Crossroads
The ACM Student Magazine

Editorial

by ***Saveen Reddy***

Many animals can communicate. Mostly, this communication occurs through various signals that alert others to danger, warn away enemies, call for help, or indicate the location of food. These signals may be vocalizations or a series of movements. There is also evidence to suggest that some animals have the cognitive ability to form simple sentences. For example, although apes lack the necessary vocal apparatus to speak, they can form simple sentences and compound words in sign language. However, this communication lacks the expressiveness of which even small children are capable. Thus, we find ourselves -- all five billion of us-- alone in the animal kingdom.

We humans have extended the languages we use to communicate with each other to serve as control mechanisms for ``communicating'' with machines. In the most basic sense, we even communicate with our toasters. We provide input (bread), fiddle with some controls (a symbol system), and receive something warm and, when I make it, lightly-browned in return. However, our communication with computers is more obvious. Here, I am not talking about the various efforts to make computers understand natural language. Instead, I am speaking of programming languages.

It has been proposed that language affects its speaker. Specifically, the claim is that it influences -- some would say determines -- the speaker's perception of his or her environment. According to this view, differences in language give rise to differences in behavior. I do not subscribe to this view. It seems likely that our culture is a stronger determinant of our behavior than our language.

What then can we say about programming languages? On the face of it, we know that

most programming languages are Turing-equivalent. In fact, they must eventually be executed in terms of machine instructions. In principle, what is computable in one language is computable most others. Yet, who would want to write a large database in FORTRAN or an OS in Smalltalk? Yes, these are pathological examples. Still, they illustrate the point that the problem or the environment may dictate our choice in language.

I contend that programming languages haven't really changed much over the years. I'll even go on record and say that I'm not that impressed with object-oriented languages at all, particularly the beautiful, yet grotesque C++ language. I feel that these languages are more than adequate for problems faced in the past, but not powerful enough for the present, much less the future. For example, it does not seem sensible to use C++, a textual language, to create a graphical user interface.

In the coming years, I believe our understanding of what comprises a programming language will change dramatically. I am confident that future programming languages will be better suited to problems like GUIs. I'm even more confident that there will be a greater number of very domain-specific languages. In this issue of *Crossroads*, we'll examine languages. We will touch upon some conventional topics, explore a small fraction of a bright horizon, and, in the end, perhaps we will see a kernel of the future in these pages.