R – A Scripting language for a call routing engine.

White Paper: COMS W4115: Programming Languages and Translators

Rajiv S Kumar

rajiv.kumar@verizonbusiness.com

Introduction:

Most call centers today use **ACD** (Automatic call distribution) technology to send incoming customer calls to Agents. Agents in a call center typically login to a Queue and wait for calls. The telephony switch used in the call center has ACD capabilities to keep incoming calls in queue and play messages to the caller until an agent becomes available. When agents are available, the call is sent to the agent who has been available for the longest time.

Most ACDs I know have limited scripting capabilities to customize call routing behavior. However, most ACDs do have the concept of a **"Route Point"**, where a call can be parked to let a third party application determine where to send the call. This third party application, called a **"Routing Engine"**, is typically custom coded in a high level language such as C++ or Java. Very few general purpose routing engines are available in the market and they tend to be proprietary and extremely expensive.

The goal of this project is to create a simple scripting language for a hypothetical general purpose routing engine. I'll call the scripting language "**R**" and call the routing engine "**R-Engine**". The scripting language **R** will have the basic features used in most call centers to let system administrators quickly customize their call routing behavior. I would like the R scripting language to have syntax similar to JavaScript, but variables would be strongly typed like in C.

The **R** Script would be triggered by a call landing at a Route Point. The ultimate purpose of every **R** Script is to specify a "target" where R-Engine should send the call. The R-Engine would have a library of inbuilt functions to provide access to things like Date/ Time, Call properties, Call Center Statistics, Database access etc. The **R** scripts should be able to work on the data returned by these inbuilt functions and ultimately specify a target. Just as JavaScript defines inbuilt objects such as Math and document, R would define an object called e (for engine) through which inbuilt functions would be accessed. The examples in the next section should make this concept clear.

Example Scripts:

Example 1:

```
begin

//The following sample sends calls to agents during office hours (9am – 5pm, Mon –Fri)
//and sends calls to voice mail for after hours
int d;
int hr;

//GetDayOfWeek is an inbuilt R-Engine function

d = e.GetDayOfWeek();

//GetHour is an inbuilt R-Engine function.
hr = e.GetHour();

//Assume days of week start with Sunday = 0

if (d >= 1 && d <= 5)  {
        if (hr >= 9 && hr < 17) {
                //The regular queue
                target = 8000;
                return;
        }
}
//Send calls to the voicemail.number.
target = 8001;

end
```

Example 2:

```
begin

//Sends the call to an English speaking agent if the caller wants English.
//Sends the call to a Spanish speaking agent if the caller wants Spanish.

string lang;

//CallProperty is an inbuilt R-Engine function that can return the properties of a call.
lang = e.CallProperty("CallerLang");

//StrComp is an inbuilt R-Engine function
if (e.StrComp(lang, "English") == 0) {
        target = 2222;
```

```
}
else if (e.StrComp(lang, "Spanish") == 0) {
        target = 3333;
}

end
```

Features:

R Script will have the basic features of a scripting language such as the ability to declare variables, loops, conditional statements, user-defined functions and the ability to call R-Engine inbuilt functions. The real power of such a scripting language will come from the ability to harness the R-Engine inbuilt functions to build sophisticated call routing logic that is tailor made to suit the call center's needs.

Implementation:

R Script will be implemented as an interpreted language. As of now, I think the ANTLR generated lexer and parser should be part of the R-Engine itself, so that the scripts can be directly fed to the R-Engine for execution. Since the main purpose of this project is to develop a scripting language, I plan to implement a simple R-Engine with a few inbuilt functions for testing purposes. The R-Engine will be developed with Java.