Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080312

Modeling the use of Programming Languages: a very Simple Approach

J. C. Chimal-Eguía[*] *and A. Chavez-Valle*

Centro de Investigacion en Computacion, Instituto Politecnico Nacional U. P. Zacatenco, Mexico D.F., Mexico

Received: 14 May. 2013, Revised: 18 Sep. 2013, Accepted: 19 Sep. 2013
Published online: 1 May. 2014

Abstract: In this paper a very simple model for the dynamics of two programming languages is presented. Considering two programming languages X and Y respectively, it is possible to explain in relation to time the obsolescence of some programming languages and the future trend of those most used. Also, it is possible to quantify the level of popularity or decadence by using a parameter related with the use of language. With the aid of the model it is possible to predict future trends of languages, something very important when we have to take some decisions related to the investment in education related to the teaching of certain programming languages or the programming of industrial applications. Thus, depending on the election of a specific language and its prevalence in a specific area a company or institution is going to be able to better navigate in the future.

Keywords: Programming Languages, Simple Approach

1 Introduction

Currently, there are a large number of programming languages. In the timeline of programming languages there have appeared around 2500 computation languages [1]. However, nowadays not all are used; instead only few of them have become relevant (popular) in the context of programming languages. For example, according to Sammet J. [2],[3] over 200 programming languages were developed between 1952 and 1972, but she considered only about 13 of them to be significant. Some of them appeared more than 50 years ago and continue in use today. Its usefulness accords to the interest of the people and the constant development of technology, while others tend to disappear. In 2001 Strogatz S. et al. [4] presented a very simple model in which they described idealized human languages as fixed, and as competing with each other for speakers. For simplicity, the authors also assumed a highly connected population, with no spatial or social structure, in which all speakers were monolingual. One of the Strogatzs paper shows that the model therefore predicts that two languages cannot coexist stably and one will eventually drive the other to extinction. However, in real societies there exist bilingual countries although separated in monolingual groups there is no competition between the two groups maintaining the idea of any coexistence between the languages. Following the ideas of Strogatz et. al. [4] in the present article we present a simple model very similar to that presented by Strogatz et al. but considering programming languages instead of human languages. The model in our opinion could explain, in relation to time, the obsolescence of some programming languages and the future trend of some of the most used. A technological indicator, obtained from the model, which quantifies the level of extinction can be used to design and evaluate the popularity of the programming language used. Other models have focused on issues such as development, maintenance releases. However, in our model it is assumed that there is a programming environment in which programmers use at least one programming language. The article is organized as follows: Section 2 shows the model that represents the interaction between to programming languages. Section 3 presents some results of the model and finally Section 4 yields some concluding remarks.

2 Modeling the interaction between two programming languages

Consider a system with two programming languages X and Y, in which the preference for one of them, according

[*] Corresponding author e-mail: jchimale@ipn.mx

to the number of developers who use it and the opportunity to use it in different strata, detracts from the scope of the other language. The model proposed here shows the language popularity trend, which leads others to reduce theirs. Thus, suppose an individual converts from language Y to X with the corresponding rate $\alpha_{xy}(x,s)$ where x is the fraction of the population which use the language X (normalized to the size of the population), s is a measure of relevance of the programming language used for computing development of systems, where $0 < s < 1$, and y is the complementary fraction of population using language and time t, given by $y = 1 - x$. A minimal model of programming language change is,

$$\frac{dx(t)}{dt} = y\alpha_{xy}(x,s) - x\beta_{yx}(y,s) \tag{1}$$

where y is the fraction of the population which uses the language Y and again an individual converts from the language X to Y with a corresponding rate $\beta_{yx}(1-x,s)$.

The exchange of one language to another, by symmetry, should happen at the same rate as an exchange in the fraction of programmers and language proficiency, thus,

$$\alpha_{xy}(x,s) = \beta_{yx}(y,s) = \beta_{yx}(1-x,1-s) \tag{2}$$

Besides, we also assume that nobody will adopt a language that has no programmers, this is $\alpha_{xy}(0,s) = 0$, or the language has no relevance to develop systems, i.e., $\alpha_{xy}(0,s) = 0$ and α_{xy} is smooth and monotonically increasing in both arguments. From the stability analysis[5], we observe that we have three fixed points $x_1 = 0$, $x_2 = 1$, x_3, the last one depends on the value of rate $\alpha_{xy}(x,s) = 0$ and the rate $\beta_{yx}(1-x,s)$, the other two are stable fixed points and from this we can predicts that two languages cannot coexist stably (one will eventually drive the other to extinction).

For testing purposes we consider two programming languages X and Y, and collected data on the number of developers who use and adjust the solution according to the following rates;

$$\alpha_{xy}(x,s) = cx^a \tag{3}$$

(3) and

$$\beta_{yx}(1-x,s) = d(1-x)^a(1-s) \tag{4}$$

where the constant a approximates the mean value of the language programmers by $a = \bar{x} \pm \sigma$, where $\bar{x}$ is the average percentage of the population program in language X and σ is the standard deviation.

3 Comparison between two programming languages: some interesting results

Our first interest is in the following languages: FORTRAN and C / C++, which historically are two of the most important languages used in the area of Computer Sciences, but our study is not limited to them. The Popularity data of each programming language was obtained from the TIOBE index [6], and depicted in figure 1 and figure 2 for FORTRAN data and C / C++ languages respectively.

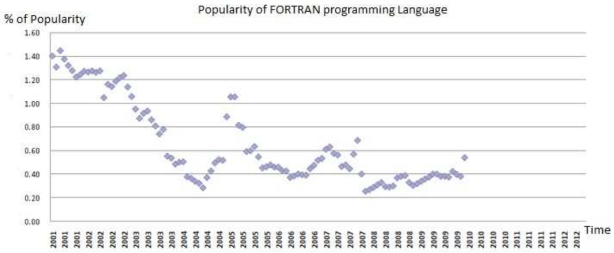

Fig. 1: Tendency data obtained from the TIOBE index for FORTRAN.

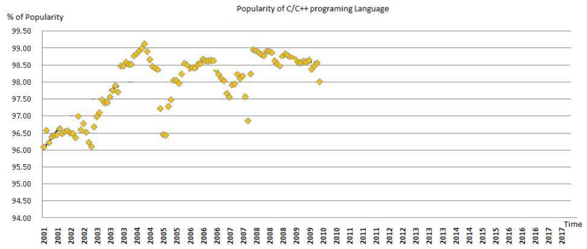

Fig. 2: Tendency data obtained from the TIOBE index for C/C++.

Data available since June 2001 to August 2012 takes the registration period monthly. From this data base, FORTRAN provides data until February 2010 and for C / C++ provides data until August 2012. Data from figures 1 and 2 are normalized by percentage, where for each language, a fraction from the total sum of significance is recorded for all programming languages, i.e., for the purpose of this work we are considering that the popularity of FORTRAN plus C/C++ represents 100% of the sample.

As a case study we proposed a significance of $90\%(s = 0.9)$ with respect to X and a value of $a = 0.46$ (mean $\pm$ standard deviation) and $c = 0.85$. It is important to notice that when we select a specific significance or popularity (s parameter in our model), our intention is to describe the usefulness for language programming. We could use other values of the parameter but for this special case we obtain the best data fit.

Therefore, when these values are substituted in equations (3) and (4) and then these rates applied to equation (1), it is possible to obtain the behavior of each of the languages by using the Runge-Kutta numerical method of 4^{th} [7]. These results are shown in figures 3 and 4.

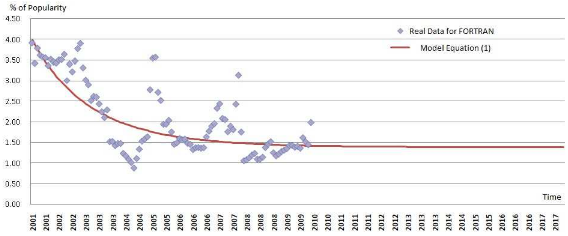

Fig. 3: Real Popularity data from TIOBE index vs the model represented by equation (1) for the FORTRAN language.

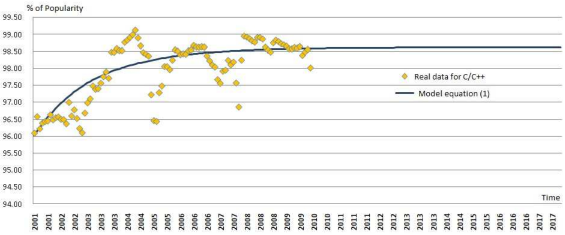

Fig. 4: Real Popularity data from TIOBE index vs the model represented by equation (1) for the C/C++ language.

4 Analysis and some concluding remarks

We have developed a very simple model which represents the dynamics of two of the most important programing languages (FORTRAN and C/C++) and the model could explain in a reasonable way the tendency of both languages in recent years, and future trends by taking into account two parameters, a (mean value of the language programmers ± standard deviation) and s (significance or popularity of the language). The exponent a is an important element in the preference of developer analysis, according to the determined values. The s estimator is an element that affects the choice of a specific language and can be used as the preference factor of this choice. For example, in the case of FORTRAN some sectors are still using it depending on the field of programming, but the status is increasingly weakened with respect to C / C++, tendency that is observed in Figures 3 and 4. Perhaps, some FORTRAN programmers may have problems developing C / C++ or to venture into the use of the latter with the programmers of C / C++. The latter is observed in our model in which the tendency observed until 2017 shows a minimum number of FORTRAN programmers. Moreover, the number of programmer is stabilized around 1.5% of all the universe of programmers (FORTRAN plus C/C++).

Contrary to the model's stark prediction, bilingual programmers do, in fact, exist. But in educational institutions it is not a common situation to teach two programming languages, so there is no real competition between programmers, instead the competition is related to the respective use of the programming language. Generally programmers involve split populations that have lived without significant interaction, effectively in separate, monolingual programming communities. The TIOBE index shows many different languages and we tested our model with other languages, for example MATLAB vs C/C++, in which the same tendency as our previous example is shown, but the value of the parameters changed. In our case the best value that fits to the tendency behavior is $s = 0.9$ and $a = 0.46$ (mean ± standard deviation obtained from the TIOBE index). Thus, it is natural to think about a significance of 90% as in the computational universe people prefer to program in C/C++ rather than in FORTRAN.

Although some programming languages are still used in a practical sense with respect to the respective field, increasingly making decisions around languages is more popular today. The question is related to the possibility of continuing to invest or not the use of resources for programming languages that tend to be obsolete.

Acknowledgement

We would like to thanks CONACYT, EDI-IPN, and COFAA-IPN for supporting this work.

References

[1] B. Kinnersley, http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm, The Language List.

[2] E. Sammet, Programming Languages: History and Fundamentals, Prentice-Hall, (1969).

[3] E. Sammet,"Programming Languages: History and Future", CACM, **15**, 601-610 (1972).

[4] Daniel M. Abrams and Steven H. Strogatz. Modelling the dynamics of language death. NATURE, **24**, 900 (2003).

[5] Non Linear Dynamics with applications to physics, biology, chemistry and engineering Steven Strogatz., Perseus Publishing, New York, (2000).

[6] TIOBE, http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html, TIOBE. The TIOBE programming comunity index.

[7] Richard L. Burden and J. Douglas Faires Numerical Analysis. Brooks/ Cole Cenage Learning. Boston, USA, (2011).

J. C. Chimal-Eguía He obtained the title of Bachelor of Mathematical Physics at the ESFM-IPN in 1994, the master's degree with a major in physics in 1996 at the ESFM-IPN and the degree of Doctor of Science degree in Physics in ESFM-IPN in 2003. Postdoctoral Fellow at the University of Alberta Canada. Currently holds the rank of professor " C" in the Computer Research Center of the IPN in Mexico. He has authored more than 50 articles in journals and international conferences. His interests are the modeling of complex systems and nonlinear dynamics. Dr. Chimal-Eguia is fellow of COFAA, EDI and member of the National Research System (SNI) level I of Mexico.

A. Chavez-Valle He obtained the title of Bachelor of Engineer in Computer Sciences at the ESCOM-IPN in 2006, the master's degree with a major in computer sciences in 2011 at the CIC-IPN.