

SHIL

(Simulated Human Interaction Language)

Moses Vaughan – mjev2123

Chun Yai Wang – cw2244

Binh Vo – bdv2112

Ian Vo – idv2101

SHIL

September 23, 2008

SHIL is a simple, strongly typed, object oriented, imperative language for simulating human interaction with web sites.

Introduction

SHIL is a language used primarily for developing HTML based automated bots. It provides the developer with an abstraction for automating interaction with both web sites and users. From the server's perspective SHIL can be used to simulate user interactions, which is useful for many applications ranging from creating spiders to website test scripts. From the user's perspective SHIL can be used to implement custom user interfaces. In conjunction with automated server interaction this potentially can be used to alter existing interfaces for websites or provide interfaces to additional functionality built on top of existing website functionality.

Motivation

Many automated browsing tasks are written now in various languages, primarily PERL and Python. For example many services such as web search engines need to crawl across existing pages on the internet, or independent users often wish to automate data collection over various sites. SHIL intends to provide a language designed specifically for this task which will reduce the complexity of writing applications of this nature.

Possible Applications

- Providing different user-customized interfaces to various web services
- Aggregating data from web-visible databases, such as craigslist
- Automated bidding utilities for auction web sites
- Writing test scripts for web interfaces
- Crawling web pages
- Automated user browsing over a large set of pages for the purpose of finding malicious sites

Sample Code

```
#Sample Program
#Provides a different interface to the user for Google searching

HTMLcode newInterface <= (ReadFile "documents/newInterface.html");

# DisplayHTMLcode creates local proxy, displays new interface in user
browser
# Intercepts next user http request and returns to program
HTTPRequest userResponse <= (DisplayHTMLCode newInterface);

# "<>" references associative arrays; HTTP requests will be associative
arrays
String query <= userResponse<q>;
HTTPRequest googleQuery <= "www.google.com/search";
googleQuery<q> <= query;

HTMLcode googleResult <= (Send googleQuery "get");
HTMLcode newResult <= (ReadFile "documents/newResultBegin.html");
foreach item in googleResult.body.ol[2]
newResult <= newResult + "<li><a>" + item.a.content + "</a>" +
item.content + "</li>";
end;
newResult <= newResult + (ReadFile "documents/newResultEnd.html");

(DisplayHTMLCode newResult);
```

This code shows how we might write a simple program that allows a user to provide his own customized interface to Google search. We begin by reading his own html search page and displaying it to him, and behind the scenes the interpreter will run a local proxy server to intercept the next HTTP request he makes from that browser, providing us with a way to query user interaction from custom interfaces. We then send this query to Google, parse the results, and display to the user another alternate search result pages of his own design.