Luis Alonso (lra2103)
Hila Becker (hb2143)
Kate McCarthy (km2102)
Isa Muqattash (imm2104)

COMS W4115-001
September 26, 2005

The CEAS Language
Project Proposal

Introduction

In the early days of the Internet it was easy to surf through interesting sites without having to deal with clutter. Website operators posted their information with simple formatting and a few representative images. Since then, browsing favorite websites has become more challenging thanks to the proliferation of easy to use animation programs, interactive images, and advertising techniques. For those who still use a dial-up connection this means spending more time downloading and less time browsing. While those with high-speed internet connections spend their time avoiding irrelevant information. The proposed language, CEAS, will alleviate this problem by providing an easy to use language that will allow the user to customize their browsing experience by removing unwanted information.

How It Works

The CEAS language provides a programmable interface to an existing web proxy named Crunch. Crunch is a highly customizable framework that can be used to extract data from HTML-formatted web pages. For instance, it can be configured to show a particular page without any images or to show just the information related to news. The CEAS language defines a simple and rich set of commands and operators that allow the user to leverage Crunch to enhance their browsing experience.

An end user starts by writing a script that defines at least one website they are interested in browsing. They then make use of simple operations to manipulate the pages they are interested in. Finally they run their script through our interpreter which will use Crunch to retrieve and modify the requested web pages. As a final step, the user can choose to write the results to a local HTML file for later browsing or they can use a built-in browsing system.

A Simple Example

Everyday Maria comes home from work and reviews a collection of websites from her favorite political bloggers. One day she realizes that there are more animated advertisements and annoying images then text on the page. She wishes that her browser was smart enough to get rid of the annoying images so she could focus on the important commentary. Maria goes in search of just such a tool and discovers CEAS can solve her problems.

Though Maria is not a trained programmer, she quickly discovers that CEAS will easily solve her problems. She writes her first script:

```
Page blog1;
blog1.url = "http://www.myblog.com/latestpost";
blog1.removeImages();
display(blog1);
show();
```

These four lines of code tell CEAS to build an internal representation of a webpage, remove any images, and then display the newly formatted page using an internal browser.

A More Complicated Example

Maria is fairly pleased with her initial success. However, she misses the ability to quickly load multiple tabs with her favorite websites as she used to do when she browsed with Firefox. She knows that CEAS has more functionality and decides to see if she can improve on her original script.

After doing some research she finds that she writes the following script:

```
PageList pl[3];
pl[0].url = "http://www.myblog.com/latestpost";
pl[1].url = "http://www.otherblog.com/latest";
pl[2].url = "http://www.newblog.com/";
for (int i=0; i < 3; i++)
{
        Pl[i].removeImages();
        display.addTab(pl[i]);
}
show();
```

These lines of code will allow Maria to define a list that holds three pages. She then moves through the list, removes images on each page, and adds the page as a tab to her display. Finally, she displays the entire environment.

Other Features

- Simple types
    - Integers: used for arithmetic operations, loop control and list indexing.
    - Strings: used to describe URLs and names of other Page attributes.
- Complex built-in types
    - Page: a type that contains attributes to describe a web page. These attributes include the target URL, genre name, title and preferred tab position.
    - PageList: a list of Page types. Includes a length() function as well as a next() function for iteration purposes. Elements can be assigned and accessed by specifying their list index (i.e. pl[2] = "foo").
- Appending "next" pages
    - A user can specify a keyword (i.e. 'next') as an attribute to a Page type, which is used to decide whether a link on the Page should be fetched and appended to the bottom of the Page. More specifically, if the keyword is found in the description of a link, the contents of the URL specified by the link are appended to the Page type. This is a useful feature for users who read various news sites in which the articles are spread over multiple pages.
- This language also provides functions to allow for common browser operations such as navigating forward and back.
- Users of the language cannot create their own data types, but rather use this language by assigning values and manipulating the built in structures. We decided to keep the language simple to ensure that CEAS would be used for its intended purpose. We decided on the specific functions and structures in order to give enough control to the user to accomplish the tasks of navigating the web and extracting web content.
- This language is platform independent and its only assumptions are that the user has an internet connection at the time of execution in order to retrieve the requested URLs, and a JVM.

- CEAS invokes a built-in web browser that supports tabbed browsing. This feature makes the language more convenient and easy to use, since the user does not need to install an external browser such as FireFox or Opera.

More About Crunch

The CEAS language makes use of a pre-existing web proxy called Crunch that extracts content from html web pages. Crunch is a pluggable framework that employs an extensible set of techniques for enabling and integrating heuristics concerned with "content extraction" from HTML web pages. Crunch parses the HTML of a given document and produces a Document Object Model tree to analyze a web page for content extraction. The Document Object Model (http://www.w3c.org/DOM) is a standard for creating and manipulating in-memory representations of HTML (and XML) content. By using a DOM tree, Crunch can not only extract information from large logical units, but also manipulate smaller units such as individual links. Crunch allows the user to select specific tags for extraction as well as predefined extraction filters such as "news" and "shopping", which are preconfigured for specific website genres. See figures 3 and 4 for a glimpse of the Crunch interface.

Sample Code and Output

Here is a simple way to extract content from a news article using our language. We choose to use the predefined extraction filters for the "news" setting provided by the language. We regard the "news" setting as the harshest setting that extracts all contents on the page except for the text. This setting is useful for reading news articles, when the user is only interested in the article's textual content. You can see the results in figure 2, while the page in figure 1 is the original web page that would have been displayed to a user not using our language.

```
Page p;
p.url = "http://www.cnn.com/2005/WORLD/meast/09/26/mideast.ap/index.html"
p.extract("news");
display(p);
show();
```

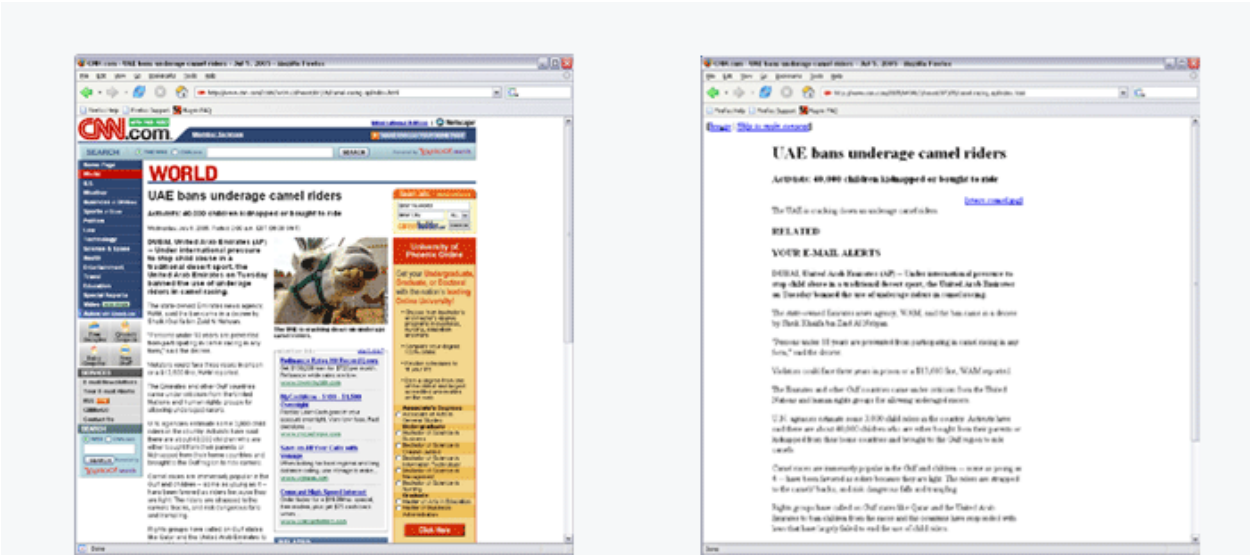

Figure 1: article, without CEAS

Figure 2: article, with CEAS

Conclusion

The CEAS programming language makes it fast, easy, and convenient to surf the web by automating the task of content extraction from various web pages. The syntax of CEAS is straight-forward, which makes the language easy to read as well as write. Simple and complex built-in data types are made available to the user, as well as functions that operate on the complex data types. In addition, CEAS provides loop structures and methods for list iteration. All these, combined with an embedded web browser, offer flexibility and allow the user to customize the content and view of web documents. Although limited in their number, the built-in structures that CEAS provides give a wide range of functionality for the user, making CEAS the language of choice for all of your web surfing needs.

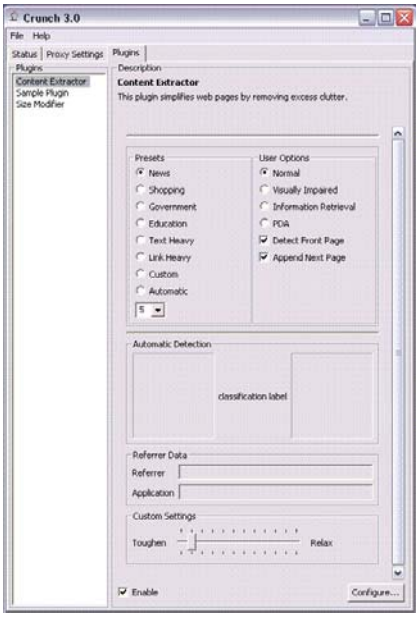

Figure 3: Crunch Content extraction UI

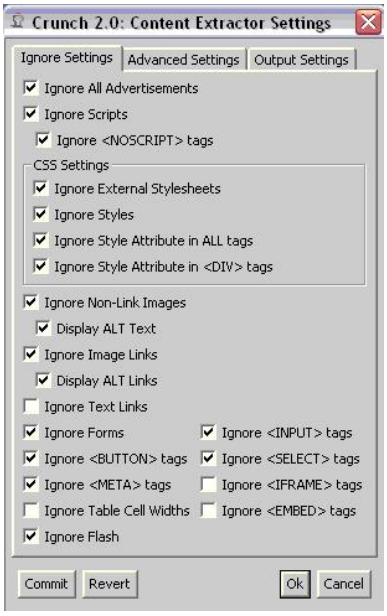

Figure 4: Crunch custom settings