Dayton Gomez

CS 442

Language Proposal - Internal App

First let us look at the **primary requirements** we need to reach with a language:

1. The system must work reliably.
2. The system must work on a variety of systems.
3. The system must be extendable.

Note that for this argument I will only focus on 1 and 2. The system is simple enough that 3 shouldn't be an issue yet.

We will be looking at two languages: C/C++ and Java. Now my assumptions are thus:

* Performance is probably not an issue with either.
* They both *will* work on the current box.
* Our knowledge of any language, or lack thereof, has no sway in what language to use.
* All libraries we use must be free to use and modify without making us release our source code.

Both Java and C/C++ and Java satisfy primary requirement 1. The real different comes when analyzing the languages for requirement 2. I propose we use C/C++ because it satisfies req. 2 better than Java.

That being said, let us look at the pros and cons of both languages.

Java

Pros:

* All libraries we require (scheduler, sound play/record, zip/unzip) are built into the java API. They seem to be usable without any licensing issues.
* Works on most platforms once you install the JRE

Cons:

* Have to install the right version of the JRE to ensure the system works.
* Will this work on a tiny embedded system?
* Runs in java byte code instead of assembly

C/C++

Pros

* Lots of prebuilt libraries for whatever we need.
* Can be compiled into an executable that should work anywhere.
* Compiled into assembly.
* Should work on any system, even small ones.

Cons

* Libraries often have some licensing issues. Have to search for one with the appropriate license.

I propose we use C/C++ as our internal application language. Again I wish to point out that both languages can completely meet primary requirement 1. The difference between the languages is apparent on requirement 2. Yes, Java is built to be cross platform, but in our case we need to build a program that could run off of a tiny embedded system. C/C++ is inherently small. With C, we simply compile the application and run it. We then don't have to worry about JVMs or proper versions of JREs.

As pointed out to by Doctor Henry, we will probably have to make small modifications per OS to our system (start/stop scripts). Having it run within a JVM will not earn us as much abstraction as we may believe. If we don't receive any benefits from the JVM, what's the point in using Java?