

Using Java and Erlang in Protocol Testing

Zsolt Werner

Even though there are established protocol testing methods and applications on mini computers (Sun), the Intel-based PCs lack any commercial application that would make it possible to either edit TTCN/MSC/etc... code on PCs or cross-compile code written in either language to any widely used programming language.

Both Erlang (<http://www.ericsson.se/erlang>) and Java (<http://www.javasoft.com>) are catching on pretty rapidly as two powerful new languages with a lot of potential, there is a significant problem in dealing with these languages, namely, the lack of protocol testing applications written in these languages.

The two languages can interface to native methods written in the other language. As both Erlang and Java offer advanced UI components that communicate with the end user, it's very easy to write Java / Erlang-based applications that can be used to edit TTCN/MSC code, print their class hierarchy, and all this in a highly portable form - the code runs on low-end PCs under Windows95/Linux, while for example, Proconsul, one of the leading products in this area, only runs on Sun workstations, and the same applies to most of the commercial applications devoted to this subject.

The goal for this paper is to introduce you to Java's/Erlang's UI (User Interface) programming and interfacing between the two languages - using the native methods written in the other language. If you are new to Java, you may not know what native methods are, and even if you are an experienced Java developer, you may not have had a reason to learn more about native methods. At the conclusion of this section you should have a better understanding of what native methods are, when and why you may want to use them, and the consequences of using them. You should also have a basic understanding of how native methods work. Of course, actual programming is not covered in the paper, it is not a tutorial, only our work is covered and ready-to-use result is presented.