

# Towards Development of an Indigenous African Language-based Programming Language

\*<sup>1</sup>Ezekiel K. Olatunji, <sup>1</sup>John B. Oladosu, <sup>2</sup>Odetunji A. Odejebi and <sup>1</sup>Stephen O. Olabiyisi

<sup>1</sup>Department of Computer Science and Engineering, Ladoké Akintola University of Technology, Ogbomoso, Nigeria

<sup>2</sup>Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria  
{[ekolatunji](mailto:ekolatunji@loutech.edu.ng) | [johnoladosu](mailto:johnoladosu@gmail.com)}@gmail.com | [odejebi@yahoo.com](mailto:odejebi@yahoo.com) | [soolabiyisi@loutech.edu.ng](mailto:soolabiyisi@loutech.edu.ng)

**Abstract**— Programming languages based on the lexicons of indigenous African languages are rare to come by unlike those based on Asian and / or European languages. It is opined that an African native language-based programming language would enhance comprehension of computer-based problem solving processes by indigenous students and teachers. This study intends to attempt a design and implementation of an African native language-based programming language using Yoruba as case study. Yoruba is the first language of over 30 million people in the south-west of Nigeria, Africa; and is spoken by over one hundred million people world-wide. In preparation towards actual implementation of a prototype of the intended programming language, a mini token recognizer has been developed in QBasic.

**Keywords**— Native language-based programming languages, Yoruba language, Digital divide, Information and communication technology, prototype implementation.

## 1 INTRODUCTION

Programming languages (PLs) are notational systems for communication with computers. More specifically, they are notations for description of algorithm and data structures for computers and people. The design and evolution of programming languages (PLs) have always been influenced by a number of factors such as discovery of weaknesses and/or deficiencies in the existing PLs. For instance, the first electronic computers appeared in the 1940's and were programmed in machine language – sequence of zeros (0's) and ones (1's) (Aho et al, 2007). Programming in machine language was very slow, tedious and error-prone. These problems led to the development of mnemonic assembly Language in the early 1950's, and this notation was more programmer-friendly (Aho et al, 2007). High Level Languages (HLLs) such as FORTRAN, Pascal, which in principle, were designed to be machine-independent, were developed as solution to the programming, debugging, maintenance and portability difficulties that were associated with the machine language and assembly languages (Olatunji, 2006). The first widely known and successfully implemented HLL was FORTRAN. It was implemented as a compiled language in 1957 (Sebesta, (2012).

Evolution of language and its implementation have further been influenced by such things as development in computer hardware; requirements of new areas of application; changing understanding of better methods of writing and maintaining large and complex programs; understanding of strength and weakness of some language features and need for standardization (Pratt & Zelkowitz, 2001). Most of these invented (programming) languages borrowed their lexical items from human languages (Gabrielli & Martini, 2010). Thus programming languages have been developed based on lexicons of Asian languages (such as Rapura, Ezhil) and European languages (like Java, Pascal, COBOL, etc).

In fact, it is observed that most of the existing popular high-level programming languages (such as Visual BASIC, Java, C++ and others), especially in the continent of Africa, use the English language lexicons. However, there is little or no research information on serious attempts on developing programming languages from the lexicons of African indigenous languages.

An important factor necessitating embarking on this study is thus the need to naturalize / indigenize computer programming by making programming in African mother tongue possible so that problem-solving process using computer by indigenous students and teachers can be better facilitated as literatures (UNESCO, 2007; Pflapsen, 2011) have shown. Furthermore, those who are only literate in their native languages can also be given the opportunity to learn how to program the computer system as their counterparts who are literate in English language.

## 2 NEED FOR PROGRAMMING LANGUAGES IN NATIVE LANGUAGES

The need for non-English-based programming languages cannot be over emphasized. Without any doubt, there are myriads of languages all over the world aside the English language. In fact, it has been estimated that a total of about 5000 – 7000 languages are presently spoken around the globe (Benson, 2006) and that the African continent is the home of about one-third of these world's living languages (Alidou et al., 2006). There are also many people, in the order of millions, who are only literate in their mother tongue. According to the national literacy survey carried out by the National Bureau of Statistics (NBS, 2010), 13.7% of adult literates (age 15 and above) in Nigeria are only literate in their mother tongue. By the estimated population of Nigeria (155 million) as at the time (year 2010) the survey was conducted over 13 million people are only literate (being able to read and write) in their mother tongue in Nigeria alone. These group of people, who are inadvertently being denied opportunity of learning how to program the computer for lack of proficiency in English language,

\* Corresponding Author

need to be empowered to do so; and this by making programming possible in their mother tongue.

In an era of increasing globalization, mother tongue computing (MTC), which will include development of programming languages with native language lexicons, will be a step in the right direction in bridging the digital divide between rich and poor countries and between the rich and poor populations within countries (Wagner et al., 2010). It is also one of the effective strategies in improving access to ICT (Pflepsen, 2011). It is now a common knowledge that mother tongue-based instruction is fundamental to achieving the Education for All (EFA) and millennium Development Goals (MDG) (UNESCO, 2007). In a similar vein, mother tongue-based programming languages will be an important strategy to achieving Computing for All (CFA) goal in the not too distant future, especially in developing and under-developed counties of the world.

Research results in mother tongue education (MTE) and instruction (MTI) abound to show that mother tongue-based instructions greatly improve learning and attainment of academic success and goals (UNESCO, 2007; Pflepsen, 2011; Alidou et al., 2006; Ball, 2010; Pamela & Walker, 2011). By extension, it can be inferred from these research results that MT/NL-based PL will greatly facilitate computing programming especially among myriads of people who are only literate in their mother tongue. In a dispensation of increasing globalization coupled with a prediction that as many as 90 – 95 % of the world's spoken languages, including most of the African indigenous languages, may be facing extinction before the end of the present century (Assembly of Nation, 2005; Benson, 2006; Awobuluyi, 2014; Azeez, 2013; Fabunmi & Salawu, 2005); development of a MT-based PL will be an indispensable means of preventing the predicted linguistic genocide, and thus saving the indigenous languages.

### 3 RELATED WORKS

Historically, the vast majority of keywords, libraries, and other building blocks of computing have been based on English. Of the 8500+ recorded programming languages on the HOPL (History of programming languages) online database of languages, More than 3000 were developed in the Anglosphere, with the United States making up the lion's share at roughly 2400 (Mao, 2015). This is followed by 600 in the United Kingdom, 160 in Canada and 75 in Australia. In other words, over a third of all programming languages were developed in a country with English as the primary language. This is apart from programming languages (such as Python from Netherlands, Lua from Brazil and Ruby from Japan) that were developed in a non-English-speaking country but used English, possibly to appeal to international audience.

Ezhil is an interpreted Tamil-based programming language developed by Annamalai (2013). Tamil is an Indian language spoken by over 60 million people (Subramania, 2015). The programming language is

targeted towards the K-12 (Junior high school) level Tamil speaking students as an early introduction to thinking like a computer scientists (Annamalai, 2013). The syntax of Ezhil is broadly similar to that of conventional BASIC.

Kanemune and Kuno (2005) described the development of a programming language called "Dolittle" whose keywords are based on Japanese language. Dolittle was developed in 2000 (Yoo et al, 2006) in response to the need for an object-oriented programming language that is suitable for children in both elementary and secondary schools in Japan. It was designed to be an object-oriented educational programming language. The programming language was written in Java and does not require elaborate declarations.

Qalb, developed by Nasser (2012), is a functional PL based on the Arabic language. The programming language enables one to write computer programs completely in the Arabic language. One of the motives for developing the programming language is to challenge the culture in which the design of most modern popular programming languages is predominantly based on the English language words. Such a culture, as noted by McAllister (2013) , makes learning programming especially difficult for students whose native language does not even use the Latin alphabets as does English language and for which English-based programming language keywords are little more than abstract symbols. Qalb is a programming language that deviates almost entirely from the use of ASCII character set for its encoding (Nasser, 2012).

Kumar (2011) described the development of the Hindawi programming system (HPS), a suite that allows users to program in Indic languages (Hindi, Bangla, Gujarati, Assamese and some other indic languages). The HPS, developed by Chaubary, A and Chaudbary, S., is a free, open-source, completely non-English-based programming platform that allows non-English medium literates of India to learn and write computer programs (Chaudbary, 2008). The HPS removes the English language barrier and enables non-English literates Indians to take up computer science and participates in the ICT at all levels of technology from primary school education to robotics and super-computing in their mother tongue. It is a scalable system which allows many programming paradigms.

### 4 PROPOSED RESEARCH APPROACH

In this research, development of a native language-based PL in which a program can be written in a PL that uses the lexicons of a native language , such as Yoruba, and then compiled and run is being proposed. The research derives its philosophy from theories of formal grammars, languages and automata as well as the orthography of Yoruba language. The architectural framework of the system is as shown in Figure 1. In figure 1, a programmer enters the Yoruba source statements through a notepad to be provided in the Yoruba integrated development environment (IDE) and stores it in a file. He then initiates the execution of the

Yoruba compiler to compile the source program file and stores the 'object code' in a file. At a later time, he can initiate the execution of the compiled program from the IDE.

A number of research methods will be employed in order to accomplish the goal of this research. Relevant information that would be gathered from literature survey will assist and guide in the design of a Yoruba-based programming language. In the design, regular expressions will be defined to specify the lexical structure of the programming language, while the syntactic structure of a valid program and program elements will be specified by designing a context-free grammar using Backus-Naur Form notations. A prototype implementation of a processor for the designed programming language will be carried out by developing a source-to-source compiler for the language. A qualitative evaluation of the resulting prototyped Yoruba-based programming language will be carried out using usability as the primary criterion. For this research, the Yoruba lexicons for implementation will be the ones without tone signs.

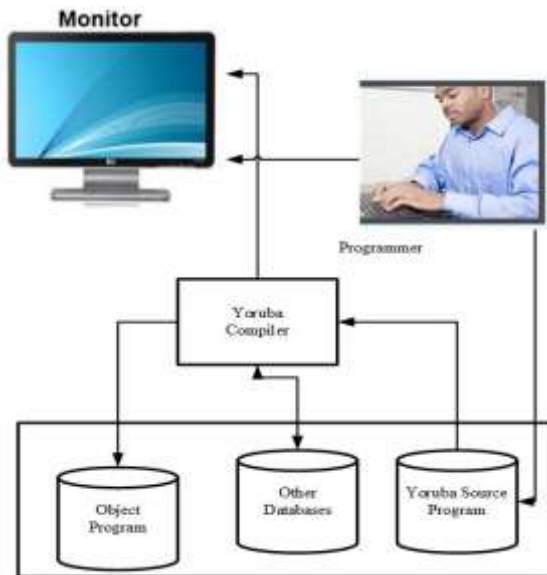


Fig. 1: Architectural Framework of the Yoruba Compiler System

### 5 PRELIMINARY RESULTS OF A TOKEN RECOGNIZER FOR THE PROPOSED PL

A mini token recognizer (scanner) has been developed in QBasic 64 within QB64 IDE for the proposed Yoruba programming language. The token recognizer was tested with a simple source program in Yoruba shown in Figure 2. Figure 3 is a sample output for source statement 4, while Figure 4 is the scanner output for the entire test data. Figure 3 contains the different tokens recognized in the source statement 4 along with their token categories. In the test data, exclamatory mark (!) precedes a comment statement, while a string literal is enclosed within two single quotes. Figure 4 shows that characters Q and Z in statement 4 of Figure 2 are correctly recognized as being invalid characters in the alphabets of Yoruba language.

```

Debo = Ade
Ade = 5
Abd = 4 (+
Jeki abd45yu = 56W / Q4 '- ) ihjkl + 'mnop' * ! rst < ul w
!Abd = 4 (E+ma) >Z - Ghogo / QZ
Abd = 4 (E+ma) >Z - Ghogo / QZ
    
```

Fig. 2: Source Program in Yoruba for Testing the Scanner

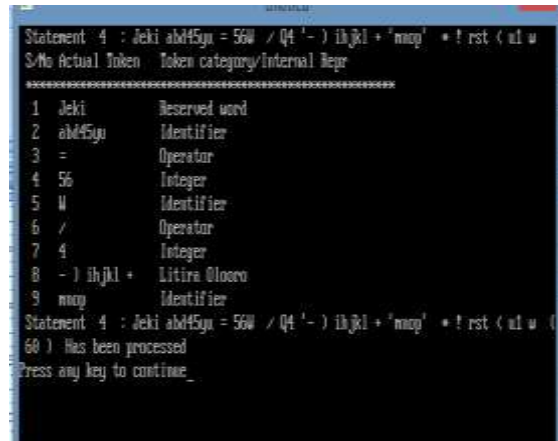


Fig. 3: Scanner's Output for Statement 4 in the Source Program

```

Scanner Produced Comments & Error File
=====
Aise ni ila 4 : Q je Karakita Ajeji
Aise ni ila 4 * ! rst < ul w Literal Olororo yi kodara to
0
Oro Alaye ni ila 5 : !Abd = 4 (E+ma) >Z - Ghogo / QZ
Aise ni ila 6 : Z je Karakita Ajeji
Aise ni ila 6 : Q je Karakita Ajeji
    
```

Fig. 4: Scanner's Output for the entire source program as will be seen by the Programmer

### 6 CONCLUSION AND FUTURE WORK

Development of an African native language-based PL has been proposed in this work. Successful accomplishment of the work will enable many people, in the order of millions, who are only literate in their mother tongue learn how to program the computer in their mother tongue. Furthermore, it will serve as an effective strategy to prevent predicted linguistic genocide of indigenous African languages (Benson, 2006, Azeez, 2013), In addition, it will ensure the continuity and relevance of indigenous languages in the age of ICT and increasing globalization as well as bridge the digital divide between the developed and under-developed/developing countries of the world (Wagner et al, 2010; Pflapsen 2011).

The next stage in this research is to embark on details of each of the intended methodology beginning with needs assessment survey for native-language-based programming languages.

## REFERENCES

- Aho, A. V Lam M.S; Sethi R. & Ullman J. D (2007). *Compilers – Principles, Techniques and Tools*, New York, Pearson Education/Addison Wesley, 2nded; pp 2, 12
- Alidou H., AliouBoly, B. B., Yaya S. D., Kathleen, H. , & EkkehardWolff, H. (2006). *Optimizing Learning and Education in Africa– the Language Factor*, Available online at [www.adeanet.org](http://www.adeanet.org), Retrieved on 30-07-2014.
- Annamalia, M. (2013). *Invitation to Ezhil: A Tamil Programming Language for Early Computer Science Rducation*; Retrieved from <http://ezhillang.org> on 17-03-2015
- Assembly of First Nations (2005). *Education through the Medium of the Mother-Tongue: The Single Most Important Means for Saving Indigenous Languages*, In *Symposium on Immersion Education for First Nations*, St. Thomas University Fredericton, N.B., Canada, October 3-6, 2005 Available online at [www.educatorsforimmersion.org](http://www.educatorsforimmersion.org), Retrieved on 08-08-2014
- Awobuluyi, O (2014). *Yoruba Must Not Die Out*, A Lecture delivered at the faculty of Arts, Obafemi Awolowo University, Ile-ife, Nigeria, Feb. 2014.
- Azeez, A.S (2013). *The Yoruba language and Literature in the 21stCentury and beyond*, *Covenant Journal of language Studies (CJLS)*, Vol 1, No 2, pp 148-159, Dec 2013
- Ball, J. (2010). *Enhancing learning of children from diverse language backgrounds: Mother tongue-based bilingual or multilingual education in the early years*, UNESCO. Available online at <http://unesdoc.unesco.org/images/0018/001869/186961e.pdf>, Retrieved on 30-07-2014
- Benson, C. (2006). *Mother Tongue First..* Available on line at [www.multicultures.com](http://www.multicultures.com), Retrieved on 30-07-2014
- Choudbary, S. (2008). *The Hindawi Programming System*, Retrieved from <http://hindawi.in/en/US/> on 17-03-2015
- Fabunmi, F. A. & Salawu, A.S (2005). *Is Yoruba an Endangered Language*, *Nordic Journal of African Studies*, Vol. 14(3):pp391-408, 2005. Available at [www.njas.helsinki.fi/.../fabunmi.pdf](http://www.njas.helsinki.fi/.../fabunmi.pdf), Retrieved on 28-01-2015.
- Gabrielli, M. & Martini, S. (2010). *Programming Languages: Principles and Paradigms, Undergraduate Topics in Computer Science*; London, Springer-Verlag; DOI 10.1007/978-1-84882-914-5-2. Retrieved on 09-07-2015 from [www.springer.com/cda/content/](http://www.springer.com/cda/content/)
- Kanemune, S & kuno, Y (2005). *Dolittle: an Objecct-oriented Language for K12 education*, *Warsaw, Eurologo 2005*, pp 144-153. Retrieved from [eurologo2005.oeiik.waw.pl](http://eurologo2005.oeiik.waw.pl) on 18-03-2015
- Kumar, L (2011). *Hindawi: Write Computer Programs in Indic Languages*, Retrieved from <http://techwelkin.com/hindawi-now-write-computer-programs-in-indic-languageson> 18-03-2015
- Mao, R (2015). *English Dominates Software*, *Berkeley Tech Review*, Available online at <http://berkeleytechreview.com/aplicaciones> Retrieved on 04-02-2016
- McAllister, N. (2013). *Meet Qalb, the Programming language that uses Arabic Script*. Retrieved on 18-03-2015 from [http://www.theregister.co.uk/2013/01/25/arabic\\_programming\\_language](http://www.theregister.co.uk/2013/01/25/arabic_programming_language)
- Nasser, R. (2012). *Qalb*, Retrieved from <http://nas.sr/%D9%82%D9%84%D8%A8> on 18-03-2015
- NBS(2010). *The National Literacy Survey. A Publication of National Bureau of Statistics, Abuja, Nigeria* Retrieved online on 25-06-2015 from [www.nigeriastat.gov.ng](http://www.nigeriastat.gov.ng)
- Olatunji, E. K (2006). *Compiler Techniques*, Ilorin – Nigeria, IgbagbogbemiPublisher 2ed, pp 6, 7,8, 63.
- Pflepsen, A. (2011). *Improving Learning Outcomes through Mother Tongue-Based Education*, , MTB-MLE Network. Available online at <https://www.eddataglobal.org> Last Accessed on 30-07-2014, stored online as [eddata\\_ii\\_mother\\_tongue\\_instruction.pdf](https://www.eddataglobal.org/eddata_ii_mother_tongue_instruction.pdf) Vol 1, No 2, pp 148-159, Dec 2013.
- Pratt, T.W. & Zelkowitz, M.V. (2001). *Programming Languages: Design and Implementation*, 4ed, Prentice Hall, ISBN-13-978-0130276780
- Pamela J. Mackenne & Jo Walker (2011). *Mother-Tongue Education: Policy lessons for Quality and Inclusion*, *Global Campaign for Education, Johannesburg, South Africa*. Available online at [www.campaignforeducation.org](http://www.campaignforeducation.org) Retrieved on 30-07-2014
- Sebesta, R. (2012). *Concept of Programming Languages*, Boston, USA, Pearson Education Inc, pp 190-192.
- Subramanian, K. (2015). *Bridging the Digital Divide with Tamil Coding*; Retrieved from [www.thehindu.com/~making-computing-in-tamil-easy/](http://www.thehindu.com/~making-computing-in-tamil-easy/) on 17-3-2015
- UNESCO (2007). *Mother Tongue-based Literacy Programmes: Case Studies of Good Practice in Asia*. Bangkok: UNESCO Bangkok, Thailand, ISBN 92-9223-113-8 Available online at [www.unesdoc.unesco.org](http://www.unesdoc.unesco.org) Retrieved on 30-07-2014
- Wagner, D. A., Daswani, C. J. & Romilla K. (2010). *Technology and Mother-Tongue Literacy in Southern India: Impact Studies among Young Children and Out-of-School Youth Creative Commons Attribution-Non Commercial-Share Alike 3.0*, Volume 6, Number 4, Winter 2010, 23–43, Available online at [www.itidjournal.org](http://www.itidjournal.org) Last accessed on 30-07-2014
- Yoo, S. W., Kim, K., Kim, Y., Yeun, Y., Kanemune, S & Lee, W (2006). *Empirical Study of Educational Programming languages for K-12: Between Dolittle and Visual Basic*; in *International Journal of Computer Sciences and Networks Security*, vol, No 6, 2006