David Schirduan 9-12-11
CSCI362 Bowring

The Future of Programming response

The article was a very interesting read. I must admit that I assumed that it had been written earlier than 2006, since some of the claims made seemed rather off-base. I know that it is beneficial to have an ultimate goal set, and to pursue improvement, but the article makes claims and predictions about the future that simply don't acknowledge the difficulties and complexities of programming languages.

The main problem in this article is that it seems to gloss over the differences between languages as if they are insignificant. This is a flaw, and a huge mistake. The differences between programming languages are extreme in some cases, with irrevocably different ways of tackling problems. Creating any sort of common library or inter-functionality would be difficult. Take Scheme and Java for instance (which we are covering in CSCI320): these two languages are very different. Java uses memory-based problem solving; things such as variables, classes, objects, etc. Scheme is a mathematical language, focusing on functions and order of operations. It would be very difficult to somehow combine these languages, or even create similar libraries for them. Creating a universal structure for every development language seems an insurmountable task.

Along that same vein is the software platform that he describes. With “ an intelligent IDE;support for writing GUI applications; support for writing Web applications; an effective virtual machine/runtime; and the ability to deploy to any operating system” among other things. Creating a universal IDE would be quite an accomplishment. Take Eclipse, for Java. It is an excellent IDE, with corrections, class diagrams, GUI design, testing, and many more features that work fairly well together. Even so, it is bloated and has quite a learning curve. Imagine the complexity of doing that for multiple languages!

I don't think that the future of software lies in simplification, but in elimination. One thing that David Hansson said perfectly was “The rise of open source is causing this. Along with this transformation, we'll see the open-source, dynamic languages start displacing their enterprisey counterparts." I agree with this. As the internet becomes more widespread, and computer fluency rises, open-source will have more developers than ever. Especially open source alternatives to commercial software. As software projects like Libreoffice and Ubuntu grow more popular, people are taking more of an interest in open source software. It follows then, that open-source programming languages would also have the same appeal as the OS software for the same reasons; flexibility, reliability, community centered testing, lack of regulation, and cross-platform usability. As open-source programming languages emerge, the strongest and most effective ones will rise to the top. Think of it as a Darwinian process.

This will automatically result in common libraries, efficient IDE's, and stable software platforms for development. I truly believe that Open-Source is the future, of software, operating systems, and programming languages. As long as the OS community continues to grow in strength and reliability, it will create a substantial competition for Commercial Software. When someone who is mildly familiar with computers can go 5 years without purchasing a single piece of commercial software, then the merits of Open-source and free software must be considered when discussing the future of development and programming languages.