Tattva - A JOURNEY INTO REGIONAL LANGUAGE DEVELOPMENT

[1]Dr. Rajashekhar Ainapur, [2]Ayesha Siddiqua, [3]Bhaskar N Prasad, [4]Chethan L, [5]Deepak Ramanujam V.R
[1]Professor, [2,3,4,5]Students
Department of CSE
East West Institute of Technology
Bengaluru, India

Abstract— The Tattva Project stands as an innovative initiative with the goal of empowering students through the creation of a user-friendly system based on Python, specifically designed for Kannada speakers. This endeavor aims to eliminate language barriers in programming education, ensuring equal access to learning opportunities. The project involves the development of an intuitive system accompanied by a comprehensive set of educational resources, including tutorials and documentation in Kannada, along with transliteration. The overarching objective is to enable students to write and execute programs in Kannada using English alphabets for transliteration. As a future enhancement, the project envisions the creation of a compiler, a crucial component that translates human-readable code into machine- executable instructions. Essentially, the Tattva Project marks a significant stride toward democratizing programming education, providing a crucial tool for skill acquisition and empowerment across diverse communities.

Keywords— Compiler Development, Democratizing Education, Educational Resources, English Alphabets, Equitable Access, Kannada Language, Programming Education, Transliteration.

I. INTRODUCTION

In the dynamic landscape of programming education, ensuring inclusivity and breaking down linguistic barriers is pivotal for equitable knowledge dissemination. The Tatva project, a pioneering initiative in this domain, focuses on developing Kannada-Script Programming Language that utilizes English alphabets for coding. This survey paper explores the theoretical foundations, design considerations, and the potential impact of Tatva in fostering inclusivity within programming education, particularly for Kannada speakers.

Tatva embodies a vision where learners fluent in Kannada can embark on their coding journey seamlessly by utilizing familiar English characters. The project's objective is to transcend language barriers, making programming education accessible and inclusive for a diverse audience. Unlike an implemented system, this survey delves into the conceptual framework and theoretical underpinnings of the Tatva project.

The core philosophy of Tatva lies in creating a user-friendly programming language that integrates Kannada syntax with English alphabets. Concepts such as using "iddare" for "if" exemplify the innovative approach of combining Kannada semantics with the ease of English letters. The survey navigates through the design principles, the potential educational implications, and the anticipated impact of Tatva on programming education.

This paper does not present an implemented version but provides an insightful survey of the conceptualization and theoretical considerations that form the foundation of the Tatva project. As we explore the theoretical landscape, we anticipate that Tatva will be a significant stride towards inclusive programming education, creating a bridge for learners to acquire essential digital skills without linguistic hindrances.

II. LITERATURE SURVEY

The literature review encompasses a diverse range of studies aimed at enhancing computer programming education, particularly focusing on effective approaches in different contexts. The review begins with a comprehensive investigation into the challenges and proposed models for teaching programming in developing countries, emphasizingresource constraints [1].

It then explores the effect of specialized programming languages, such as Calango, on introductory courses, highlighting their potential to improve interaction and simplify syntax [2]. Additionally, the review delves into the crucial aspects of teaching methods with programming language choices in software engineering education, offeringpractical insights for educators [3].

A critical examination of failure rates in an introductory programming courses underscores the necessity for adaptive teaching methods and course design [4]. The review further explores the dynamics of active learning environments, emphasizing increased student motivation but acknowledging potential limitations [5].

A comparative study between e-learning and m-learning in introductory programming courses provides valuable insights [6], while an innovative LPL(Learners Programming Language) is identified for its effectiveness in simplifying programming concepts[7]. The review concludes by considering the challenges and benefits of language choices in programming education, examining the impact by multilingualism and the importance of any cultural context in

language learning [8]. Overall, the literature survey offers a nuanced understanding of the complexities involved in programming education, catering to diverse learner needs and contexts.

Continuing the literature review, the exploration extends to studies addressing the nuances of teaching programming languages. A study on the assessment of programming language learning introduces a rubric-driven online peer code-review system, emphasizing its effectiveness in a pre-introductory computer science course [13]. However, limitations are acknowledged concerning the context specificity of the findings. The relationship between language and culture in English teaching is examined, shedding light on the importance considering cultural context in language learning[14], although specific advantages and disadvantages are not explicitly outlined.

Shifting focus to the design and deployment of programming courses, a study investigates strategies, tools, difficulties, and pedagogy [11]. While offering valuable insights, it does not provide a systematic review of the literature on designing and deploying programming courses, and potential gaps in coverage are acknowledged.

The literature survey also explores the role of programming assistance tools (PATs) in supporting IT scholars' programming knowledge and skills in South African secondary schools [12]. Though highlighting the potential benefits, the study falls short of offering a comprehensive review of PATs in this context, leaving roomfor further exploration.

Lastly, a study emphasizing the benefits of multilingualism in education provides a broader perspective on language in education, focusing on cognitive development, academic achievement, and cultural awareness [10]. However, potential disadvantages, such as language dominance and increased cost, are recognized.

In summary, the literature review encompasses a wide array of studies that collectively contribute to a comprehensive understanding of the challenges, opportunities, and innovative approaches within the realm of programming education. Each study brings a unique perspective, collectively building a foundation for the development and enhancement of effective programming education methodologies

Authors	Year	Advantages	Disadvantages
Idongesit Eteng, Sylvia Akpotuzor, Solomon O. Akinola, and Iwinosa Agbonlahor.	2022	1. The review addresses the challenges faced by universities in developing countries and possibly some developed countries in offering computer programming courses due to a lack of technological infrastructure and peculiar limitations.	1. The study's findings and proposed model may be based on a limited number of sources, which could affect the generalizability of the conclusions. 2. The article may not provide a comprehensive review of the literature on teaching computer programming in developing countries
Geovana Silva, et.al	2020	It offers a straightforward syntax that facilitates the development of advanced logic, allowing students to focus on logic without being burdened by language intricacies.	Calango may not be well-suited for advanced programming courses, as its design is geared towards facilitating algorithm learning and preparing learners before the introduction of the C language.
L. Gu, N. Yan, and Y. Xiu	2020	1. Relevance: The paper is likely to be advantageous for educators, curriculum designers, and researchers in any field of software engineering education due to its focus on teaching methodology and programming language selection. 2. Practical Insights: Offers practical insights into the decisions and challenges associated with selecting appropriate programming languages for software engineering majors.	1 Specific Context: If paper is specific to a certain region, institution, or context, its applicability may be limited to that particular setting, potentially reducing its generalizability. 2. Limited Methodological Information: If the paper lacks detailed information on the research methodology employed, it may be challenging for readers to assess robustness of the findings.
C. Watson and F. W. Li	2019	Identifies that students who are struggling with the course tend to have difficulty with fundamental programming concepts.	While the paper is enhanced with valuable insights into the high failure rates of introductory programming courses, the study's sample size is limited to the conference attendees, and the results may not be representative of the broader population of introductory programming students.
L. Deslauriers, et.al	2019	Elevated motivation and engagement can result from learning in context, allowing students to recognize the significance of disciplinary knowledge and skills.	The study only examines introductory programming courses and does not provide insights into the success rates of more advanced programming courses.
Sohail Iqbal Malik, Roshan Mathew, and Raed Al-Nuaimi	2019	1. The article compares e-learning and m-learning in an intro programming course, providing insights for educators and institutions. 2.The study introduces PROBSOL to enhance novice programmers' problem-solving.	1. The article's findings and implementation of the PROBSOL application may be based on a specific educational context. 2. The study's focus on a single application may not fully capture the potential challenges and benefits of implementing e-learning and m-learning in different educational environments.

III. CONCLUSION

This survey delves into the conceptual foundations of the Tatva project, an innovative initiative dedicated to creating a Kannada-Script Programming Language using English alphabets to enhance inclusive programming education. Although not an assessment of the implemented system, the survey offers a thorough exploration of the theoretical underpinnings, design considerations, and potential impact of Tatva in promoting inclusivity within the programming education realm.

As we look ahead, the insights gained from this survey pave the way for future research, implementation, and refinement of the Tatva project. The theoretical framework explored here provides a solid foundation for educators, researchers, and policymakers to understand the potential implications and contributions of Tatva to the field of programming education.

In summary, Tatva stands at the forefront of a movement towards inclusive programming education, where the native language becomes a facilitator rather than a hindrance. This survey serves as a valuable resource for those interested in the theoretical nuances of Tatva, laying the groundwork for further exploration and eventual implementation, with the ultimate goal of creating a more inclusive and accessible programming education landscape.

REFERENCES

[1] Eteng, I., Akpotuzor, S., Akinola, S. O., Agbonlahor, I., "Effective Teaching of Programming in Developing Countries," Computer Science Department, University of Calabar, Nigeria; Computer Science Department, Arthur Jarvis University, Nigeria; Computer Science Department, University of Ibadan, Nigeria. Received 19 Nov 2021, Revised 30 Mar 2022, Accepted 3 Jun 2022, Published online 6 Jun 2022, Version of Record 13 Jun 2022.

[2] Silva, G.R.S., Santos, G., Canedo, E.D., Rissoli, V., "Impact of Calango Language in an Introductory Computer Programming Course," University of Brasília. October 2020. DOI: 10.1109/FIE44824.2020.9274150. Conference: 2020 IEEE Frontiers in Education Conference (FIE).

[3] Gu, L., Yan, N., Xiu, Y., "Discussion on Teaching Methods and Choice of Programming Language on Software Engineering Major," DEStech Transactions on Engineering and Technology Research.
June 2017. DOI: 10.12783/dtetr/apetc2017/11364

[4] Watson, C., Li, F.W.B., "Failure Rates in Introductory Programming Revisited," ITiCSE '14, 2014. DOI: 10.1145/2591708.2591749. This paper systematically reviews and analyzes pass rate data from 161 CS1 courses across 15 countries, revealing a consistent mean worldwide pass rate of 67.7%. Moderator analysis identifies notable differences based on grade level, country, and class size, providing quantitative evidence on challenges in introductory programmingeducation..

[5] Deslauriers, L., McCarty, L.S., Miller, K., Kestin, G., "Measuring Actual Learning versus Feeling of Learning in Response to Being Actively Engaged in the Classroom," Louis Deslauriers: ORCID 0000-0002-4800-5770, louis@physics.harvard.edu; Logan S. McCarty: ORCID 0000-0002-7981-9267. Edited by Kenneth W. Wachter, University of California, Berkeley, CA. Approved August 13, 2019 (received for review December 24, 2018)..

[6] Malik, S.I., Mathew, R., Al-Nuaimi, R., Al Sideiri, A., "Learning Problem Solving Skills: Comparison of E-learning and M-learning in an Introductory Programming Course," Education and Information Technologies, 24(4), September 2019. DOI: 10.1007/s10639-019- 09896-1. Authors: Sohail Iqbal Malik, Al Buraimi University College; Roy Mathew, Al Buraimi University College; Rim Al- Nuaimi, Al Buraimi University College; Abir Al Sideiri, Al Buraimi University College..

[7] Naveed, M., Sarim, M., Ahsan, K., "Learners Programming Language: A Helping System for Introductory Programming Courses," Mehran Univ. Res. J. Eng. Technol., 35(3), July 2016. DOI: 10.22581/muet1982.1603.05. Authors: Muhammad Naveed, Muhammad Sarim, Kamran Ahsan, Federal Urdu University of Arts, Science and Technology..

[8] Guo, P.J., "Non-Native English Speakers Learning Computer Programming: Barriers, Desires, and Design Opportunities," CHI '18: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, April 2018. Paper No.: 396. Pages 1–14. DOI: 10.1145/3173574.3173970\.

[9] Yoon, I., Kim, J., Lee, W., "The Analysis and Application of an Educational Programming Language (RUR-PLE) for a Pre- introductory Computer Science Course," Cluster Computing, 19(1), March 2016. DOI: 10.1007/s10586-016-0540-6. Authors: IlKyu Yoon, Jamee Kim, Korea University; Wongyu Lee, Yonsei University.

[10] Okal, B.O., "Benefits of Multilingualism in Education," Universal Journal of Educational Research, 2(3), 223-229, 2014. DOI: 10.13189/ujer.2014.020304. Corresponding Author: Benard Odoyo Okal, School of Arts and Social Sciences, Department of Kiswahili and other African Languages, Maseno University, Kenya.

[11] "Designing and Deploying Programming Courses: Strategies, Tools, Difficulties and Pedagogy," Education and Information Technologies, Volume 21, Pages 559–588,

Published: 15 July 2014.

[12] Koorsse, M., Cilliers, C., Calitz, A.P., "Programming Assistance Tools to Support the Learning of IT Programming in South African Secondary Schools," Computers & Education, 82, 162-178, March 2015. DOI: 10.1016/j.compedu.2014.11.020. Authors: Melisa Koorsse, Charmain Cilliers, André P. Calitz, Nelson Mandela University

[13] Wang, Y., Li, H., Feng, Y.Q., Jiang, Y., "Assessment of Programming Language Learning Based on Peer Code Review Model: Implementation and Experience Report," Computers & Education, 59(2), 412-422, July 2012. DOI: 10.1016/j.compedu.2012.01.007. Authors: Yanqing Wang, Hang Li, Y.Q. Feng, Yu Jiang, Harbin Institute of Technology.

[14] Sha Pan and Zhong Cao, "Relation between language and culture in English teaching," 2010 2nd International Conference on Education Technology and Computer, Shanghai, China, 2010, pp. V1-186-V1-188, doi: 10.1109/ICETC.2010.5529272.