Issues in Information Systems
Volume 24, Issue 3, pp. 222-231, 2023

DOI: https://doi.org/10.48009/3_iis_2023_119

Adaptive programming language learning system based on generative AI

Myungjae Kwak, *Middle Georgia State University, myungjae.kwak@mga.edu*
Jonathan Jenkins, *Middle Georgia State University, jonathan.jenkins2@mga.edu*
Joobum Kim, *Middle Georgia State University, joobum.kim@mga.edu*

Abstract

In recent years, there has been an increasing recognition of the importance of early programming education. A foundational understanding of programming languages can have extensive influence, impacting problem-solving capabilities and computer science studies and future careers. Unfortunately, high failure rates in introductory programming courses have become a common phenomenon, with the cause often attributed to the abstract concepts and difficulties with programming syntax. To address these issues, intelligent tutoring systems have been developed to teach computer programming. However, most of these systems lack research on their effectiveness and are constrained to a specific institution. With the recent development of Generative AI, a new type of adaptive intelligent tutoring system can be developed to accommodate individual students and improve their understanding and performance in programming education. This system would use Generative AI to generate personalized adaptive learning materials for individual learners. The research will analyze existing tutoring systems to determine relevant features and propose an appropriate framework for a Generative AI-based adaptive learning system.

Keywords: generative AI, adaptive learning, programming language learning

Introduction

In recent years, people increasingly have come to recognize the significance of introductory coding education. The emphasis of these introductory programming courses is to improve foundational problem-solving capabilities in students and to teach them programming languages that can be used to endure through issues (Gomes & Mendes, 2007). Consequently, it is of utmost importance for Computer Science (CS) students to enhance their programming knowledge and skills since they are expected to use programming languages throughout their entire CS coursework and future career. Unfortunately, high failure rates in introductory programming classes have become a common phenomenon, inspiring researchers to investigate the reasons and potential solutions to this predicament. As several studies have suggested, learners often experience insurmountable difficulty in grasping and utilizing certain abstract concepts, data structures and algorithms when learning new programming languages (Gomes & Mendes, 2007). On top of that, researchers have proposed a multitude of possible explanations as to why programming can be so challenging: it can be complicated due to its high abstraction level and labyrinthine syntax; its learning necessitates extensive and practical study; personalized teaching is rarely possible in classroom environments; and last, but not least, students display varying learning styles.

To address the problems faced by CS educators, many intelligent tutoring systems have been developed to teach computer programming languages. These systems come with automated feedback, point-and-click reference materials, various tasks and quizzes, and efficient planning processes. Although there are many approaches for making AI-based educational tools, there is not much research on their effectiveness and comparative analysis. Additionally, most of these systems are implemented to fulfill specific needs of institutions and are not open to the general public. Therefore, there is a need to create an adaptive intelligent tutoring system, utilizing AI, that can accommodate individual students and improve their understanding and skill-level in programming.

Recent progress and expansion in machine learning has led to more sophisticated, innovative digital content generations such as Generative Artificial Intelligence (AI) (Lim, Gunasekara, Pallant, Pallant, & Pechenkina, 2023). Generative AI is an unsupervised or partially supervised machine learning framework, which uses statistics and probabilities to generate manmade artifacts (L, 2022). The latest advancements of Deep Learning (DL) allows Generative AI to create artificial artifacts using existing digital content, such as video, images/graphics, text, audio, etc., by learning their patterns and distributions. Additionally, Generative AI can also generate programming codes for specific tasks and debug programming errors.

This research aims to analyze existing intelligent adaptive tutoring systems, choose the most relevant and essential features, propose an architecture of a Generative AI-based adaptive programming language learning system.

Literature Review

Adaptive Learning Systems

Currently, studies on adaptive learning focus on creating learning platforms that take into account learners' needs, motivations, and favored learning styles (Ennouamani & Mahani, 2017). This underlines the concept of adaptive learning systems as an alternative to the traditional "one-size-fits-all" approach in developing teaching materials (Ennouamani & Mahani, 2017). The authors in (Jonassen, et al., 2008) define adaptation as the system's ability to modify its behavior based on the learners' knowledge levels and other characteristics (Ennouamani & Mahani, 2017).

Li, He, and Xue (2021) argued that adaptive learning systems can be varied in terms of function and how the content is displayed. Normally, there are three core procedures to implement adaptive learning: adapting material choice, providing adaptive navigation assistance, and optimizing content delivery (Li, He, & Xue, 2021; Brusilovsky, 2012). Essential technologies to make personalized learning and teaching assistance a reality are largely comprised of data mining, learning assessment, artificial intelligence, knowledge mapping, cognitive advisor, guided studying, edge computing, virtual reality, etc. (Li, He, & Xue, 2021).

Intelligent Programming Language Learning Systems

Many researchers have designed and developed intelligent learning systems for CS and programming language education. Müller, Bergande, and Brune (2018) demonstrated the viability of IBM Watson as a cognitive software for creating a virtual teacher to answer the common questions from students within an introductory Java programming class. An evaluation was conducted by means of a qualitative study with chosen members of the course, and the prototype software was examined in a field test. The findings of the assessment suggest that it could be used for a course with the particular purpose (Müller, Bergande, & Brune, 2018). Marouf, Yousef, Mukhaimer, and Abu-Naser (2018) designed an intelligent tutoring system called Computer Science Course Tutor, which dynamically adapts to the progress of CS students (Marouf, Yousef, Mukhaimer, & Abu-Naser, 2018).

Generative Artificial Intelligence

Generative AI is a branch of artificial intelligence that focuses on creating new content or information. It involves training models to generate original and realistic outputs such as images, music, text, or even entire video sequences. These models use complex algorithms and large datasets to learn patterns and generate digital content that resembles human-created data (Dwivedi, et al., 2023).

Generative AI can be used for learning by creating interactive and engaging educational materials. It can generate customized learning content tailored to individual students' needs and preferences, enhancing the learning experience. Additionally, generative AI can be used to create virtual tutors or digital assistants that can provide personalized guidance and support to learners. By leveraging generative AI, educational institutions can harness the power of technology to create adaptive and dynamic learning environments that promote creativity and active participation (Lim et al., 2023).

Recent surveys of artificial intelligence assisted learning systems demonstrate system benefits in terms of adaptability and technical measures but remain limited in practical application in real learning scenarios and student learning benefits (Kabudi et al., 2021).

Adaptive Learning System based on Generative AI

Architecture of Adaptive Learning System based on Generative AI

We propose an architecture of adaptive learning system based on generative AI as shown in Figure 1. The system was devised by referring to the adaptive e-learning systems' components diagram summarized by (Ennouamani & Mahani, 2017) and consists of three modules: Learning Module, Knowledge Domain Module, and Adaptation Module.

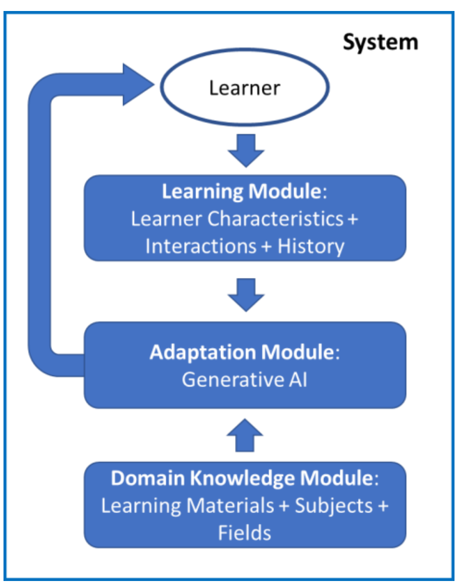

Figure 1: Generative AI-based Adaptive Learning System Components

Learning Module continuously collects and analyzes various data about learners, including their learning styles, preferences, interactions, and history, to provide them with a personalized adaptive learning experience. Domain Knowledge Module organizes domain knowledge - like learning materials, topics, and fields - into an abstract form. Adaptation Module then continuously combines data from the Learning Module and Domain Knowledge Module to create adaptive learning materials generated by Generative AI, tailored to the individual learner's needs and pace.

User Interface

Figure 2 illustrates an example of a user interface used by learners to interact with the system. The system produces learning tasks and questions within a particular domain by drawing on the data from the Learning Module and the Domain Knowledge Module. If the user completes the task successfully, they will be taken to the next level. If the user is unsuccessful, the system will provide feedback on how to bolster their knowledge and present easier or related tasks. This learning process will continue until the learner achieves a pre-defined level of knowledge in a particular subject.

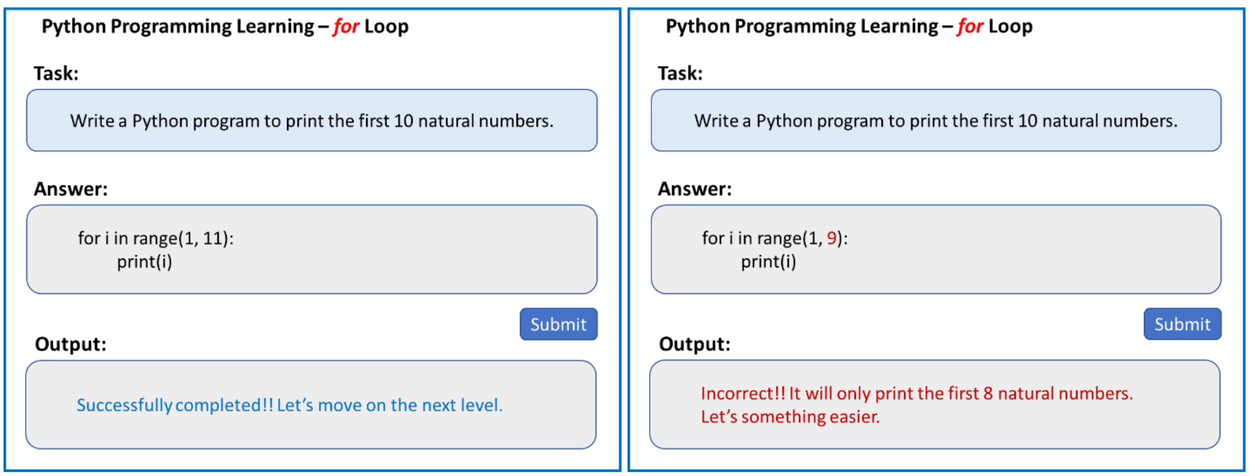

Figure 2: User Interface of Generative AI-based Adaptive Learning System

Feasibility Test

To test the feasibility of the system, we asked Bard, which is a conversational generative artificial intelligence chatbot developed by Google (Pichai, 2023), to generate five python programming questions related to a particular programming concept, and it was able to do so successfully, as shown in Figure 3. We also tested to see if the generative AI can evaluate the response from the learner, and it was able to do so with proper helpful explanations, as shown in Figure 4.

The results of the two feasibility tests indicated that, by leveraging Generative AI, tasks in a particular field could be generated for learners, their responses evaluated, and appropriate results with helpful explanations produced.

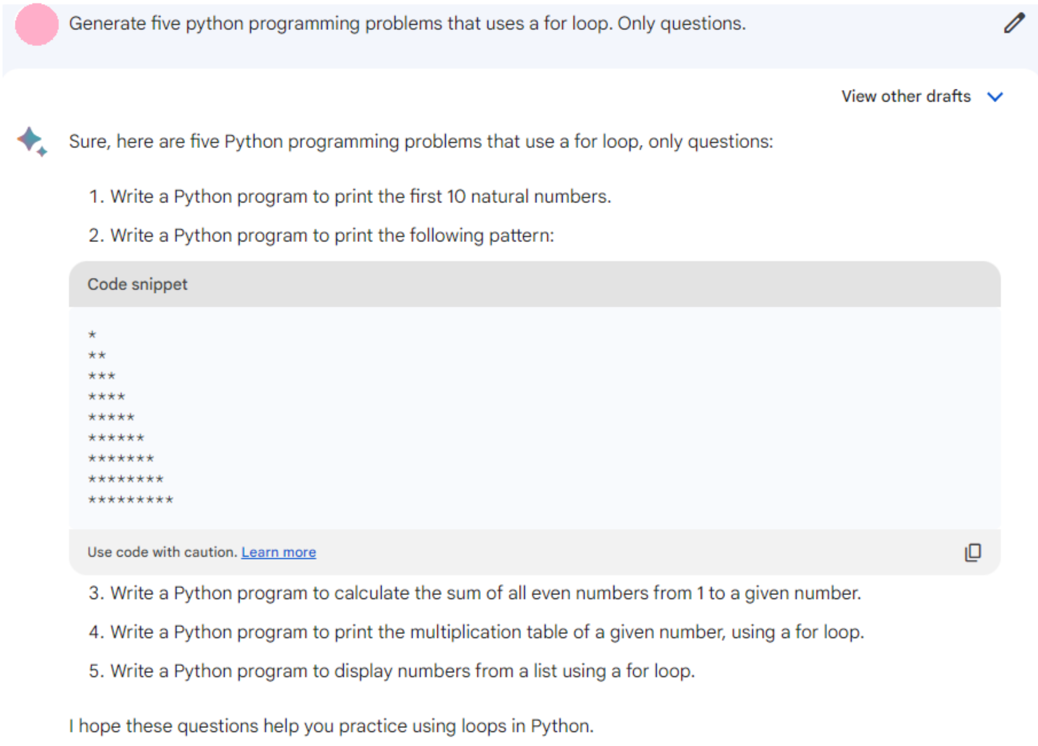

Figure 3: Feasibility Test using Google Bard – Task Generation

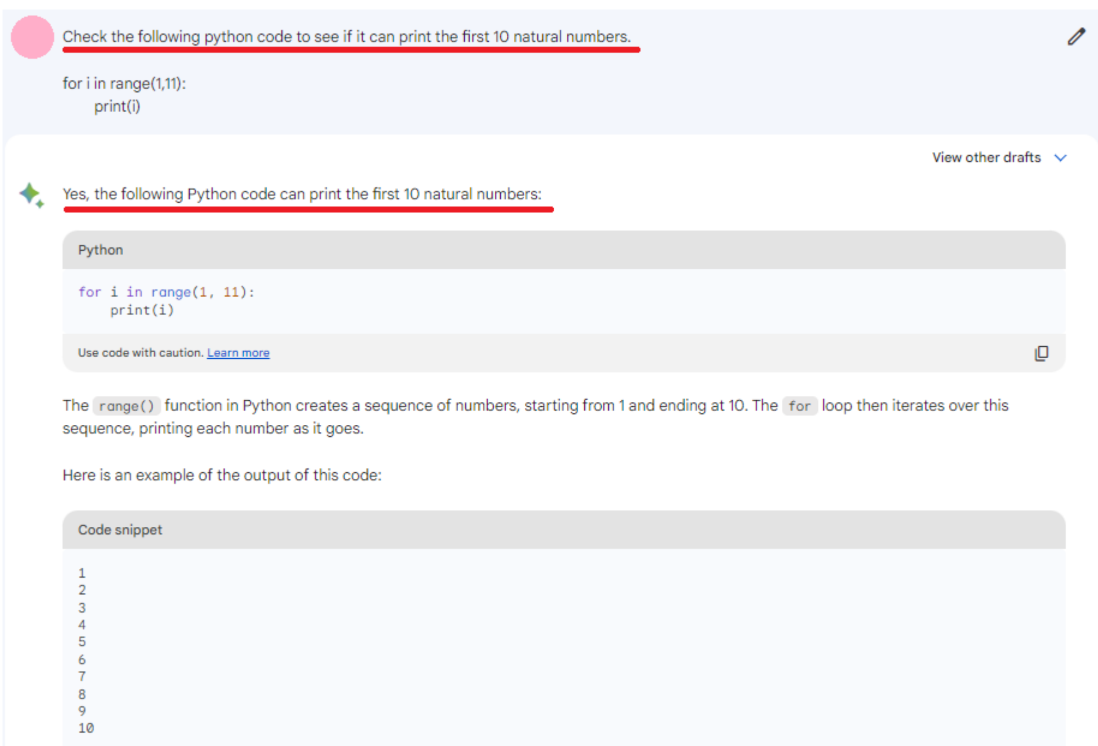

Figure 4: Feasibility Test using Google Bard – Response Analysis

Development of Prototype

To develop a prototype of a generative AI-based adaptive learning system for introductory CS courses as a proof of concept for programming language education, we evaluated three large language models, OpenAI's GPT-3 and GPT-4 (OpenAI, 2023), Google's LaMDA (Language Models for Dialogue Applications) (Collins & Ghahramani, 2021), and Meta's LLaMA (Large Language Model Meta AI) (Meta, 2023) to see if they can be customized for the proposed adaptive learning system. We have discovered that while all three models are capability of doing the desired tasks, OpenAI offers relatively easier-to-use APIs for application developers. Therefore, we have developed a prototype system using OpenAI's GPT-3.5-turbo model (OpenAI, 2023).

We utilized Python and the Python web framework Django to construct the prototype system in the form of a web application. Figure 5 displays the Python code used to formulate a programming question in relation to a certain topic, while Figure 6 illustrates the Python code employed to evaluate the user's answer.

```
import openai
openai.organization = "ORGANIZATION_ID"
openai.api_key = "OPENAI_API_KEY"
completion = openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[
    {"role": "system", "content": "You are a helpful programming language learning assistant."},
    {"role": "user", "content": "Generate a python programming question that uses a for loop. Only question."}
  ]
)
completion.choices[0].message.content
```

Figure 5: *Python Code for* Task Generation

```
import openai
openai.organization = "ORGANIZATION_ID"
openai.api_key = "OPENAI_API_KEY"
completion = openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[
    {"role": "system", "content": "You are a helpful programming language learning assistant."},
    {"role": "user", "content": """Evaluate the following python code if it can print the first 10 natural numbers.
        for i in range(1,11):
            print(i)
     """}
  ]
)
completion.choices[0].message.content
```

Figure 6: Python Code for Task Evaluation

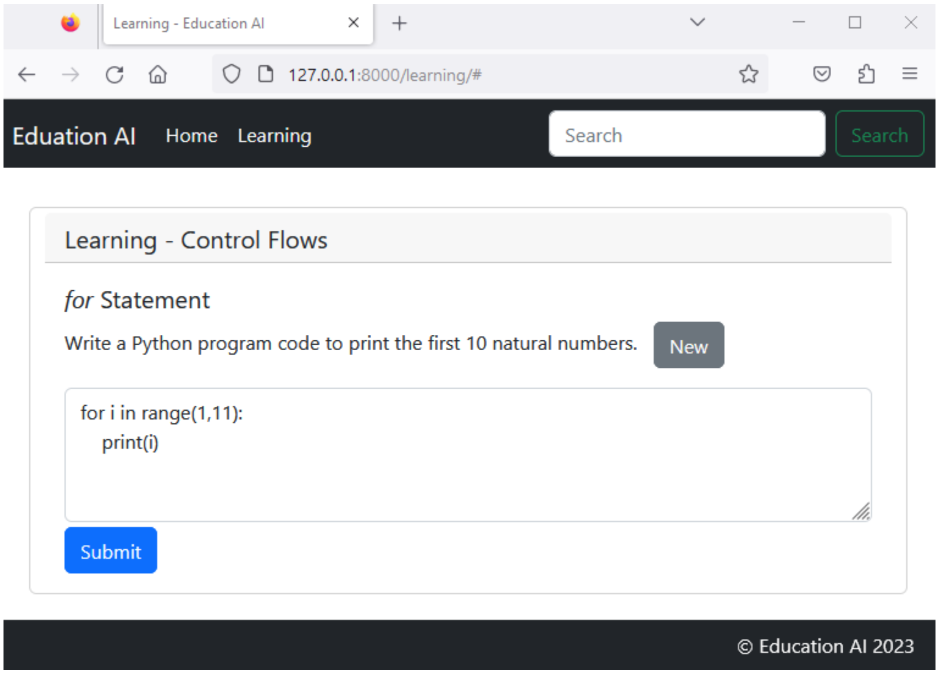

Figure 7: User Interface for Task Generation and Answer Submission

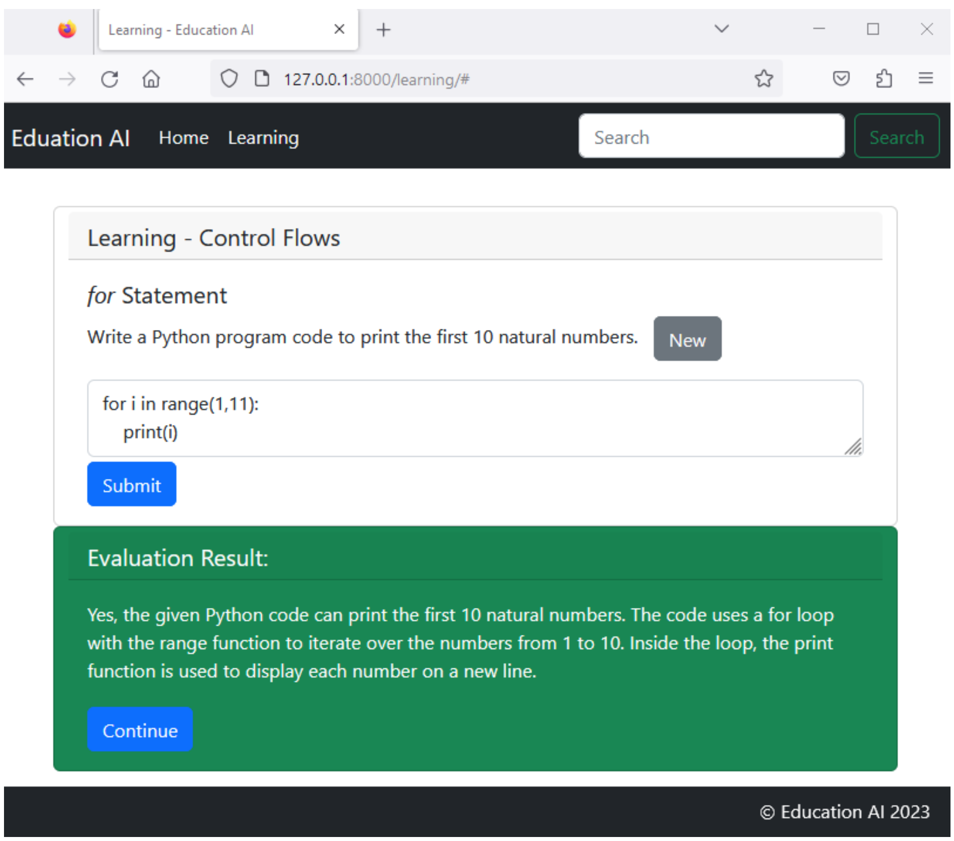

Figure 8: User Interface for Evaluation Result

Figure 7 shows the user interface of the prototype system when a specific learning module is selected. The AI then generates a programming question and presents it to the user. After submitting their response, the AI evaluates it and provides the user with detailed feedback, as illustrated in Figure 8. After reviewing the feedback, the learner can move on to the next task by clicking the Continue button.

In addition to developing the prototype system as a proof of concept, we are investigating what data is required and how it should be structured in the Learning and Domain Knowledge Modules, in order for the Adaptation Module to better generate adaptable learning materials for individual learners.

Discussion and Conclusion

It is essential for CS students to build up their programming proficiency, as they will need to utilize programming languages in all their CS classes and in their future professional lives. Unfortunately, high failure rates in introductory programming courses have become increasingly prevalent, prompting researchers to investigate the causes and devise potential solutions to this obstacle. In an effort to help teach computer programming, CS educators and researchers have developed a number of intelligent learning systems. These systems usually provide automated feedback, point-and-click reference materials, various tasks and quizzes, and effective planning processes.

Personalized adaptive learning has demonstrated its effectiveness in improving learner's performance. In this research, we proposed an adaptive learning system architecture based on Generative AI, a rising technology for digital content generation. We examined if Generative AI can be applied for context-based adaptive learning and validated its feasibility. The proposed system consists of three components including Learning Module, Domain Knowledge Module, and Adaptation Module and is designed to interact with individual learner and provide adaptable learning materials for them. We have developed a prototype of the proposed system as a proof of concept using OpenAI GPT-3.5-turbo model. We are currently enhancing the three main components.

This work has demonstrated that current A.I. capabilities allow the generation of a set of assessments for a limited complexity software development topic and the assessment of the compliance of an assessment response with a software requirement. However, key to the application of artificial intelligence to learner adaptation is the investigation of the model of assessment adaptation, since it is conceivable that multiple mutually exclusive code-resident alterations could progress the assessment toward the true reflection of the learner’s skill. Several potential model elements are possible, including the reduction of abstract task complexity, substitution of code elements (including different skill level techniques yet with equivalent effect) and relabeling.

A serviceable, initial model of task adaptation is to select the new assessment from a set of assessments classified to match the inferred skill “level” of the learner (whether the inferred level exceeds or does not reach the one implied by the current assessment). In the context of existing work, the approach advocated in this work can be classified as an application of adaptation of the materials to the learner.

The acquisition of programming problem domain knowledge and its delivery to the adaptation module is a similarly critical component in the operation of the proposed framework. Initial proposals for the operation of the domain knowledge module include the provision of the programming language structure type and software requirement as inputs to the adaptation module.

An additional pathway for investigation is the characterization of the adaptation goal, since there are valid alternative objectives for learner skill development, from the development of initial mastery of a particular topic to the maintenance of skill. As an initial effort, a possible approach is the iterative increase of expert-assessed difficulty in response to successful outcomes. In future work, the degree to which learner-specific patterns can be considered by the adaptive learning system can be investigated.

We anticipate that the proposed system can enhance learning for some CS students by providing personalized learning materials for certain programming concepts. However, this research has certain limitations, including the fact that the proposed system has yet to be developed and examined in an appropriate manner. Consequently, it is uncertain if it will function as intended or be effective.

References

Brusilovsky, P. (2012). *Adaptive hypermedia for education and training.* Cambridge, United Kingdom: Cambridge University Press.

Collins, E., & Ghahramani, Z. (2021). *LaMDA: our breakthrough conversation technology.* Retrieved from Google: https://blog.google/technology/ai/lamda/

Dwivedi, Y., Kshetri, N., Hughes, L., Slade, E., Jeyaraj, A., Kar, A., . . . Chowdhury, S. (2023). "So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. *International Journal of Information Management, 71*, 102642.

Ennouamani, S., & Mahani, Z. (2017). An overview of adaptive e-learning systems. *Proceedings of the 2017 eighth international conference on intelligent computing and information systems (ICICIS)* (pp. 342-347). IEEE.

Gomes, A., & Mendes, A. (2007). An environment to improve programming education. *Proceedings of the 2007 international conference on Computer systems and technologies*, (pp. 1-6).

Jonassen, D., Spector, M., Driscoll, M., Merrill, M., van Merrienboer, J., & Driscoll, M. (2008). *Handbook of research on educational communications and technology: a project of the association for educational communications and technology.* Routledge.

Kabudi, T., Pappas, I., & Olsen, D. H. (2021). AI-enabled adaptive learning systems: A systematic mapping of the literature. *Computers and Education: Artificial Intelligence, 2*, 100017.

L, H. (2022). *Generative AI and Future.* Retrieved from Medium: https://pub.towardsai.net/generative-ai-and-future-c3b1695876f2

Li, F., He, Y., & Xue, Q. (2021). Progress, challenges and countermeasures of adaptive learning. *Educational Technology & Society, 24*(3), 238-255.

Lim, W., Gunasekara, A., Pallant, J., Pallant, J., & Pechenkina, E. (2023). Generative AI and the future of education: Ragnarök or reformation? A paradoxical perspective from management educators. *The International Journal of Management Education, 21*(2).

Marouf, A., Yousef, M., Mukhaimer, M., & Abu-Naser, S. (2018). An Intelligent Tutoring System for Learning Introduction to Computer Science. *International Journal of Academic Multidisciplinary Research (IJAMR)*, 1-8.

Meta. (2023, February). *Introducing LLaMA: A foundational, 65-billion-parameter large language model.* Retrieved from MetaAI: https://ai.facebook.com/blog/large-language-model-llama-meta-ai/

Müller, S., Bergande, B., & Brune, P. (2018). Robot tutoring: On the feasibility of using cognitive systems as tutors in introductory programming education: A teaching experiment. *Proceedings of the 3rd European Conference of Software Engineering Education*, (pp. 45-49).

OpenAI. (2023). *GPT-4.* Retrieved from OpenAI: https://openai.com/research/gpt-4

Pichai, S. (2023, 2 06). *An important next step on our AI journey.* Retrieved from Google: https://blog.google/technology/ai/bard-google-ai-search-updates/