

Implementation of CFG Based Natural Language Framework in Description of Syntax Errors in Imperative First Programming Languages: A Case Study from University of Baluchistan

Shafaque Saira Malik¹, Shumail Naveed² & Furqan-ul-haq Siddiqui³ and Mohammed Umer⁴

Department of Computer Science and Information Technology,

University of Baluchistan, Quetta, Pakistan

shafaque.malik@gmail.com, mshumailn@gmail.com

ABSTRACT— The purpose of this research is to study usefulness and impact of natural language framework for description of error messages on the performance, motivation, cognitive load of novices in imperative first programming language like C. This study is about investigating the impact of error messages description in natural language on debugging skill of students. It is used as a teaching tool in introductory programming language. In this research a framework based on natural language was constructed based on context free grammar (CFG) underpinning human computer interaction (HCI) to facilitate debugging of errors. This paper reports that using natural language to describe error messages decrease debugging time. We concluded that novices using this framework performed better with good understanding of static error handling, error correction with fewer number of errors. This framework has appreciable effect on learning outcomes of the students. The results of study reflect the time spent to debug and correct error was reduced as a result performance, motivation of the students increased and had impact on the cognitive load of students.

Keywords— FPL, NLF, Novices, CFG, Error Message (EM)

I. INTRODUCTION

Syntax is very difficult to learn for neophyte programmers. Debugging is complex skill for novices. Learning programming is very difficult for novices. The factors like syntax, error handling and correction of errors have noteworthy influence on the performance of novices in programming. The error messages (EM) indicate poorly cause of error as a consequence novice struggle very hard to retort to these errors messages as a result student do inappropriate proclamations and cause even more errors often. Discrepancy of errors and indicative messages generated by compiler is often hard nut to crack and is aggravated when same EM are generated for different errors and hence compiler generate perplexity and ambiguity accompanied by obscurity to eradicate error ensuing de-motivation, frustration and poor performance of novices. Cognition and identification of errors cannot be automated. Over the years many tools are developed to resolve the issue and to assist

debugging by incorporating “enhanced error messages” like Bluefix, BlueJ, Glid, HelpmeOut, pre-compiler tool called Espresso.

Context-free grammars are associated with linguistics where they are used to illustrate the structure of sentences, phrases and words in a language. In computer science they are used for describing programming

Programming language concepts and syntax, describe structure of the programming languages. A CFG is based upon a simple, mathematically precise mechanism for describing how phrases in language are built and are simple enough to represent construction of any parser mechanism. CFG play very important role for describing syntax of any programming language.

This paper represents research which reflects that NLF has significant influence on novice understanding of static errors and debugging by enhancing capability their

capability of understanding errors and reducing debugging time, hence ensuring self-directed static error resolution. CFG based NLF ensemble natural language description of error messages, underpinning HCI approach in IDE for resolution of errors. It is used as teaching tool for novices enrolled in CS1. It has significance influence on self-efficacy, performance, motivation and self-learning of novices thus resulting in less student retention / dropout rates in computer science majors. It also served as a tool to enhance interest, performance and programming skills of novice programmers.

II. LITERATURE REVIEW

Novices in their first introductory programming course face high mental effort and cognitive load. Learning programming languages is very difficult for most of the students of Baluchistan. It is usually cause of stress and trouble for many students enrolled in computer science and has “substantial effect on registration and retention for the programs” Dann, Cooper & Pausch (2006). Hooshyar, Alrashdan & Mikhak (2013) signposted that novices are very weak in problem solving and analysis and it is embellished due to intricate environments and syntax of the programming language.

If syntactical order of the parser is despoiled then syntax errors occurs. Error messages related to syntax are inscrutable and novices fail to comprehend them. User friendly EMs provide assistance to novices in learning programming language Schliep (2015). Correcting the syntax is the main phase to learn debugging therefore compiler error message usability is of prudish importance and it is used as a tool to discover and fix mistakes in the programs coded ; if errors messages are not helpful then learning programming becomes very hard, these error messages are basic source to understand what is wrong in the program. Schliep (2013). EM should not promote confusion Isa B.S. et al (1983). Marceau et al (2011) suggests that if the EMs are complicated and multifaceted then it will not be comprehended by novices and may often lead to erroneous path and thus engender frustration in novice programmers thus hampering their learning ability. They suggested that complex obscurities in EMs lead to new errors. Novices interrelate with compiler through EMs and they serve as vital mechanism for feedback

representing problems in program translation. EMs are description where translation broke down., novices are required to do alteration in their programs consequently, hence this incongruity is difficult for novices according to Munson & Schilling (2016). It is easier to debug errors if students have clear understanding of processing phases and outcome of problem statement noted by Iqbal & Coldwell (2017).

Ovsyannikov et al (2014) & Kuchuganov (2017) concluded that multifaceted development environment and hard syntax uplift poor problem investigation and resolving capabilities for novice as a result their motivation is reduced and they are badly overstressed to learn syntax .Denny et al (2014) noted that learning syntax is problematic for beginners .

The syntax and semantic of any programming language has substantial effect on the performance and motivation of pupils and as a result novices spent their most of the time contending with the grammar of the programming language and often flop to master in skills like problem solving Hooshyar, Alrashdan & Mikhak (2013). According to Marceau, Fisler & Krishnamurth (2011) the EMs received by novices do not directly indicate original error and they also noted that there are several issues related to effectiveness of Ems, for example error message does not reflect properly the actual error student have received. Traver, 2010 conducted case study on compiler errors and concluded that EMs are cryptic and awkward to comprehend by novices. Schliep (2015) noted that highlighting of mistakes / errors generate haziness and corresponds to ambiguous reference and recommended that EMs should be friendly and represented in simple vocabulary rather than compiler mumbo jumbo, complex terms and ambiguous sentences.

This paper represent research focusing on usefulness of natural language to explain errors, recommend simple easy to understand solutions thus warranting effectiveness of EMs serviceableness to ease debugging and ensuring elevated performance of novices. It is very maddening for the students who are new to programming to engrave code and learn syntax of today’s programming languages Daly.p. et al (2014). The semantic and syntax error are hectic for an

inexpert person. The errors are nastiest if they are semantic errors. Furthermore, it has been observed that syntax error reported by the compiler are at a location within the program, is many lines away from the source of the error, however semantic errors are difficult to track and are not highlighted by compiler as a result novice faces high altitude of discomfort and cognitive effort / cognitive load. This is frustrating for the students, as a result they may drop the program all together Porter & Calder (2004).

“Numerous exertions were depleted in order to make programming easier in introductory programming courses” noted by Anewalt (2008) & Daly (2011). Students have problems reading, writing, tracking, designing, debugging simple code segments MD Derus, Rosminah, & Ali (2012). EM are generated by compilers to help novices to locate and correct errors, however compiler messages are often scarce and ineducateand. Syntax error is a source of disillusionment and hurdle to students success Denny et al (2014).

Naveed, Sarim & Nadeem (2018) introduced the concept of learning mini language before learning introductory programming language with complex syntax and semantics and was called as LPL (Learners Programming Language) as a ZPL (Zeroth Programming Language), it aided to understanding syntax of FPL to deal with the lumpy syntax of the programming language in plain natural language and express syntax in the form of algorithmic way based on the computational statements.

Shafaque et al (2020) noted that static code analyser with enhanced error messages in simple plain natural language for description of error messages has considerable impact on novice error resolution skills.

CFG is based upon a simple, mathematically precise mechanism for describing how phrases in language are built and are simple enough to represent construction of any parser mechanism. CFG play very important role for describing syntax of any programming language. It is central feature associated with the word's organization, contents of phrases suggested by Rohrmeier et al (2016).

III. METHODOLOGY

The research framework in the figure.3.1 reflects the blueprint for error illustration and resolution. This NLF will ease novice programmers for generating and writing bug free source code by implementing approach of enhanced compiler error messages. Figure.3.1 represent conceptual frame work of this research reflecting that in order understand code there exists relationship between miscellaneous types of knowledge required by the beginner programmer like syntax, programming principles, programming concept and skills required by novice programmers.

The only way to communicate with the machine is through compilers, interpreters, assemblers. For novice programmer's complicated compiler error messages are challenging and are the only way to understand debugging Becker (2016). Commercial compilers generate unhelpful and sometimes misleading Ems reported by Juded (2005).

This conceptual model represents concept of alluring NLF to demonstrate and solve static syntax errors when writing source code, prior to compilation, underling concept of natural language programming assistance tools (PAT) to make program writing syntactically error free before compilation.

In figure.3.1 conceptual frame work is represented with the set of independent and dependent variable. It is demonstrated that novice enrolled in CS1 is treated as independent variable, rest of the framework is dependent upon this sole variable. Novices encounter errors, it is dependent variable, hence errors messages, number of errors encounter and static syntax error resolution significantly influence performance of novices therefore errors are considered independent variable. Performance of novices is dependent upon errors therefore it is dependent variable.

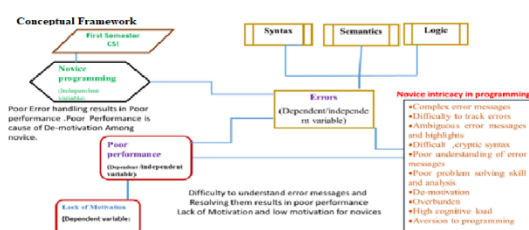


Figure 3.1. Conceptual Design

Performance directly influence motivation to learn programming. Motivation is dependent upon performance and is dependent variable. Novice programming is influenced by the errors encountered and resolved and these errors have impact on the performance of novices and performance.

The conceptual frame work of this research is to use natural language as core component for error description and solution in programming, which have correlation with performance and motivation of novice programmers. EM should be user friendly and described in familiar vocabulary or showing hints instead of compiler verbiages, complex terms and ambiguous sentences and is very important in designing EMs suggested by Schliep (2015). The syntax and semantic of any programming language have significant effect on the performance and motivation of students.

RQ1. Co-relation exists between natural language design, impetus and performance of novice programmers.

RQ2. Re-composition of compiler EMs by induction of natural language will have significant impact on the performance and motivation of novices in introductory programming courses.

RQ3. The performance, motivation and retention of novices are correlated with the induction of natural language in error resolution.

This paper reflects the research about outcomes of implementation of NLF based on CFG to describe static errors and their solution in FPL like C, novices are enrolled in this course majoring CS. This paper reports its impact on the performance, motivation of students in CS1. It has significant influence on their programming ability and final scores, diagnosis and solution of static errors, effect of EMs on novice response and rectification time to static errors, program writing time thus ensuring understanding of syntax errors in a better way than conventional tools used to write programs. The results are indicating optimization of diagnosis time for static syntax errors, and effect of enhanced EMs frame work in natural language for solution of static errors before compilation ensuring determination,

insistence and performance of novices, impacting their self-efficacy by means of ensuring low cognitive load and high interest in learning.

The participants were from CS1 who were enrolled in course of “introduction to programming” and C language as FPL, they were divided into two groups (Control group) and (Testing group) one group was instructed to use conventional programming tools and IDEs like classical Turbo C, Code Block, Dev; other group was initially instructed to use aforementioned tools and then were instructed to use NLF. Both groups had novices who attempted FPL, enrolled in CS and IT majors, at the university of Baluchistan during, 2018-19 academic year.

Aforementioned tools were used to write programs by the novices in their assignments and in class practice programs where as other group required to work independently alone on their assignments and in-class practice programs in NLF based on CFG. After few months’ groups were switched and performance was analysed in terms of error handling, writing, compiling, debugging time of programs ranging. Live data was collected as primary data and secondary data was their overall programming scores..

In the control group, grouping was done randomly with keeping in view factors like class performance. Some of the members were changed and were allowed to work in collaboration on the basis of demographical factors like living in dormitory / hostel with same background, language and remote areas. Data was combined and compared from each group.

Students in both groups were required to submit 15 home assignments and 10 to 15 in-class practice programming projects, both types of assignments were given scores for functionality, rate of error and their correction and error handling, readability and also time estimated time to write, execute and debug particular code, debugging time for static errors was also calculated for each programming task. Novices were required to submit their error logs in different programming assignments in both the groups along with the time scale.

Understanding of programming, static error description, error solving knowledge solo and in groups has

deep influence on performance, cognitive load and, motivation of novices.

Students working in conventional IDEs (control group) groups were disheartened, took more time to write code, debug static errors, rate of static typing errors was high, number of self-assumed errors was also high furthermore frequency of same error encountered was also high and most of the cases they fail to understand what actual error was as a result they were tired, uninterested, fed-up with less self-assurance, high cognitive stress, in the solutions they have developed. Most of the time they were observed inundated with error correction and on average spent more time to correct static syntax errors. It has been observed that conventional IDEs group (control group) members often complained that they can't learn what was taught in class during their projects. The NLF (test group) however performed better than the other group due to proper highlighting, simple description of errors and correction suggestion. All groups were given 15 programs to write, afterwards that 6 programs were given to each group with errors in order to analyse how much time they take to debug programs. As expected conventional control group debugging time was longer than NLF test group.

Data was collected regarding novices scores in FPL. Programming time, debugging time was also collected in order to analyse their performance and motivation in programming, which were later compared for each group.

The programming assistance tool (PAT) we developed using CFG framework generates error messages in natural language and on spot correction tips before compilation for static error, CFG for this framework is adopted from W. Kernighan & Dennis M. Ritchie (1988).

Natural Language Framework for Description of Static Errors in Programming Language



Figure.3.2 Illustration of error and solution in NLF based on CFG

Figure.3.2 Demonstrating enhanced error message in natural language along with the solution tip to correct and resolve the static syntax error, which is in simple and easy to understand format for novices and as a result they have to spent least time on understanding and resolving errors.

IV. DATA ANALYSIS & RESULTS

Course outline of FPL was same in all the groups. Hypothesis of this study was set keeping in sight hypothetical research perspective of novices to succeed. Final scores for all the groups were collected along with the time consumed to write program in class and at home was also analysed for each of 15 programs, debugging time in different tools and NLF based on CFG were compared to determine if our hypothesis holds that performance and motivation of novices in FPL is influenced by static errors, error diagnostic time description of errors, and cognitive load of novices which is considerably effected by EMs generated whenever static syntax error encountered.

Performance in Conventional IDES Verses NLF For Elucidation of Errors / Debugging Time

We collected data from novices who were enrolled in CS1 in FPL. The novices used instructional programming environment called “NLF for description of static errors in programming language” and has been designed based on CFG. Novices were given 45 programs from easy to complex. The debugging time required to fix the errors was better in NLF. The average time required to debug single program was calculated and compared with the debugging time in Conventional IDEs. The results are illustrated in the following figures. 4.1, 4.2, 4.3.

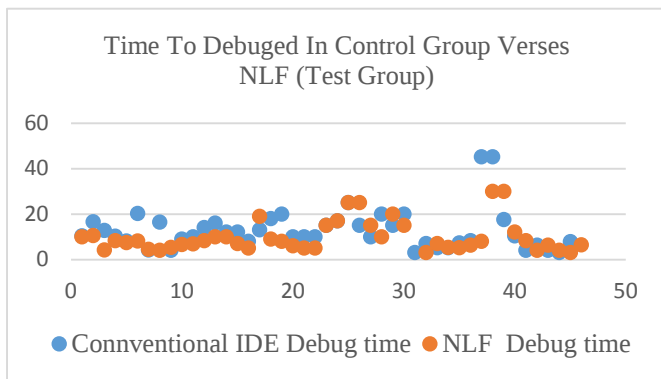


Figure.4.1 Debugging time comparison Control VS Treatment Groups

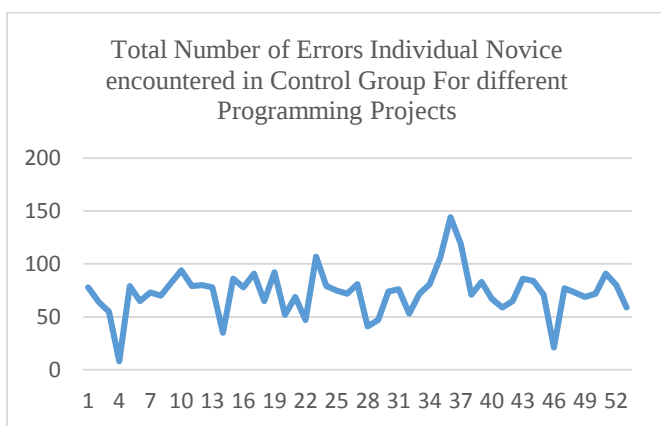


Figure.4.2 Total Number of Errors in Programming in Control Group using conventional IDEs

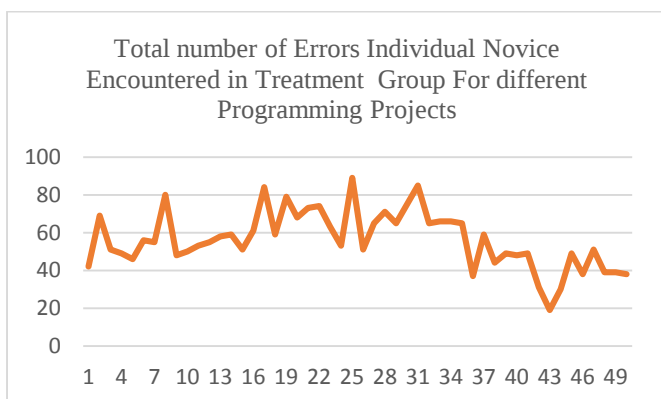


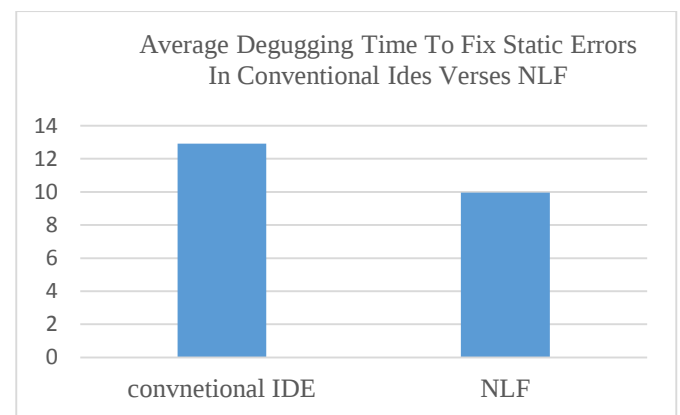
Figure.4.3 Total Number of Errors in Programming in Test Group using NLF

It is inferred from the results that hypothesis RQ2 holds that is, decomposition of compiler EMs in natural language has strong connotation with the performance of novices. It is inferred from results that students in NLF group performed better as compare to male /female counterparts in CS1 in control group. The time required for each program is on average greater in Conventional IDEs control group, even for

simple programs like pyramid of stars as compared to NLF test group. Novices in NLF test groups performed better in solo in class for each of given assignments with less number of errors and time then novices in the conventional IDEs control group. However when they were asked to switch to NLF their performance enhanced both in grades and in debugging time as a result their overall debugging time improved in NLF due to easy description of error messages and solution suggestions. They have to make less efforts to correct static syntax errors than conventional environments used to write programs in FPL. Performance and motivation was better necessitating deep learning then surface learning, with high self-efficacy, better understanding of errors and much improved degree of persistence in FPL.

V. DISCUSSION

Novices were given 45 programs. Programs were to be written, debugged in conventional IDEs like Code Block, Dev, Turbo C. Later on they were instructed to write same programs in NLF, average time required to debug each program was less in NLF. Furthermore overall average debugging time was also less in NLF as compared to conventional tools, results are illustrated in figure.5.1.



The figure.5.1 Comparison of Average Debugging Time in Convention IDEs and NLF for Programming assignments

Quality of programs produced by NLF test group was better, with fewer errors and more readability. The T-test conducted on the performance of subjects shows that there was a significant difference on the score of conventional IDEs control groups and NLF test group such that **t-value = 1.67** and **p < .05**. It is concluded that over all NLF is useful. It is easy for majority of novices in FPL. Through this research it is certain that natural language based framework for

handling errors in programming is especially beneficial for novices in their first programming language course. NLF broadly addressed many considerable aspects which ensured participation and progress of novices in computer science and programming. It is inferred that debugging becomes easy if the description of error messages is in simple, easy to understand natural language then complex terms of compiler hence facilitates learning of programming language with fortitude and guaranteeing creativity, steadfastness and effective software development in computer science majors and inspires novices to trail their potential programming careers in CS.

VI. CONCLUSION

The results of this study specify that NLF based on CFG for description of errors in programming language enhance learning which in turn improve performance as a result motivation of novices is increased, and such progress is of prudish importance in order to enhance performance and inspiration of novices reliably on fixing static errors. We assume that NLF will benefit students to develop potential skills to overcome hurdles in their first programming language course and develop improved debugging skills

The results of this research indicate that understanding of error messages (EMs) to resolve static errors plays a very important role in engaging, approving, nourishing interest, passion of novices whether male or female in learning programming language and hence ensuring real progress of programming and promising future in computer science.

The results of this study provide real indication that debugging time in programs effects performance of novices in their first programming course effectively and later on it will play remarkable role in later programming oriented subjects and has deep impact on novice performance and is one of most effective pedagogical tool that has deep impact and effect on the learning capability and performance of novices in CS1. The results portray that CFG based NLF guarantees completion of the course with high retention, less drop-out and high pass rate. It plays a very crucial role in perseverance, determination to stay in CS majors. Novices

who were using NLF showed high efficiency in programming with high degree of error detection, quick fix solutions of errors, good problem solving and produces better quality programs than others. NLF is effective tool in learning programming for novices and increase their skill in programming although and has profound outcome on the performance of both novices in FPL. NLF has significant impact on error resolution, diagnosis of errors, understanding of errors. Identifying classes of errors in programming and effective debugging skills of novices is left for future work.

Limitation of study

The limitation of this research is that it is confined to only introductory first programming languages. This study was confined to imperative first programming languages like C and not object-oriented languages. Demographical factors in evaluation was not considered in this research and it was confined to static syntax errors.

VII. ACKNOWLEDGMENT

The authors are obliged to the Department of Computer science for the benefaction and facility. The authors would also like to thank all the pupils who contributed in the study. We are grateful to Mr. Mohammed Umer, Mr. Saeed Kaker, Mr. Yashran, Mrs. Mumtaz Idress for their support.

REFERENCES

- [1] Anewalt, K. (2008). Making CS0 fun: an active learning approach using toys, games and Alice. *Journal of Computing Sciences in Colleges*, 23(3), 98-105.
- [2] Becker, B. A., Glanville, G., Iwashima, R., McDonnell, C., Goslin, K., & Mooney, C. (2016). Effective compiler error message enhancement for novice programming students. *Computer Science Education*, 26(2-3), 148-175.
- [3] Brian W. Kernighan and Dennis M. Ritchie (1988). *The C programming language*, 2nd edition, Section A13. Prentice Hall.
- [4] Danial Hooshyar, Moslem Yousefi and Heuseok Lim, A systematic review of data-driven approaches in player modeling of educational games, *Artificial Intelligence Review*, (2017)

- [5] Dann, W., Copper, S., & Pausch, R. (2006). Learning to program with Alice. Upper Saddle River, NJ: Prentice Hall.
- [6] Daly, T. (2011, May). Minimizing to maximize: An initial attempt at teaching introductory programming using Alice. *Journal of Computer Science in Colleges*, 26(5), 23-3
- [7] Denny et al. (2014). Code write :dupporting students driven practices of java. *ACM* , 471-476.
- [8] Denny, P., Luxton-Reilly, A., & Carpenter. . (2014). Enhancing syntax error messages appears ineffectual. *ACM* , 273-278.
- [9] Derus, S. R. M., & Ali, A. Z. M. (2012). Difficulties in learning programming: Views of students. In 1st International Conference on Current Issues in Education (ICCIE 2012) (pp. 74-79).
- [10] Hooshyar, D., Maïen, T., & Masih, M. (2013). Flowchart-based programming environments aimed at novices. *International Journal of Innovative Ideas*, 13(1), 52-62.
- [11] Hooshyar, D., Ahmad, R. B., Yousefi, M., Yusop, F. D., & Horng, S. J. (2015). A flowchart- based intelligent tutoring system for improving problem- solving skills of novice programmers. *Journal of Computer Assisted Learning*, 31(4), 345-36.
- [12] Iqbal Malik, S., & Coldwell-Neilson, J. (2017). Impact of a new teaching and learning approach in an introductory programming course. *Journal of Educational Computing Research*, 55(6), 789-819
- [13] Isong, B. (2014). A Methodology for Teaching Computer Programming: first year students' perspective. *International Journal of Modern Education and Computer Science*, 6(9), 15.
- [14] Isa, B. S., Boyle, J. M., Neal, A. S., & Simons, R. M. (1983, December). A methodology for objectively evaluating error messages. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (pp. 68-71). ACM.
- [15] Jadud, M.C. A First Look at Novice Compilation Behaviour Using BlueJ. *Computer Science Education*, 15(1):25--40, 2005.
- [16] Kuchuganov, V. N., Kasimov, D. R., & Kuchuganov, M. V. (2017, September). Principles of organizing the semantic ontology of programming. In *Computer Technology and Applications (RPC), 2017 Second Russia and Pacific Conference on* (pp. 122-126). IE
- [17] Munson, J. P., & Zitovsky, J. P. (2018, February). Models for Early Identification of Struggling Novice Programmers. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 699-704). ACM.
- [18] Kummerfeld, S. K., and Kay, J. The neglected battle fields of syntax errors. In *Proceedings of the Fifth Australasian Conference on Computing Education -Volume 20* (Darlinghurst, Australia, Australia, 2003), ACE '03, Australian Computer Society, Inc., pp. 105–111
- [19] Koorse, M., Cilliers, C., & Calitz, A. (2015). Programming assistance tools to support the learning of IT programming in South African secondary schools. *Computers & Education*, 82, 162-178
- [20] M.C. Jadud, "A First Look at Novice Compilation Behaviour Using BlueJ," *Computer Science Education*, vol. 15, Mar. 2005, p. 25--40.
- [21] Malik, S. I., & Coldwell-Neilson, J. (2017). A model for teaching an introductory programming course using ADRI. *Education and Information Technologies*, 22(3), 1089-1120.
- [22] Marceau, G., Fisler, K., & Krishnamurthi, S. (2011, October). Mind your language: on novices' interactions with error messages. In *Proceedings of the 10th SIGPLAN symposium on New ideas, new paradigms, and reflections on programming and software* (pp. 3-18). ACM.
- [23] Mason, R., & Cooper, G. (2014, January). Introductory Programming Courses in Australia and New Zealand in 2013-trends and reasons. In *Proceedings of the Sixteenth Australasian Computing Education Conference-Volume 148* (pp. 139-147). Australian Computer Society, Inc.
- [24] Munson, J. P., & Schilling, E. A. (2016). Analyzing novice programmers' response to compiler error messages. *Journal of Computing Sciences in Colleges*, 31(3), 53-61.
- [25] Naveed, S., Sarim, M., & Nadeem, A. (2018). C in CS1: Snags and viable solution. *Mehran University Research Journal of Engineering & Technology*, 37(1), 1.

- [26] Ovsyannikov, M. K., & Kasimov, D. R. (2014). Editor and Interpreter of Program Flowcharts for Distance Learning Programming. Bulletin of Kalashnikov ISTU, (3), 154-156.
- [27] Porter, R., & Calder, P. (2004). Patterns in learning to program: an experiment? Proceedings of the Sixth Conference on Australasian Computing Education – Volume 30, 241 -246. Retrieved April 18, 2008 from ACM.
- [28] Rohrmeier, M., Fu, Q., & Dienes, Z. (2012). Implicit learning of recursive context-free grammars. PloS one, 7(10), e45885.
- [29] Schliep, P. A. (2015). Usability of Error Messages for Introductory Students. Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal, 2(2), 5.
- [30] Shafaque, Naveed & Furqan (2020). Creation of CFG based natural language framework for explication of syntax errors in first programming language featuring novices. LC International Journal of STEM, 1(2).
- [31] V.J. Traver, "On compiler error messages: what they say and what they mean," Technical Report, Computer Languages and Systems Department, Jaume-I University, 2010.

In 2007, joined Institute of Management Science, University of Baluchistan as Lecturer. He is now serving as Assistant Professor.

BIOGRAPHY



Shafaque Saira Malik was born in Quetta, Pakistan. She received the BCS degree in computer science from the Allama Iqbal Open University, ISB, Pakistan and MCS from UOB, Quetta, Pakistan. ETE From AIT, Thailand and STKLC from Ehwa University, Seoul, South Korea.

In 2007, she joined the Department of Computer Science and Information Technology, University of Baluchistan as Lecturer. Her current research interests include programming, agile programming, pair programming, computer science education. She is Coordinator for University of Baluchistan, and is a Fellow of National Academy of Young Scientists Pakistan; She has received Technology award in Invention for Innovation Summit (2019).



Furqan-ul-haq Siddiui was born in Mastung, Pakistan. He did graduation in Commerce from UOB, Quetta, Pakistan and MBA from UOB, Quetta, Pakistan. He did MS from Iqra University, Karachi, Pakistan,

