An Adaptive Learning Support System based on Ontology of Multiple Programming Languages

Lalita NA NONGKHAI[a*], Jingyun WANG[b] & Takahiko MENDORI[c]
[a] *Graduate School of Engineering, Kochi University of Technology, Japan*
[b] *Department of Computer Science, Durham University, UK*
[c] *Graduate School of Engineering, Kochi University of Technology, Japan*
* 256010u@gs.kochi-tech.ac.jp

Abstract: This research proposes to develop an adaptive ontology-based learning support system for computer programming learning. Firstly, the system adopts a previously developed ontology called CONTINUOUS, which represents programming concepts and their relation in a graph and makes use of its content to serve as hints within programming questions. Secondly, we design an adaptive strategy for recommending suitable exercises to learners, which uses CONTINUOUS as metadata of exercises and the Elo rating system to estimate learners' skills. This work aims to design a system to provide personalized exercise path for the learner and evaluate its effectiveness.

Keywords: Adaptive Learning Support, Ontology, Recommendation System, personalized learning, programming learning

1. Introduction

Computer programming utilizes coding, debugging, and evaluating of solutions (Friday et al., 2019). However, the complexity of programming problems can confuse learners regarding the necessary knowledge for the solution (Yorah et al., 2016). To improve the efficiency of learning CT (Computational Thinking), this research focuses on developing a web-based Learning Support System (LSS) based on ontology to recommend suitable programming exercises for learners. There are several educational systems leverage ontology. For instances, Ibrahim et al. (2017) utilized three ontologies to suggest suitable course for university students, and Bouihi et al. (2019) proposed an approach to apply their ontologies in recommendation system to support learners within their learning experiences by recommending appropriate materials.

In our previous work (Lalita et al., 2022), we developed an ontology named CONTINUOUS, encompassing common concepts from Python, Java, and C#, describing programming concepts, sub-concepts, and concept relation. This research adapts CONTINUOUS to visualize programming concepts as a graph to help learners understand the related concepts in a specific programming language. Furthermore, the design principle of CONTINUOUS determines knowledge points (KP) as minimal syntax of a specific programming language and enables KP to serve as a hint of each programming question to guide learners in solving a specific problem.

Moreover, this research also focused on designing a personalized recommendation approach to enhance learner programming skills. Previously, Zheng et al. (2022) have devised a knowledge structure tree for C programming, leveraging it as meta-data in their recommendation system. They used the matching degree calculation to match score between the student’s cognitive level and exercise difficulty to facilitate personalized exercise recommendations. Similarly, Michlík et al. (2010) have employed linear extrapolation to estimate potential knowledge levels in exercise recommendations, with the goal of aiding learners in achieving favorable exam results within a limited time. To enhance the assessment of learner’s understanding level, we adapt the Elo Rating System (ERS) (Elo, 1978), which is a method used to assess and measure the relative proficiency of the participants in competitive games and has also been utilized in education research to estimate learner's skill and item difficulty (Radek, 2016). Additionally, Klinkenberg et al. (2011) demonstrated the efficacy of ERS in assessing player skill and item ratings within the Math Garden, a mathematics game for primary students.

In summary, the contribution of this research is the adoption of CONTINUOUS in an LSS and the design of an adaptive strategy for exercise recommendations to explore whether this LSS can effectively enhance learners' programming skills.

2. Development of an LSS Based on Ontology in Computer Programming

2.1 The adoption of CONTINUOUS with the system

Our system utilizes CONTINUOUS to visualize programming concepts as a graph to guide learners with a clear understanding of the relation between concepts. For instance, as shown in Figure 1, if learners decide to learn the "data_structure" concept in Python, they have the option to practice with "List" or "Dictionary." Opting to study the "List" concept necessitates understanding of the associated "list_method", which serves as a function for manipulating elements within the "List" concept.

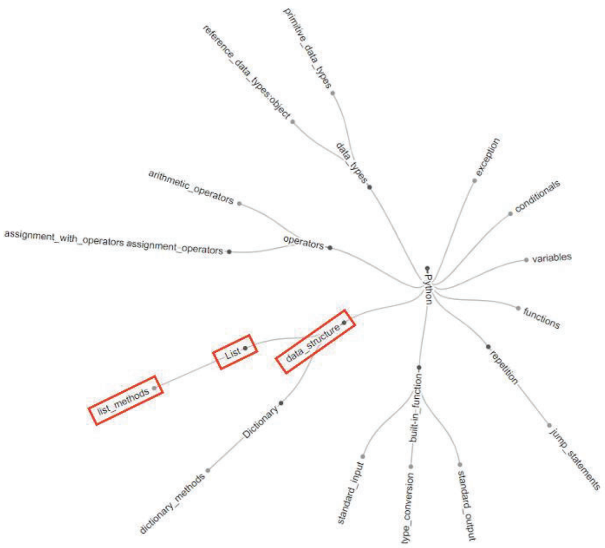

Figure 1. A Python-based visualization of programming concepts from CONTINUOUS.

In addition, the contents in CONTINUOUS are also used as a hint in programming questions, depending on the code answer in each question. Currently, our system contains questions from a question bank, which has 68 questions in three levels of difficulty (easy, standard, and difficult) involving concepts in CONTINUOUS. For example, when learners are tasked with verifying the equivalence of input numbers, which involves the concepts of "conditionals" for decision-making and "standard_input" for keyboard input, the system will assist learners in grasping essential concepts ("conditionals" and "standard_input") for addressing the question.

2.2 The design of adaptive strategy to recommend suitable programming exercises

To recommend suitable programming exercises for learners, the system will update a learner's skill based on the formulation of ERS presented by Radek (2016) in an educational setting as follows:

$$\theta_s := \theta_s + K \cdot \left(correct_{si} - P(correct_{si} = 1)\right) \qquad (1)$$

From formulation 1, we denote θ_s as learner's skill of a learner s, the value of K as a learning rate, which is a weight value in the formulation, $correct_{si}$ as the correctness of code answer of a learner s on an exercise i and $P(correct_{si} = 1)$ as the probability of a correct answers as shown in formulation 2:

$$P(correct_{si} = 1) = \frac{1}{(1 + e^{-(\theta_s - d_i)})} \qquad (2)$$

The initial value of θ_s is set to 0. After code submission, the system evaluates it with the testcases. If all testcases are passed, $correct_{si}$ is set to 1; otherwise, it is set to 0. Learner's skill is estimated in the range of 0 to 1 and compared within three difficulty levels of exercises. If the result is nearly 0 indicates a slight increase of maintain the same, while the result close to 1 which means learner's skill increase, leading to the system providing the upper level. If learners reach the highest level within the current concept, the system will show the practice process of one of its related concepts to learners.

3. Conclusion and Future Work

This research focuses on developing an LSS, which adopts CONTINUOUS to visualize programming concepts as a graph, enabling learners to understand the relation between concepts within a specific language. Additionally, the system makes CONTINUOUS serve as exercise metadata and uses the programming concepts of CONTINUOUS as hints to support learners during problem-solving. Furthermore, this research designs an adaptive strategy for exercise recommendations, utilizing the ERS formulation to estimate and update learners' skills. In the future, we plan to conduct several experiments to evaluate the system and answer the research questions: whether our system which provides personalized exercise recommendation based on individual understanding levels, can effectively enhance learners' programming skills. Learning log data will also be collected and analyzed.

References

Bouihi, B., & Bahaj, M. (2019). Ontology and Rule-Based Recommender System for E-learning Applications. International Journal of Emerging Technologies in Learning 14(15), 4–13.

Elo, A. (1978). The rating of Chessplayers, Past and present. New York: Arco Publishers.

Friday, J. A., Solomon, S. O., Jarkko, S., & Sunday, A. (2019). A Systematic Review of Computational Thinking Approach for Programming Education in Higher Education Institutions. Proceedings of 19th Koli Calling International Conference on Computing Education Research on Proceedings (pp. 1-10). Association for Computing Machinery.

Ibrahim, M.E., Yang, Y., & Ndzi, D. (2017). Using Ontology for Personalised Course Recommendation Applications. International Conference on Computational Science and Its Applications. Springer, Cham.

Klinkenberg, S., Straatemeier, M., & van der Mass H.L.J. (2011). Computer adaptive practice of Maths ability using a new item response model for on the fly ability and difficulty estimation. Computers & Education 57(2011). 1813-1824.

Lalita, N., Jingyun, W., & Mendori, T. (2022). Developing an Ontology Of Multiple Programming Languages From The Perspective Of Computational Thinking Education. Proceedings of 19th International Conference on Cognition and Exploratory Learning in Digital Age (pp. 66-72). International Association for Development of the Information Society.

Michlík, P., & Bieliková, M. (2010). Exercises recommending for limited time learning. Procedia Computer Science 1(2), 2821-2828.

Radek, P. (2016). Applications of the Elo rating system in adaptive educational systems. Computers & Education 98(2016). 169-179.

Yorah, B., & Macro, A. G. (2016). Why is programming so difficult to learn? Patterns of Difficulties Related to Programming Learning Mid-Stage. ACM SIGSOFT Software Engineering Notes, 41(6), 1-6.

Zheng, W., Du, Q., Fan, Y., Tan, L., Xia, C., & Yang, F. (2022). A personalized programming exercise recommendation algorithm based on Knowledge Structure Tree. Journal of Intelligent & Fuzzy Systems 42(3), 2169-2180.