

Proposal

Travel Assist Scripting Language

Yogi Saxena ys2332@columbia.edu

1. Introduction

1.1 Overview

There are several websites that offer quotes for air fare and other services such as lodging and rental cars. These websites do not provide tools and/or options that can be customized for such searches. The user is required to revisit the site and start a new search.

The Travel Assist Scripting Language aims to provide a simple yet powerful tool to search for latest fares/rates for desired services. The user could have the executable program run periodically and keep track of the various options.

1.2 Goals

1.2.1 Ease of Use

The main goal of this language is to keep the syntax simple so that any user without prior programming experience can use it.

1.2.2 Portable

As the end program will result in a java class file the program can be run across various operating systems such as windows, linux and Solaris.

1.2.3 Customizable

The users will have the option to either import java libraries or include the final .java file or class file in their own applications to add or enhance the features or applications.

2.0 Features/Syntax

2.1 Statement

A basic statement can contain reserved or conditional keywords, operators and form a complete instruction. A statement will always end with a semi-colon.

2.2 Data Types

2.2.1 Numeric Unsigned Integer will be used to represent quantity and date as 09012007. Also time will be represented as hours and minutes as 1824. No second's resolution.

2.2.2 Numeric Float will be used to represent the price as 256.55.

2.2.3 Boolean True or False for conditional evaluation of statements.

2.3 Reserved Keywords

Certain keywords will be reserved.

- 2.3.1 Search – To indicate the start of a new search.
- 2.3.2 Repeat – To repeat the search at a specified interval.
- 2.3.3 All upper case airport symbols ex: EWR will be reserved.

2.4 Conditional Keywords

- 2.4.1 If – else basic conditional statement.
- 2.4.2 For – Basic for loop.
- 2.4.5 While – basic while loop.

2.5 Operators

2.5.1 The following logical Operators will be supported.

Logical AND	and
Logical OR	or

2.5.2 The following Relational Operators will be supported.

Less then	lt
Greater then	gt
Equal to	eq
Not equal to	ne
Assignment	=

2.7 Functions

2.7.1 The following functions will be standard to the language:

- 2.7.1.1 Print – Can be used to output results to STDOUT.
- 2.7.1.2 Read - Can be used to read in user data in a defined format.
- 2.7.1.3 Email - Can be used to email the results to an email id.

2.7.2 Functions defined by users in will be supported.
functionName (arg1, arg2);

2.8 Grammar

The following grammar will be used in the language:

;	Semi-colon to mark the end of a statement
,	Comma to separate a list of arguments in a function
()	Left and right parenthesis to group the values of a function
[]	Square brackets for array representation
{ }	Curly braces for grouping a set of statements in a if else block

3.0 Operator Precedence

The operator precedence and associativity will be similar or same as the C language since its common and it's widely used.

3.2 Scanner and Parser

ANTLR 2.0 will be used to generate the scanning and parsing modules of the language.

4.0 Examples

Keyword Search followed by the airport codes to Origin and Destination followed the desired quantity and the dates.

```
Search EWR SFO 2 09012007 09052007
If ( Fare lt 250.00 )
{
    Email(ys2332@columbia.edu);
}
else
{
    Repeat;
}
```