Project Proposal

BELL

Columbia University
COMS W4115 Programming Languages and Translators

Fall 2007
Professor Stephen Edwards

Alicia Boyzk amb2129@columbia.edu
Yousry ElMallah yae2103@columbia.edu
Robert Lin rcl2106@columbia.edu
Carlene Liriano cl2294@columbia.edu

INTRODUCTION

BELL is an interpreted language created for the purposes of dynamic widget generation within the WEBELL framework. WEBELL is the online consolidated desktop system within which BELL works. WEBELL is a single website where users will manage all of their online widgets. Widgets range in a great variety and include the likes of Notes, Calendar, Weather, and Movie Times widgets. All WEBELL users will begin with stock set of BELL widgets upon account creation. Advanced users will then use the BELL programming language to customize the widgets and interoperability between widgets.

WEBELL is built on the Model View Controller framework which ensures an architectural pattern that divorces the presentation layer from all business logic. All widgets will be maintained in two files: the .bell file which details all of the business logic required by the widget, and a .web file which outlines the presentation of the widget. Note that the .web files will predominantly contain plain html with occasional ternary tags (containing the business logic) interspersed as necessary.

BACKGROUND

The World Wide Web is more or less the Wild Wild West of the technology world. A great multitude of "Standards" are everywhere with no truly efficient means of communication between disparate components and objects. While web applications have made esteemed jumps in the web world paralleling the performance and usability of many standalone desktop products, standalone widgets and useful components such as charts, graphs, tickers, and miscellaneous note utilities largely remain unconnected and thus underutilized. This is the problem we aspire to fix.

MOTIVATION

BELL was conceived for the single purpose of being the customizable glue and duct tape that makes the WEBELL system interesting. WEBELL is an open source framework that encourages users to build and share their widgets with the rest of the online community. BELL is simply the language that community speaks.

The motivation behind the WEBELL system is to encourage a single standard for interoperability of an eventual rich library of user-created content, in the form of widgets. Imagine a world where all widgets have direct line of communication with each other. A top ten playlist ticker widget can automatically connect to the Calendar widget and display concert times/album release schedules. An instant message window can trigger the sending of email. A friend updating his Christmas wishlist can automatically update your to-buy list. With WEBELL, the possibilities are endless.

FUNDAMENTAL FEATURES

WEBELL consists of a main base class widget from which all other future widgets will derive. WEBELL will also contain four prepackaged interfaces to be used by the user to customize their own widgets. These interfaces include:

Movie

Calendar

Top-Selling Books

Weather

Each interface contains basic method signatures to allow for communication between a widget, the WEBELL framework environment and other widgets. They will also contain a fetch method that will eventually contain the RSS source from which the widget can retrieve information from. All interfaces will have an export and import method to support interoperability. Furthermore, when users create a widget of their own they can model it after the prepackaged interfaces.

Other WEBELL features include classes, data types, reserved words, and control structures. WEBELL comment notation will be the standard e.g. //I am a comment.

DATA TYPES

Data types will consist of mainly those data types found in standard high level languages such as Java and C++. These include int, floats, strings, and lists. These will essentially have the standard functionality we've seen in the past.

Since WEBELL is not super strictly typed (but more strictly typed than, say, PHP, which allows dynamic typing), the list data type will be able to contain objects of different data types. The motivation behind this design decision is to allow for the interaction of different widgets via the list data structure. Lists will be used to pass information consisting of varying data types between widgets.

WEBELL will also include a date object. Date objects will facilitate the passing of date objects between widgets. For our purposes, the date data type will function similar to the PHP implementation.

The most significant feature of WEBELL will be the widget object. The widget object is the main base class. The widget base class will contain all of the basic functionality of a widget. It will also include PERMANENT methods which cannot be overridden by subclasses. Typically, the PERMANENT keyword will only pertain to methods dealing with the WEBELL framework connection and port connectivity. All user defined widgets must extend this base class object in

conjunction with a predefined interface specific to the widget. In other words, an instantiation of a user-defined Movie Widget would look like the following code snippet:

```
WEBELL Movie_Widget extends Widget implements Movie //Where Movie_Widget is
                                                    the class name
```

K E Y W O R D S

PERMANENT – Creates an immutable variable that is user-specified

FXN – The function declaration key word which contains the default constructor

EXTEND – The matter by which a new widget extends the base fxn (superwidget)

IMPLEMENT – When used, the IMPLEMENT keyword signals an incoming interface

WEBELL (class) – The declaration that indicates the start of a new object

CRAFT (new) – Instantiate a new class with the class's constructor if necessary

STEAL (import) – "borrow" convenient methods and fxns from libraries created by others

PRELIMINARY C O N T R O L S T R U C T U R E S & D A T A T Y P E S

(This section will be definitely expanded once development beings.)

If

else

switch

case

do

while

for

foreach

break

continue/next/proceed

String

Int

Double

List (Collection)

Widget

Date

eg. date('Y-m-d H:i:s', strtotime('now'));

. (concatenation)

+, -, *, / (typical math operators)

LANGUAGE LAYOUT

BELL essentially serves as the communication language of the WEBELL framework and thus requires an intermediary to interpret its instruction in between widgets. Furthermore, WEBELL (think of it as a simple operating system) needs to keep track of all concurrent threads and processes and is responsible for directing network traffic between widgets.

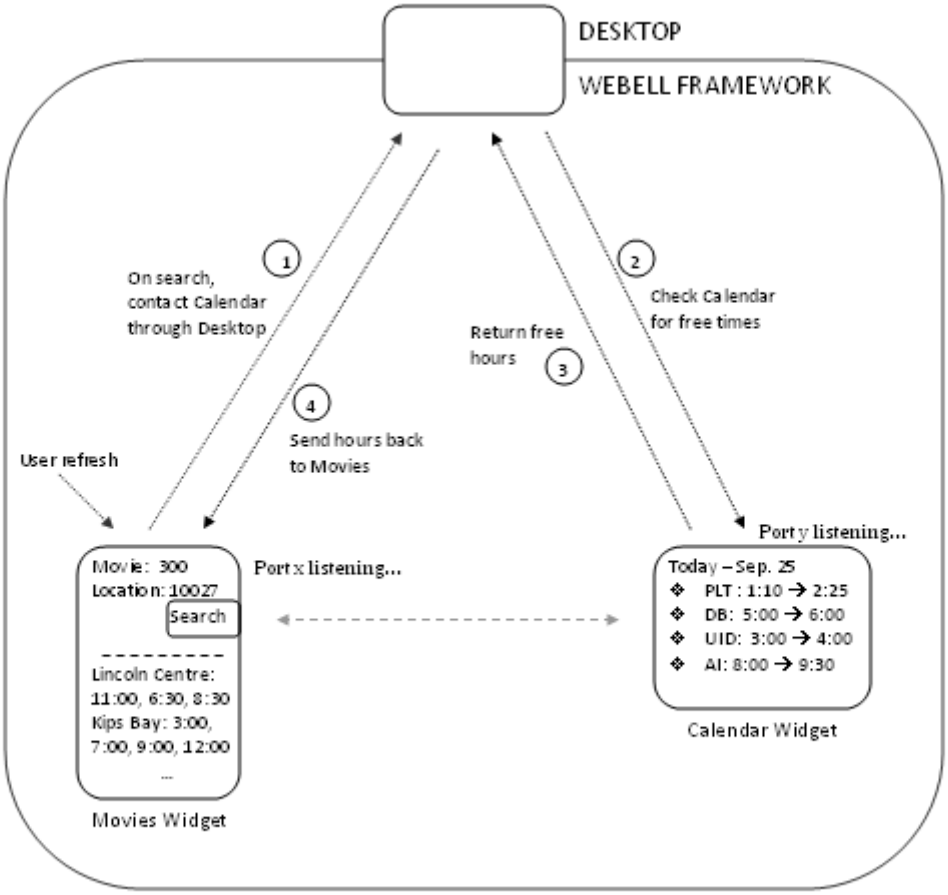

Figure 1a

CODE SAMPLE

```
//example of the MovieWidget:

import Movie;

String Constant MOVIE_LIST = "http://movierssfeed.rss";

FETCH myMovieTimes = new FETCH.get( MOVIE_LIST );

mmTimes.getTimes( September, 10020, "Superbad" );  //returns a key/value of
movie name(string)→ array of times

MovieWidget MW = new MovieWidget;
MW.import(myMovieTimes );
MW.display( 090107 – 090707 );  //month-month range
MW.sendTo( Widget Calendar );


//example of the MovieWidget interface:

Interface{

      String MovieName => { String Day1 => {(TIME1_START – TIME1_END),
                                            (TIME2_START – TIME2_END),
                                             ...                      }
                            String Day2 => {(TIME1_START – TIME1_END),
                                            (TIME2_START – TIME2_END),
                                             ...                      }
                          ...}
```

FUTURE DIRECTIONS

- We will need to develop the overarching online desktop environment in which the widgets live. This is not a trivial task and will indeed involve much work. Essentially, we need to create the environment first that will utilize the language.

- Given the nature of the project with its emphasis on interoperability between widgets, WEBELL and BELL will only become truly interesting once there is a rich library of available widgets for users to play with.

- Eventually we will want to expand the standard libraries of widgets that come with each WEBELL distribution.