Using LLMs to Adapt Serious Games with Educators in the Loop

Federico Bonetti[1], Antonio Bucchiarone[1], and Vanissa Wanick[2]

[1] Motivational Digital Systems (MoDiS) - Fondazione Bruno Kessler, Trento, Italy
{fbonetti,bucchiarone}@fbk.eu
[2] Winchester School of Art, University of Southampton, UK
vwv1n12@soton.ac.uk

Abstract. Recently, the advancement of generative AI has brought about the opportunity to adapt and personalize learning material to individual students with little effort. This paper explores the application of large language models, such as ChatGPT, to help educators enhance and adapt educational serious games at runtime. Adaptive serious games can benefit from this opportunity, particularly by incorporating generative AI within the MAPE-K loop framework, which provides a systematic approach to monitoring, analysing, planning, and executing adaptations in real time. A key focus is the inclusion of educators in the adaptation process, ensuring that AI-driven changes align with educational goals. Based on this, we propose an architecture that integrates player/learner data, game logic, and AI-generated adaptations, monitored and approved by educators via a dedicated browser-based dashboard, in a human-in-the-loop fashion. We show how we integrated this architecture into Untitled Bee Game, an existing educational serious game for eco-sustainability.

Keywords: serious games · adaptation · mape-k loop · education · sustainability.

1 Introduction

Generative Artificial Intelligence (GAI) refers to the capability of machines to autonomously generate content, such as text, images, and audio [11]. Recent generation models, such as large language models (LLMs) like ChatGPT and Gemini, and text-to-image models like DALL·E 3 and MidJourney represent a viable direction to explore new possibilities to generate game content and address previously identified challenges in educational serious games (SGs) [18,14]. ChatGPT, for instance, is a fine-tuned dialogue-oriented version of the GPT model (the latest version being 4.0o at the time of writing) which took advantage of billions of parameters during training. These models allow the execution of complex tasks even without specific fine-tuning, thanks to the ability of these models to learn the task in the context of a single prompt, a capability called in-context learning (ICL) [29]. Tasks may range from creative text generation to next-item prediction (e.g., the next optimal challenge for a particular type of user). Through *textification* of data (e.g., player/learner profile data, game session data, etc.) and prompt engineering techniques that produce standardized, reusable templates [29,9], challenges in content generation and adaptation for SGs can be faced in new ways. In the field of game-based learning, for example, [12] have applied a similar approach to the design of educational escape rooms, by using prompts to obtain content from ChatGPT. Although this may sound promising, little has been discussed regarding the applicability of these strategies to *adapt educational SGs at runtime using LLMs*. This is particularly crucial in educational settings where maintaining attention and motivation facilitates effective learning experiences. By integrating AI-driven generation and adaptation processes within SGs, we aim to enable educators to dynamically tailor the gameplay experience to their learners. Yet one of the main challenges is how to include humans (educators in particular) in the loop to monitor and assess the adaptations carried out by the system [10,4,14]. Technical limitations of the AI, such as the phenomenon of hallucination [20,27], further highlight the need to include educators in the adaptation stack. In our work, we address this issue by proposing an architecture for AI-driven serious game adaptation with educators in the loop. Ideally, such a loop would need a phase where data about the players/learners and the in-game actions are collected and analysed, and one where the correct adaptations are planned with the help of the educator and then integrated into the system dynamically. Data should also be stored to make future adaptations faster and improve the materials. We identified the MAPE-K loop [1,15] as the most appropriate framework for such an approach as it involves a Monitoring, an Analysing, a Planning, and an Execution phase, as well as a data repository represented by the Knowledge component. In addition, it has already been applied to adaptive serious games [26] and has significant overlap with some concepts derived in serious game frameworks [14] as explained in Section 3.

The research questions guiding our work are: *How can generative AI aid SG adaptation? And how can educators be involved in the process?* The remainder of this paper is organized as follows: In Section 2 we describe the related work on SG adaptation, player modelling, and self-adaptive models such as the MAPE-K loop. In Section 3 we outline the architecture we propose in terms of modules and runtime loop. In Section 4 we show how our architecture is applied to an existing educational SG, namely Untitled Bee Game. Finally, we discuss limitations and future work in Section 5.

2 Related Work

2.1 Player modelling and adaptation in SGs

SGs can be defined as games whose primary purpose goes beyond amusement alone as they integrate educational design and content and do not have entertainment as their primary purpose [2,21]. SGs may be designed for education, well-being, military training, advertisement, and healthcare, among other purposes [17,13]. SGs pose a significant challenge in terms of design and development, as they need to engage players while pursuing a concurrent objective (e.g., learning or exercising) and provide personalized experiences. Thus, AI strategies might help software engineers and designers exploit the potential of personalization in SGs. Areas in which AI is commonly used in SGs include assessment of scores, game design, difficulty balancing, player profiling, and content adaptation [25,28]. Content adaptation, in particular, is key to overcoming the one-size-fits-all approach [24,8] and tailoring the experience to individual user needs, and is one of the key success factors of SGs identified by [17]. Many attempts exist that try to establish frameworks to categorize and cluster players [3,19] and to address the improvement of educational SGs with player profiles and types [7]. In their review, Hare and Tang [14] compare different player modelling and adaptation strategies. Two key modules are essential for adaptive serious games: a player model and an adaptation module [14]. These two modules are included in our architecture in Section 3, which is largely based on the framework extracted by [14]. The framework by Berger and Müller [4] is also relevant, as they extract a flowchart for explainable adaptive AI where manipulated variables (similar to the MAPE-K effectors) influence players while they interact with other variables (similar to the MAPE-K sensors); adaptation is supervised by a human. Speaking of humans in the loop, in a similar vein Dobrovsky et al. [10] introduce an educator-in-the-loop approach to correct serious game adaptations.

2.2 MAPE-K loop

The MAPE-K loop is a fundamental concept in autonomic computing, providing a systematic approach to managing and optimising complex systems [1,15]. The MAPE-K loop consists of four interconnected phases over a *Knowledge base*: *Monitor*, *Analyse*, *Plan*, and *Execute*. The *Monitoring* phase involves the continuous observation and collection of data regarding the system's behaviour, performance metrics, and environmental factors. *Monitoring* helps to assess the current state of the system and detect any deviations from expectations. In the *Analysis* phase, the collected data is processed and analysed to identify patterns, anomalies, and potential issues within the system. Based on the analysis stage, decisions are made in the Plan phase on how to adapt the system to meet its objectives. This may involve formulating strategies, setting goals, or generating action plans to improve system performance. In the *Execution* phase, the planned actions or changes are implemented within the system. Finally, the *Knowledge* base is employed for capturing and storing the lessons learned and domain knowledge gained throughout the process. In the literature, various uses of MAPE-K loops can be found across different application domains (i.e., cloud, cyber-physical systems, Internet of Things) [23]. In the field of adaptive SGs, The MAPE-K loop has been adopted for dynamic difficulty adjustment in an exergame where physiotherapists can monitor their patients' cycling performance [26]. These examples underscore the versatility and effectiveness of MAPE-K loops in diverse contexts, highlighting their significance in the provision of autonomous and adaptive systems.

3 Proposed Architecture

Hare and Tang [14] propose a framework that involves 5 entities or components and that constitutes the basis of our architecture: a *Player*, a *Virtual Environment*, an *Adaptation Module*, a *Player Model*, and *External Sensors* (as depicted in Figure 1). We keep the *Player* (and name it Player/Learner) and the *Adaptation Module* and introduce small changes for the other components: we split the *Virtual Environment* into *Game Front-end* and the *Game Back-end*. The rationale behind this choice is that it should be possible for researchers and practitioners to decide where the actual game mechanic computations are carried out (i.e., on the player's computer versus the server). The Game Front-end module contains the *Effectors* and *Sensors* (overlapping with External Sensors) provided by the MAPE-K loop. These consist mainly of dynamic *Game Parameters* and *Interaction Logs* respectively. *Game Parameters* in our case study reflect the style and difficulty of educational exercises such as reading and quizzes. The *Game Back-end* contains the logic and data needed to run the game. The *Player Model* by [14] is contained in the Knowledge component in the form of *User Data*. The User Data should include anything that reflects how the Player/Learner interacts with the SG. For example, learning scores, selected preferences, and interaction logs such as reading times, reading speed, the tendency to skip dialogues, and so forth. Our architecture also largely conforms to the model extracted by the authors in [4], where their *Game* component (the Game Front-end/Back-end) is divided into *Player-influenced Variables* (the Sensors) and *Manipulated Variables* (the Effectors); their *Adaptation Function* is mirrored by the Adaptation Module, except we treat the user data (the Knowledge) separately. Our proposed architecture, involving the MAPE-K loop, thus conforms to the previous literature while being integrated with LLMs (through ad hoc APIs).

In their architecture, [14] conceptualize the loop as a sequence in which the Player interacts with the Virtual Environment, their data are saved and analysed, together with those collected by the External Sensors, and then the Adaptation Module adapts the Virtual Environment. Our approach is similar, except we add the Educator entity and a Dashboard to control AI's adaptations. Here we propose the step-by-step of the architecture flow (see Figure 1). In the ***Monitoring*** phase, the Player/Learner interactions are saved, and the Game Logic and Data are updated along with the User Data stored in the Knowledge component. In the ***Analysis*** phase, the AI produces human-readable statistics that the Educator evaluates. Then, in the ***Planning*** phase, the AI produces potential plans that the Educator may greenlight or customize. Finally, in the ***Execution*** phase, the Planner component of the Adaptation Module outputs the adaptations to the Learning Material (if the Educator made any changes), and the Game Logic and Data, which in turn updates the Effectors to produce an impact on learning. Crucially, in our architecture, adaptations can be approved and customized for one student only or for an entire class. To sum up, the AI is adopted in this architecture to i) suggest adaptations automatically, and ii) refine adaptations through specific textual prompts inserted by the Educator.

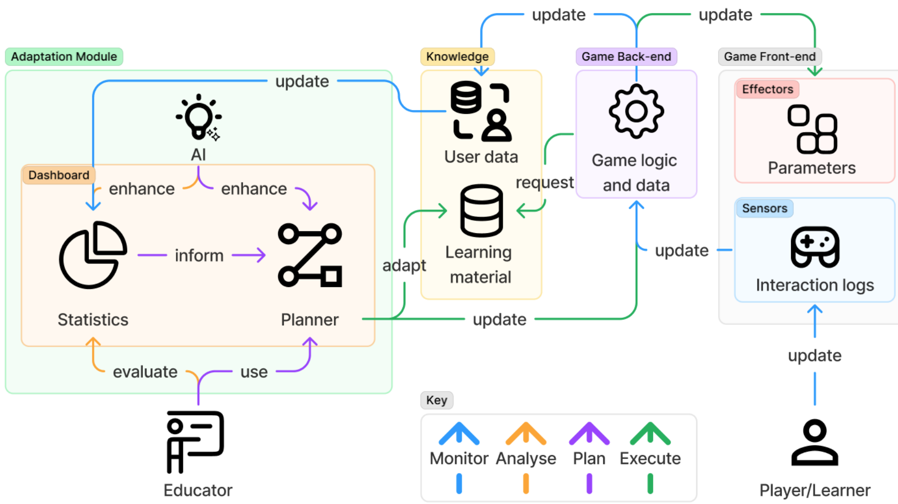


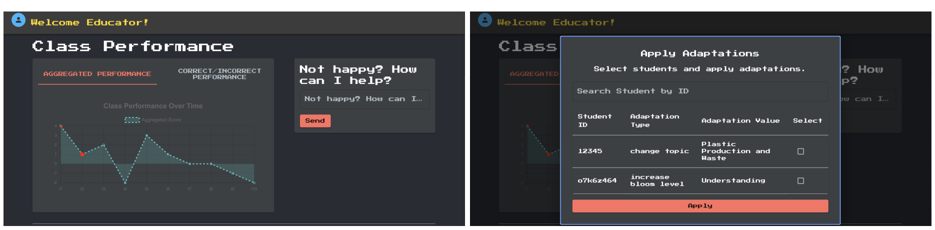
Fig. 1: Architecture design for LLMs in Adaptive SGs. Modules and flows are represented according to the MAPE-K loop (in particular, the Knowledge, Effector, and Sensor components, and the four phases).

3.1 Educators in the loop: dashboard design

The Dashboard is the Conversational User Interface (CUI) between the AI and the Educator. It is conceptualized as a Toolkit (see Figure 2) that supports the Educator in creating learning material and summarizing how the learning progresses within the game. It is built using React[3] and JavaScript for the front-end, and Node.js[4], Express[5], and MongoDB[6] for the back-end and data management. Generative AI is used to create educational materials, while Chart.js[7] provides data visualization for monitoring student performance. It is made up of two main components: the first is a user interface to generate educational materials using generative AI[8], simplifying the creation of content for educators; the second monitors the progress of the student's learning[9], providing aggregate information about overall class performance in learning activities (Figure 2a) and detailed insights into quiz accuracy. This also enables educators to decide when and how to adapt learning pathways for specific students. The AI generates personalized adaptations for each student based on their performance (Figure 2b). This dashboard includes two sub-modules, *Statistics module* and the *Planner*.

[3] https://react.dev/
[4] https://nodejs.org/
[5] https://expressjs.com/
[6] https://www.mongodb.com/
[7] https://www.chartjs.org/
[8] https://gala24demo-genmaterial-production.up.railway.app/
[9] https://gala24demo-ui-production.up.railway.app/

The **Statistics module** corresponds to the *Monitoring* and *Analysis* phases of the MAPE-K loop and offers insights about the class and Players/Learners. One is about the direction of the learning rate in a tab named ***Aggregated Performance*** (Figure 2a), which showcases the aggregated data from quizzes and the performance of the learner. This is the more heuristic part of the data visualization and it should help Educators grasp at first sight how the adaptation is faring. In the second tab, named ***Correct/Incorrect Performance*** or just *Performance*, it is possible to see the raw scores. The **Planner** corresponds to the Planning and Execution phases of the MAPE-K loop and is responsible for co-generating correct adaptations with the Educator. It is always up to the Educator to approve a specific adaptation strategy or to generate a new one. To carry out the necessary adaptations, the architecture is connected to a specific learning content generation API, based on GPT-3.5Turbo, which we detail in Section 3.2. Suggested adaptations are proposed to Educators in the form of buttons that, once clicked, send the necessary updates to the Game Back-end and then to the Front-end effectors. Additional adaptation suggestions can be generated by requesting new ones. If none are satisfying, Educators can always use an input field to prompt specific adaptations. It is worth noting that the Educator does not have to select adaptations every time a student fails. The Dashboard planner makes ideal adaptation suggestions (red point in the chart of Figure 2a and the Educator can choose which students to apply the adaptation to (Figure 2b).



(a) Aggregated Performance View. (b) Adaptation View.

Fig. 2: The Educator Toolkit.

3.2 Learning Content Generation APIs

Our approach uses a set of Application Programming Interfaces (APIs)[10] to enhance the teaching experience. These APIs, developed to leverage LLM capabilities, offer a comprehensive range of features that support Educators in any of their tasks. While they can be used individually, they are designed to work sequentially to streamline the educator's workflow within our architecture. This

[10] `https://skapi.polyglot-edu.com/swagger/index.html`

module can be used to generate the material beforehand. However, it also has the purpose of generating adapted material dynamically. For each learning material used in our system (i.e, PDF documents), we utilize the **Text Analyser API**, which extracts parameters like language, macro subject, title, educational level, and covered topics. It also suggests possible assessment categories based on the identified topics. The final step involves populating the learning activities. The **Activity Generator API** requires additional specifications such as activity type, assignment type, and parameters such as the number of correct answers and distractors. The generated activity includes the assignment, solutions, and necessary details, making it ready for our application. The generative AI APIs support a variety of educational activities and exercises. In particular, for the sake of simplicity, in our architecture, we focus on multiple-choice questions. This category allows for both single and multiple correct answers. These activities include assignments, distractors, and easily discardable distractors, challenging students to identify the correct answers among various options. To adapt learning content and define learning objectives and levels, the current version of our architecture employs Bloom's taxonomy [16], a widely recognized framework for classifying educational objectives into levels of complexity and specificity.

4 Applying our architecture: Untitled Bee Game

Untitled Bee Game (UBG) [6] is a serious game about eco-sustainability developed in Unity[11] that lets players control a bee in a small town (see Figure 3). It was inspired by the mechanics of Untitled Goose Game[12], where players control a goose and wreak havoc by disrupting human activities. The objective is to dissuade the NPCs from pursuing certain behaviours. The game implements two non-player characters (NPCs). The first one wastes plastic bottles, while the second one uses polluting substances to wash her car. If the behaviour is harmful, players are asked to sting the NPCs. If stinging is performed correctly, new flowers appear in the garden and new fish appear in the river. Once a flower has appeared, players can interact with it and read facts in the form of brief paragraphs. The game then quizzes players with multiple-choice questions about the materials.

4.1 Learning path and game loop: adapting UBG

Learning paths are defined as "*a sequence of activities with designated goals to help students build up their knowledge or skills in a subject area*" [30]. This sequence is composed of 'nodes' also known as *knowledge elements* (KEs) [30] or *learning objects* (LOs) [22]. To explore the feasibility of our approach, we start from a repository of learning materials, which we call *learning map*. We include

[11] `http://www.unity.com`
[12] `https://goose.game/`

Fig. 3: A multiple choice quiz in Untitled Bee Game.

a very simple learning path in Untitled Bee Game that can be adapted. The learning path is made of *learning nodes*, each of which contains several *activities* consisting of a reading node and an assessment node. We mapped the learning path onto the *game loop*, that is, the sequence of actions performed by the player in a typical play session. The game loop integrating the learning path unfolds as follows. Players have to complete a learning node, i.e., they read a fact by a flower and answer a quiz in a specific game location. Then, they perform game actions (mainly stinging humans), and finally, they get back to the learning location for the next learning node[13]. When all learning nodes are complete, the learning path is finished. When the first learning node is completed it becomes possible for educators to apply adaptations. There are two possible outcomes: i) if performance is below a threshold set by the Educator, the AI planner suggests making variations on the activity **topic** to pass the learning node; ii) if performance is above the threshold, the planner suggests a higher **Bloom level** and moves on to the next learning node in the path, which may be adapted or not. In the current implementation, adaptations are suggested according to thresholds. This will be replaced in the next version by LLM predictions from prompts including the student's performance history. In terms of MAPE-K components, the learning material, together with user performance history, is the *knowledge*; the content of learning nodes is the *effectors*; the quiz history is the *sensors*.

5 Conclusion and future work

In this work, we have presented an architecture for GAI-Educator collaboration in educational SGs where adaptation is driven by LLMs. The architecture

[13] A video is available at `https://youtu.be/wNMfLPgJhDU`.

is based on the MAPE-K loop, a well-known framework for self-adaptive systems. The adaptation is based on the Bloom taxonomy [5]. We show how the LLM-based approach and the MAPE-K loop can function in synergy to provide dynamic and robust adaptation, in compliance with Bloom taxonomy components, curated by a human in the loop. In addition, we have shown how the generated content is integrated into an existing educational serious game. The main limitation is that we have not yet evaluated the architecture in an experimental setting. In the future, we aim to adjust the granularity of the adaptation to smaller bits of the learning path, i.e. the individual activities that make up the learning nodes. We will also employ open-source LLMs that can be run locally (e.g., Gemma[14] instead of ChatGPT 3.5). Research in this area is still in its infancy. We expect this paper to be a starting point for researchers and practitioners willing to include LLMs in Education and SG design and execution.

References

1. An architectural blueprint for autonomic computing. Tech. rep., IBM (Jun 2005)
2. Abt, C.: Serious Games. Viking compass book, Viking Press (1970)
3. Bartle, R.: Hearts, clubs, diamonds, spades: Players who suit MUDs
4. Berger, F., Müller, W.: Back to basics: Explainable AI for adaptive serious games. In: Fletcher, B., Ma, M., Göbel, S., Baalsrud Hauge, J., Marsh, T. (eds.) Serious Games. pp. 67–81. Springer International Publishing (2021)
5. Bloom, B., Krathwohl, D.: Taxonomy of Educational Objectives: The Classification of Educational Goals. No. v. 1 in Taxonomy of Educational Objectives: The Classification of Educational Goals, Longmans, Green (1956)
6. Bonetti, F., Bassanelli, S., Bucchiarone, A., Gini, F., Marconi, A.: Untitled Bee Game: Be(e)ing Mean to Learn More About Eco-sustainability. In: Proceedings of the 8th Annual International GamiFIN Conference 2024. Ruka, Finland (2024)
7. Brandl, L., Schrader, A.: Student player types in higher education—trial and clustering analyses **14**
8. Böckle, M., Micheel, I., Bick, M., Novak, J.: A design framework for adaptive gamification applications. In: Proceedings of the 51st Hawaii International Conference on System Sciences (2018)
9. Clarisó, R., Cabot, J.: Model-driven prompt engineering. In: 2023 ACM/IEEE 26th International Conference on Model Driven Engineering Languages and Systems (MODELS). pp. 47–54. IEEE (2023)
10. Dobrovsky, A., Borghoff, U., Hofmann, M.: An approach to interactive deep reinforcement learning for serious games
11. Feuerriegel, S., Hartmann, J., Janiesch, C., Zschech, P.: Generative AI **66**(1), 111–126
12. Fotaris, P., Mastoras, T., Lameras, P.: Designing Educational Escape Rooms with Generative AI: A Framework and ChatGPT Prompt Engineering Guide. European Conference on Games Based Learning **17**(1), 180–189 (Sep 2023)
13. Frutos-Pascual, M., Zapirain, B.G.: Review of the Use of AI Techniques in Serious Games: Decision Making and Machine Learning. IEEE Transactions on Computational Intelligence and AI in Games **9**(2), 133–152 (Jun 2017)

[14] `https://huggingface.co/blog/gemma`

14. Hare, R., Tang, Y.: Player Modeling and Adaptation Methods Within Adaptive Serious Games. IEEE Transactions on Computational Social Systems **10**(4), 1939–1950 (Aug 2023)
15. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer **36**(1), 41–50 (2003)
16. Krathwohl, D.R.: A revision of bloom's taxonomy: An overview. Theory Into Practice **41**(4), 212–218 (2002)
17. Laamarti, F., Eid, M., El Saddik, A.: An overview of serious games **2014**, 1–15 (2014)
18. Lavoue, E., Monterrat, B., Desmarais, M., George, S.: Adaptive gamification for learning environments **12**(1), 16–28
19. Marczewski, A.: User types HEXAD. pp. 65–80
20. McIntosh, T.R., Liu, T., Susnjak, T., Watters, P., Ng, A., Halgamuge, M.N.: A culturally sensitive test to evaluate nuanced gpt hallucination. IEEE Transactions on Artificial Intelligence pp. 1–13 (2023)
21. Michael, D., Chen, S.: Serious Games: Games that Educate, Train and Inform. Thomson Course Technology
22. Nabizadeh, A.H., Leal, J.P., Rafsanjani, H.N., Shah, R.R.: Learning path personalization and recommendation methods: A survey of the state-of-the-art. Expert Systems with Applications **159**, 113596 (2020)
23. Oh, J., Raibulet, C., Leest, J.: Analysis of MAPE-K loop in self-adaptive systems for cloud, iot and CPS. In: Service-Oriented Computing - ICSOC 2022 Workshops - ASOCA, AI-PA, FMCIoT, WESOACS 2022, Sevilla, Spain, November 29 - December 2, 2022 Proceedings. Lecture Notes in Computer Science, vol. 13821, pp. 130–141. Springer (2022)
24. Orji, R., Tondello, G.F., Nacke, L.E.: Personalizing persuasive strategies in gameful systems to gamification user types. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. pp. 1–14. ACM (2018)
25. Pérez, J., Castro, M., López, G.: Serious Games and AI: Challenges and Opportunities for Computational Social Science. IEEE Access **11**, 62051–62061 (2023)
26. Souza, C.H.R., De Oliveira, S.S., Berretta, L.O., De Carvalho, S.T.: DDA-MAPEKit: A framework for dynamic difficulty adjustment based on MAPE-k loop. In: Proceedings of the 22nd Brazilian Symposium on Games and Digital Entertainment. pp. 1–10. ACM (2023)
27. Wang, T.: Navigating generative AI (ChatGPT) in higher education: Opportunities and challenges. In: Anutariya, C., Liu, D., Kinshuk, Tlili, A., Yang, J., Chang, M. (eds.) Smart Learning for A Sustainable Society. pp. 215–225. Springer Nature Singapore (2023)
28. Westera, W., Prada, R., Mascarenhas, S., Santos, P.A., Dias, J., Guimarães, M., Georgiadis, K., Nyamsuren, E., Bahreini, K., Yumak, Z., Christyowidiasmoro, C., Dascalu, M., Gutu-Robu, G., Ruseti, S.: Artificial intelligence moving serious gaming: Presenting reusable game AI components. Education and Information Technologies **25**(1), 351–380 (Jan 2020)
29. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C.: A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT (Feb 2023)
30. Yang, F., Li, F.W., Lau, R.W.: An open model for learning path construction. In: Advances in Web-Based Learning–ICWL 2010: 9th International Conference, Shanghai, China, December 8-10, 2010. Proceedings 9. pp. 318–328. Springer (2010)