

CBCJVM — Applications of the Java Virtual Machine with Robotics

Braden McDorman (braden@betabot.org)
Benjamin Woodruff (odetopi.e@gmail.com)
Jonathan Frias (freakinjonathan@gmail.com)

May 23, 2010

1 Thesis

The Botball tournament requires programmers to rapidly develop and prototype programs. However a disconnect is visible between the language and the tools provided and the goals of the game. Although very fast, C is generally considered to be a low-level language, making it what many of us believe to be the wrong tool for the job. The result is buggy programs, cluttered code, and a lack of flexibility in user libraries. The CBCJVM is the product of the realization that although slower, interpreted Object-Oriented programming languages are better for the type of prototyping used in the Botball competition. As Kipr has stated many times, it is highly unlikely that the CBC will ever be officially changed from C ¹. CBCJVM shares many of the same goals as Nease’s CBCLua, but it is aimed at a wider audience, with monthly releases, detailed documentation, and support for more than just one language.

2 Languages

The CBCJVM originally went under the name “CBCJava”, but as the project grew, it was soon realized that using the JVM only for Java was a waste. The Java Virtual Machine has become a de facto standard of virtual machines, and almost every interpreted language imaginable has some port to the JVM, and these languages typically have support for interfacing with Java libraries. The JVM is fast, taking advantage of Just-In-Time compilation and various other optimizations, making it’s performance somewhat comparable to C++ and C (but performance is still in no way better than a native language). CBCJVM has been used with JavaScript (via Mozilla Rhino) and Scala (which compiles directly to Java byte-code), but it can be used with many other languages such

¹<http://community.botball.org/forum/miscellaneous/suggestions-and-bugs/can-there-be-kiss-c-rather-kiss-c>

as Ruby, Python, Lua, and even LOLCODE. While it cannot match the performance of native interpreters for these languages, such as WebKit (JavaScript), CPython, or the standard Lua interpreter, it offers a fast way for these teams to get up and running with these high level languages. What's more, is that CBCJVM already includes a set of powerful libraries that can be used in conjunction with these languages to help teams that would otherwise be on their own with a language. (Not everyone can port and develop libraries in a language entirely by themselves and have it work well... *cough* Nease, CBCLua *cough*)

3 Access to the Kiss-C Libraries

Rather than being a straight port of the JVM to the CBC,