A Proposal of Adaptive Learning System for Object-Oriented Programming Education

Angela Vitoria Mota Vieira
Institute of Computing (IComp)
Federal University of Amazonas (UFAM)
Manaus, Brazil
avmv@icomp.ufam.edu.br

Ramayana Assunção Menezes Jr
Institute of Computing (IComp)
Federal University of Amazonas (UFAM)
Manaus, Brazil
ramayana@icomp.ufam.edu.br

Jose Francisco Magalhães Netto
Institute of Computing (IComp)
Federal University of Amazonas (UFAM)
Manaus, Brazil
jnetto@icomp.ufam.edu

Abstract - This paper presents an Adaptive Learning (AL) system for teaching Object-Oriented Programming (OOP). The OOP paradigm is one of the foundations for introductory programming education, being one of the ACM (Association for Computing Machinery) Computing Curricula's Programming Languages Fundamentals. However, studies show that students have difficulty associating abstract concepts with real-world analogies. When trying to solve a problem, the difficulty in understanding and applying concepts can hinder using the paradigm. Therefore, it is necessary to create alternatives to assist in teaching Object-Oriented Programming. One solution is to use Adaptive Learning, which aims to adjust the knowledge acquisition process according to the student. Learning Objects (LO), which can be games, quizzes, interactive experiences, videos, and images, among others, can be used to support this method. Thus, this work aims to present a proposal for an approach to teaching OOP using Adaptive Learning. The approach consists of a support system for OOP students, which will support their teaching with a Learning Object recommendation mechanism that is more relevant to the student's learning style. To this end, agents will be used to control the selection of LOs and adjust the profiles. The student's actions will be collected and used as a basis for the recommendation. From the data obtained, constant adjustments will be made to the recommendation of LOs and the student's profile, thus providing more suitable content for their learning. As a result of this study, a functional prototype of the recommendation method was developed in a mobile application. The use of this application will be evaluated quantitatively and qualitatively to validate the solution. After using the application, the evaluation results will provide us with information about the users' perception of the proposal, its recommendation accuracy, and usability. With this, we can build a version that will be used in the classroom and contribute to the student's OOP learning process.

Keywords - Adaptation Learning, Object-Oriented Programming, Prototype, Survey, SUS.

I. INTRODUCTION

Object-Oriented Programming is a fundamental paradigm in computer science education, emphasized in the ACM Computing Curricula's Programming Languages Fundamentals. Despite its importance, students often need help to grasp abstract OOP concepts and apply them to real-world scenarios. This difficulty can hinder their ability to utilize the paradigm in problem-solving effectively. Adaptive Learning systems have emerged as a promising solution to address this challenge. AL tailors the learning experience to individual student needs, utilizing various Learning Objects such as games, quizzes, interactive experiences, videos, and images.

This paper proposes an innovative approach to teaching OOP using Adaptive Learning. We present a support system designed to enhance OOP education for students. The system incorporates a recommendation mechanism that suggests LOs based on the student's Learning Style, aiming to improve their performance in programming courses. The system continuously adapts and refines its recommendations by collecting data on student interactions and preferences, ensuring that the content aligns with the student's unique learning style. We have developed a functional prototype of this recommendation method in the form of a mobile application. The prototype will undergo rigorous quantitative and qualitative evaluation to validate its effectiveness. The evaluation results will provide valuable insights into user perceptions, recommendation accuracy, and overall usability. This feedback will guide the development of a refined version for classroom use, ultimately contributing to a more effective and personalized OOP learning experience for students.

II. THEORIC FOUNDATION

Some concepts are essential to understanding our application's goals.

A. Adaptive Learning

According to [1], "Adaptive learning systems consist of multiple components, which together allow instructions to be tailored to the needs of each student." Various authors have conducted studies on adaptive learning using algorithms, human-computer interaction techniques, and other methods to analyze students' learning processes.

Reference [2] highlights the Brazilian educational platform GEEKIE GAMES, created in 2011. It provides a platform for preparation for the ENEM (National High School Examination). With adaptive learning, GEEKIE GAMES offers users a differentiated approach: "with an indication of educational content, video lessons, and, through an algorithm, identifies the main difficulties of the student in each subject" [2].

The systematic review conducted in [3] highlighted studies with an interest in the correlation between adaptive learning and machine learning techniques, such as heuristic methods, bayesian methods, and artificial neural networks, among others.

Reference [2] emphasizes in their study that using platforms that employ adaptive learning provides quality learning materials and allows students to have a more effective knowledge acquisition process at their own pace, with their personal goals, and with the application of acquired learning.

B. Learning Style

Reference [4] introduced the Theory of Meaningful Learning, which underscores the influence of a student's prior knowledge on the learning process. In his research, he delineates two learning approaches: meaningful and rote. Meaningful learning entails translating the logical significance of pedagogical material into psychological meaning, whereas rote learning involves the mechanical absorption of new information.

Researchers have devised learning styles inventories (LSIs) to classify diverse learning styles [5]. Among the prominent models utilized in studies are [6], [7].

As in [7], learning styles will be employed in delineating learning profiles. Reference [8] underscores that this model ranks among the most widely used in suggesting Learning Objects within the Adaptive Learning paradigm.

The Felder-Silverman's Learning Styles Model (FSLSM) was introduced by [7]. This probabilistic model encompasses four dimensions:

1. Perception (Sensing/Intuitive);

2. Information Input (Visual/Verbal);

3. Information Processing (Active/Reflective);

4. Understanding (Sequential/Global).

C. Learning Objects

Learning Objects (LOs) serve as instructional aids for students, comprising educational resources like videos, reading materials, and games [9]. Reference [10] highlights the potential of LOs to motivate students and enhance meaningful learning. Reference [11] stresses that LOs are digital resources crafted to support teaching and enable reusability.

Reference [12] emphasizes the connection between learning styles and learning objects. Table 1 illustrates the relationship between the styles and LOs.

TABLE I
Relationship between the format of Learning Objects and Learning Style [12]

Learning Object Format	Learning Style
Text	Sequential, Reflective, Intuitive
Image	Visual, Sensing
Audio	Verbal, Sequential
Video	Visual, Verbal
Animation	Visual, Sensing
Hypertext	Visual, Sequential, Reflective, Intuitive
Chat Application	Sensing, Visual, Active, Global

III. METHOD

This work lies in applied research geared towards generating actionable insights for practical application.

Applied research endeavors to tackle specific issues, be they at individual, group, or societal levels.

In educational software development, applied research is paramount to ensure the efficacy and efficiency of educational technologies. By delving into the needs and challenges of educators and learners, this process informs the design of educational software, tailoring it to meet their unique demands. Furthermore, research plays a pivotal role in assessing the effectiveness of these tools, ensuring they genuinely enhance the teaching and learning process and contribute to academic success, grounded in empirical evidence.

A. Adaptive System Overview

The solution consists of a programming language support system for students, which, through an adaptive system for smartphones, will have their learning supported by a recommendation mechanism for Learning Objects that are most relevant to the student's Learning Style, thus aiming to improve their performance in programming courses.

Fig. 1 illustrates the method, where (1) is the system the student interacts with; (2) the Learning Objects recommendation system; (3) the learning history (logs); (4) the Learning Style Agent; (5) the Student's Learning Style; (6) the Learning Objects Agent, and (7) the Learning Objects database.

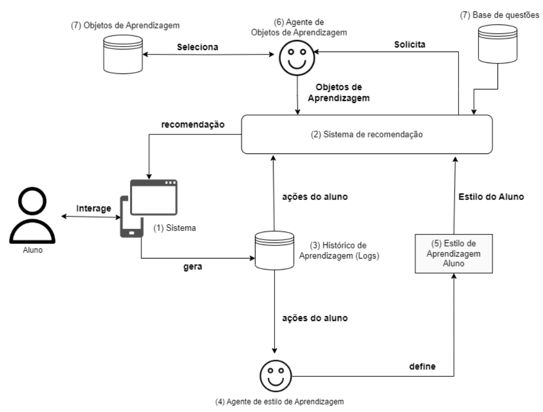

Fig. 1 Illustration of Proposed System Architecture

The Learning Objects recommendation system uses the student's actions collected from the system. These same actions are used by the Learning Style Agent to define the student's Learning Style. This Agent will constantly check if the student's LS should be reclassified.

The actions and the Learning Style will be inputs for the recommendation system, thus enabling the recommendations of Learning Objects. Once the LO is defined, the system informs the Learning Objects Agent which object should be selected from the Learning Objects database. The definitions for each stage in the recommendation system will be presented below.

B. Prototyping and Survey

To ensure software development is evidence-based and effectively supports students in achieving academic milestones, a three-phase development approach is recommended: Prototyping, development, and usability testing.

As cited in [13], prototyping is a cornerstone approach in the software development domain. It involves crafting initial versions or simplified models of software prior to full implementation. This tactic grants stakeholders insight into the software's functionality, enabling them to conduct tests and gather feedback before committing extensive resources. Prototypes can take various forms, such as paper models, interactive wireframes, or functional prototypes, all geared toward identifying and addressing potential issues, thereby enhancing the final product's quality and efficiency. Moreover, prototyping serves as a medium for communicating ideas and soliciting user feedback, ensuring software meets user needs and expectations.

Following the prototyping and development phases, a crucial step is the usability test, aimed at assessing the system's quality. This test attempts to pinpoint usability issues, evaluate efficiency, effectiveness, and user satisfaction, and recommend design enhancements. Usability tests typically involve selecting participants representative of the target user base and tasking them with specific interactions with the system. Observations, recordings, questionnaires, interviews, and other tools facilitate data collection. Analysis encompasses quantitative metrics like task completion time and errors alongside qualitative insights into user perceptions and attitudes, culminating in a comprehensive evaluation of the user experience and actionable insights for design improvements.

In this system, we adopted the System Usability Scale (SUS), a pivotal tool for gauging product, system, and user interface usability. Conceived by John Brooke in 1986 and seen in [14], SUS has evolved into a ubiquitous metric for assessing usability across diverse products and services. Its operation entails:

1. Collecting Responses: Participants respond to a standardized questionnaire comprising ten usability-related queries.

2. Evaluating Responses: Each response is scored on a scale from 1 to 5, reflecting the degree of agreement with a usability statement.

3. Calculating Scores: Individual scores are aggregated and multiplied by 2.5 to yield a total score for each participant, ranging from 0 to 100.

4. Usability Assessment: Aggregate scores undergo statistical analysis, furnishing an overarching measure of the product or system's usability.

5. Interpretation of Results: The average SUS score provides insight into overall usability quality, with higher scores indicating superior usability.

6. Comparison and Improvement: Results facilitate comparisons with previous tests or other products' usability, aiding design teams in identifying and addressing issues.

7. Qualitative Feedback: Participants' comments and qualitative feedback offer nuanced insights into specific usability challenges.

SUS emerges as a potent and straightforward tool for evaluating product and system usability, furnishing both quantitative and qualitative data to inform design decisions and enhance the user experience.

IV. RESULTS

A. Prototype

The system prototype showcases the initial screens and content recommendations for the student.

The initial screen will mark the student's first interaction with the system. Upon logging in, the student can choose which topic they wish to learn, as depicted in Fig 2.

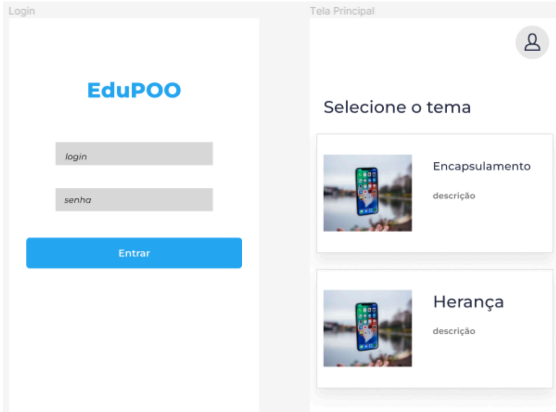

Fig. 2 Login and initial screen of the system

Fig. 3 presents the questionnaire interface that presents to the user on the top, the question itself, and below the options with the radio interface component, where the user selects the respective answer to the question. At the bottom of the screen are two buttons: a blue one that represents the confirmation written in Portuguese, the equivalent to the answer word, and below it, with the icon of an eye, the help button.

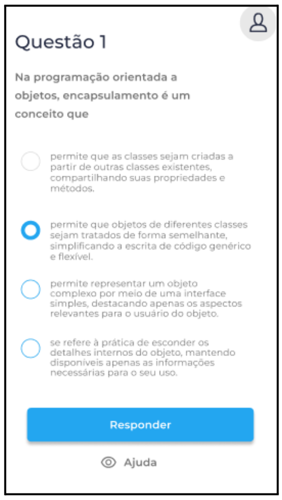

Fig. 3 depicts the content that will be presented to the student.

The system will offer the student the option to request assistance anytime. This assistance will recommend content according to the student's learning style. After each recommendation, the system will inquire about the student's experience with that learning object.

The app will feed the user with a flow of recommended content based on the theory involving learning styles versus learning objects in an artificial intelligence (AI) technique based on their own feedback (Reinforcement Learning) in each recommendation.

Reinforcement Learning (RL) is an AI technique that enables a machine to learn through experimentation and feedback. RL mimics the trial-and-error learning process that humans use to achieve their goals.

B. Development

To make the prototypes capable of being evaluated, we add to our mockups interactions for each component trying to achieve a high fidelity prototype that will behave like a fully programmed application.

That way we can focus on the feedback about how the application should work in a real life scenario, without spending much time fully developing the application.

To achieve this, we use Figma. Figma is an online collaborative design and prototyping tool that enables users to create, share, and collaborate on design projects. The 'Prototype' panel in Figma allows users to define interactions between screens of a prototype, such as navigating to another screen by interacting with a button. It also allows for creating multiple flows for a prototype on a single page to preview the user experience across designs as seen in Fig. 4.

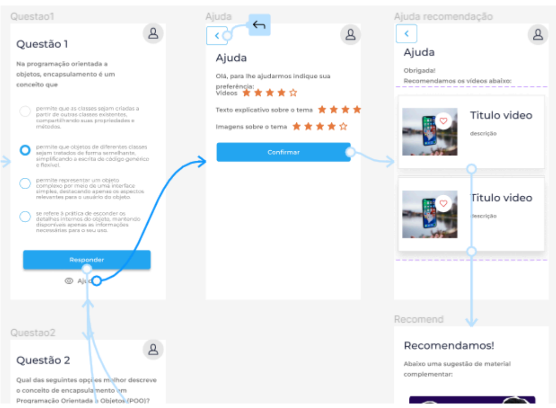

Fig. 4 Overview of Figmas Board Showing the View of Prototype that shows the Links Between Components

C. System Usability Assessment

To gain a comprehensive understanding of our system's user-friendliness, we conducted a thorough usability assessment using the System Usability Scale. This standardized 10-item questionnaire, featuring a balanced mix of positive and negative statements, was administered to a diverse group of 3 participants, including both educators (math teachers) and computer science students. This intentional diversity aimed to capture a wide range of perspectives on the system's ease of use.

Each participant rated their level of agreement with statements covering various facets of usability, such as:

Intuitiveness: How easy is it to navigate and understand the system?

Efficiency: Can users accomplish tasks quickly and with minimal effort?

Satisfaction: Do users feel positive about their overall experience with the system?

Learning Curve: How quickly can users get up to speed and become proficient?

Integration: How seamlessly do the different features and functions work together?

Upon analyzing the results, we were pleased to discover a high average SUS score of 73.33. This score, well above the 50 midpoint, firmly positions our system within the realm of positive user experiences. Participants generally found the system to be intuitive, efficient, and satisfying to use. A detailed breakdown of individual responses and the calculation methodology can be found in Fig. 5.

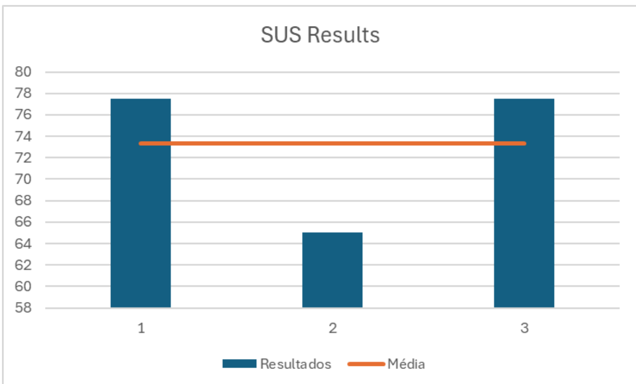

Fig. 5 Chart grouping results score by individual

In Fig. 6 and 7, it is possible to analyze the relation between the answers of the survey and the type of user (teachers of OOP Classes or not).

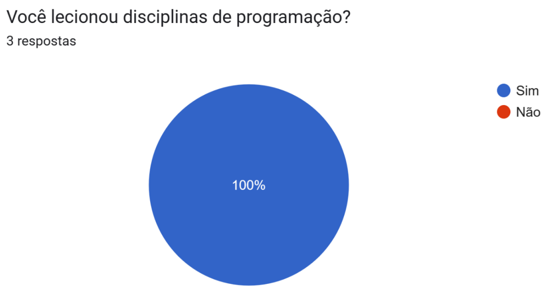

Fig. 6 The question shows the proportion of teachers of programming classes.

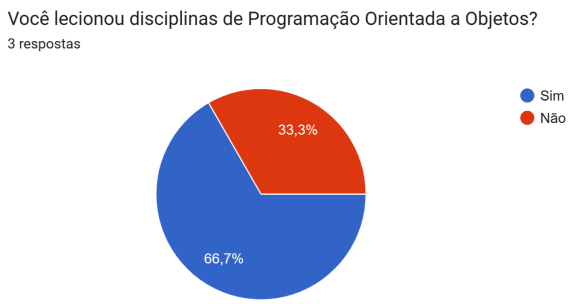

Fig. 7 Question that shows the proportion of teachers of OOP classes. Question translation: "Did you teach OOP classes?", in blue the affirmative answers.

The results about the classes that teachers experienced is important information to give us confidence about their feedback, using real experiences of programming and OOP teaching to build a better application.

These promising findings offer valuable insights for future iterations of the system. By pinpointing areas of strength and identifying opportunities for improvement, we can further enhance the user experience and ensure that our system continues to meet the evolving needs of its users.

V. CONCLUSION

With this research, it was possible to see several points of improvement in the application that will be developed. Besides that, it was confirmed through the SUS technique that the testers well received the prototype, and considering the target audience of the survey (teachers of OOP classes), they perceive the app as an excellent supportive tool in the classes.

To make the research more reliable and complete on future works we will improve the evaluation by adding qualitative questions to the survey, allowing more meaningful insights to built a better application.

The following works are intended to develop a native application alongside the AI engine, delivering a complete tool to support the learning quality of students in the OOP classes.

REFERENCES

[1] J. Van Seters, M. Ossevoort, J. Tramper, and M. J. Goedhart, "The influence of student characteristics on the use of adaptive e-learning material," Computers & Education, vol. 58, no. 3, pp. 942–952, 2012.

[2] C. V. Teixeira, K. M. R. de Lima, C. F. de Oliveira, and L. P. Moutinho, "Novas Formas de Aprendizagem: Utilização da Plataforma Adaptativa Geekie Games/ New Ways of Learning: Using the Adaptive Platform Geekie Games," Brazilian Journal of Development, vol. 7, no. 4, pp. 39765–39773, 2021, doi: 10.34117/bjdv7n4-438.

[3] A. Aslan, "Problem-based learning in live online classes: Learning achievement, problem-solving skill, communication skill, and interaction," vol. 171, 2021, doi: https://doi.org/https://doi.org/10.1016/j.compedu.2021.104237.

[4] D. P. Ausubel, "Chapter XII: Enhancing the Acquisition of Knowledge," Teachers College Record, vol. 81, no. 5, pp. 227–250, 1980, doi: 10.1177/016146818008100513.

[5] D. M. da Silva, "O impacto dos estilos de aprendizagem no ensino de contabilidade na FEA-RP/USP," phdthesis, Universidade de São Paulo, 2006.

[6] D. A. Kolb, "Experience as the source of learning and development," Upper Sadle River: Prentice Hall, 1984.

[7] R. Felder and L. K. Silverman, "Learning and teaching styles in engineering education.," Engineering Education, vol. 78, pp. 674–681, 1988.

[8] J. Valaski, A. Malucelli, and S. Reinehr, "Revisão dos modelos de estilos de aprendizagem aplicados à adaptação e personalização dos materiais de

aprendizagem," SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, Aracaju, 2011.

[9] A. Rodrigues, L. Assis, A. Andrade, C. Pitangui, and C. Silva, "Inconsistências no Processo de Recomendação de Objetos de Aprendizagem utilizando Teoria de Estilos de Aprendizagem," in Anais do XXXI Simpósio Brasileiro de Informática na Educação, 2020, pp. 1303–1312. doi: 10.5753/cbie.sbie.2020.1303.

[10] S. M. Nafea, F. Siewe, and Y. He, "A Novel Algorithm for Course Learning Object Recommendation Based on Student Learning Styles," in 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), 2019, pp. 192–201. doi: 10.1109/ITCE.2019.8646355.

[11] D. A. Wiley II, "Learning object design and sequencing theory," phdthesis, Brigham Young University, 2000.

[12] A. Al-Abri, Z. AlKhanjari, Y. Jamoussi, and N. Kraiem, "Identifying learning styles from chat conversation using ontology-based dynamic bayesian network model," in 2018 8th International Conference on Computer Science and Information Technology (CSIT), 2018, pp. 77–84.

[13] S. Soobia.et.al., "Analysis of Software Development Methodologies", Int. J. Comput. Digit. Syst., vol. 8, n.º 5, pp. 445–460, jan. 2019. doi: 10.12785/ijcds/080502.

[14] J. Brooke, "SUS: A quick and dirty usability scale," Usability Eval. Ind., vol. 189, Nov. 1995.