	Administrators are able to edit, add, or remove users or saved documents in the system.	Administrators are able to edit, add, or remove users or saved documents in the system.
SLASH Document Viewing Mode	Ability to view documents with slashes inserted and SLASH Reader.	Ability to view documents with slashes inserted and SLASH Reader.

Table 1: Real Product vs Prototype Feature Comparison

4.1 Hardware and Software Prototype Architecture

The server used for prototype development is a virtual machine provided by Old Dominion University. To allow the continued development into the full product it uses the same software packages as the full version namely: Ubuntu Server, NGINX server, Node.js server, MySQL database, the NLTK, and software written to coordinate the interaction from users web browsers with the component modules.

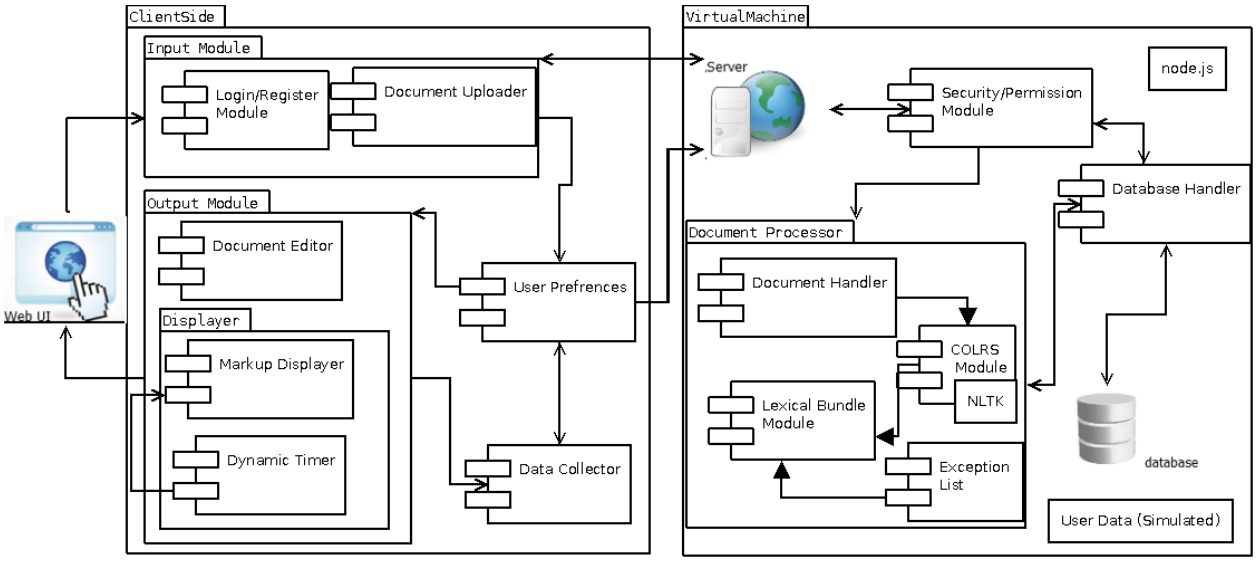

Figure 3. Prototype Major Functional Component Diagram

4.2 Prototype Features and Capabilities

Table 1, identifies the limitations of the prototype. The prototype shall parse text entered to a form on the web-page, but will not accept file uploads. In order for the students to access the reading samples they must be stored in the database and this behavior shall not require modification to implement the full product. The prototype will store the samples in JSON format and interpret them for rendering as a color coded sample or a slashed reading stream in the web browser using the JavaScript programming of the SPA. User authentication is provided to identify particular students and the database includes data structures to handle the futere implementation of progress statistics for the full product version. The only statistic implemented per student in the prototype version is lexical bundles per minute. The homework mode allowing students to select the part of speech from uncolored text is not implemented. Both the slash stream mode and the paragraph mode for slashed text is available to both Instructors and Students.

4.3 Prototype Development Challenges

Figure 3 shows the major functional components of the system and highlights the over all complexity of interactions between separate software components. From this complexity the system faces a variety of challenges to its viability. The challenges to develop this software include failures of the natural language processing system, the unfamiliarity of the developers with the required software package components, modules intentionally omitted from the prototype being expected or required by other modules, the building of the exceptions list, and other as yet unforeseen issues.

The NLTK software is fairly accurate in determining parts of speech but phrases such as “banana pudding” are “adjective noun” rather than “noun noun” as NLTK would indicate. The design by implementing human corrections to the automatically generated data mitigates this issue. Errors in the slashing procedure will stem from the same situation and again are resolved through the use of human intervention. The developers are rapidly learning the requisite software however mistakes in implementation will require resolution before each module can correctly interact with the others. Modules omitted in the prototype shall require either manufactured data as place holders during prototyping or completion of components to finalize the development of the real world product.

THIS SPACE INTENTIONALLY LEFT BLANK

GLOSSARY

CLASH - Color Lexical Analysis algorithm and Slash Handler

Client Side – The user-interface of CLASH

COLRS – Colored Organized Lexical Recognition Software

Document Processor – A Server Side component responsible for processing the text entered by an Instructor user type.

ELC – English Learning Center

ELL - English Language Learners

ESL – English as second language

GUI - Graphic User Interface

HTML - HyperText Markup Language

IBT – International Benchmark Test

Intensive English Program – A short and intensive English language training program offered by US colleges and universities to improve the English language skills of international students who did meet the minimum TOEFL scores for typical enrollment.

JS – JavaScript

JSON – JavaScript Object Notation. A nested data structure commonly used to pass data between a server and a client.

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

MFCD – Major Functional Component Diagram.

NLP – Natural Language Processing

NLTK – A suite of libraries and programs for symbolic and statistical natural language

processing (NLP).

Node.js – an open source, cross-platform run-time environment for server-side and networking application.

POS – Parts of Speech

Server Side – The back-end of the CLASH system responsible text processing, the database, user-authentication, and web-hosting.

SLASH – Aspect of CLASH that displays slashed text

Slash Player – Aspect of CLASH that displays a text stream showing one lexical bundle, of three to five words, at a time with the feature of speed control for display time.

Software as a Service (SaaS) – Software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet.

Token: Text that has been processed into individual words by the Document Processor

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.

Spreeder – Speed reading tool; www.spreeder.com

TOEFL – Test of English as a Foreign Language

Ubuntu – a Debian-based Linux operating system

VM – Virtual Machine

THIS SPACE INTENTIONALLY LEFT BLANK

REFERENCES

Engelbrecht, K. (2003, June 18). The Impact of Color on Learning. Retrieved February 25, 2015, from http://sdpl.coe.uga.edu/HTML/W305.pdf

Hoffman, D. (n.d.). Academictips.org - Reading and Highlighting Tips. Retrieved February 25, 2015, from http://www.academictips.org/acad/literature/readingandhighlighting.html

Lab 1 - Old Dominion University Fall 2014 CS 410 Team Blue

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

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

Professor Greg Raver-Lampman

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.

Open Doors 2013 Report. (2013, November 11). Retrieved from http://www.iie.org/Who-We-Are/News-and-Events/Press-Center/Press-releases/2013/2013-11-11-Open-Doors-Data

U.S. Department of Homeland Security, U.S. Immigration and Customs Enforcement. (July 2014) SEVIS by the Numbers General; Summary Quarterly Review. Retrieved from http://www.ice.gov/doclib/sevis/pdf/by-the-numbers1.pdf