A Conceptual Agent Based Framework for Personalized Interface Recommendation in Educational VR Games

Lada Maleš, Jelena Nakić, Marko Rosić

Abstract— **Choosing the optimal user interface is crucial for enhancing learning outcomes and user satisfaction in educational environments. This paper presents a conceptual agent-based framework designed to recommend more suitable interface between desktop and mobile, based on individual user profiles. The proposed architecture is designed for the educational virtual reality game ThinkLand and is based on the collected quantitative and qualitative user data collected through previously conducted usability study. The architecture utilizes a blackboard-based communication model to synchronize several specialized agents organized into three modules. This approach ensures selection of the interface that enables users to interact with educational content while keeping the focus on learning rather than on interface features with ultimate goal to increase learning outcomes.**

I. Introduction

The integration of digital games into pedagogical initiatives has been recognized as an effective strategy for enhancing student motivation, engagement and developing key skills, such as computational thinking (CT) [1]. Computational thinking is defined as a set of skills that includes decomposition, abstraction, pattern recognition and algorithm design [2], [3].

Virtual reality (VR) technology offers particular potential in this domain, enabling the visualization of complex and abstract concepts through interactive 3D environments [4]. However, the success of such solutions largely depends on usability of their interfaces, which is defined in terms of effectiveness, efficiency, and user satisfaction [5]. The usability of educational environments, particularly those based on advanced technologies such as virtual reality, directly affects learning outcomes [6][7]. When an interface is intuitive, clear, and easy to use, students can focus on the content and learning tasks rather than struggling with how to interact with the system itself. Although VR technology offers numerous advantages, such as immersiveness and interactivity, these features can also introduce additional challenges. For example, the use of VR in education can increase motivation, but at the same time, it may create challenges such as increased cognitive load if the interface is not well designed [8].

In this paper, we propose an innovative agent-based framework designed to suggest interface type i.e., desktop or mobile for an educational VR application called ThinkLand. This proposition uses a hybrid approach, integrating empirical data through a k-Nearest Neighbors (k-NN) algorithm and rule-based reasoning with the predictive modeling capabilities of neural networks. The architecture is structured into three functional modules. First module is responsible for collecting new user input data and creating recommendations and notifications based on empirical data from database of previous users. Second module aims to make final decision on the type of user interface which is more suitable for each user. Third module provides continuous system evolution.

This paper is structures as follows. Section II brings description of the ThinkLand game and briefly reports the results of previously conducted usability study. Section III describes a multiagent architecture for ThinkLand proposed in this paper. Section IV concludes the paper.

II. ThinkLand - Educational VR Game for Computational Thinking

ThinkLand is an educational game in virtual reality developed on the CoSpaces Edu platform [9]. The game is designed to encourage the development of computational thinking through the adaptation of tasks from the international Bebras Challenge [3] competition into a 3D environment. It consists of four specific tasks, each placed in a separate scene:

- Beaver Modulo is a simple introductory task where user clicks on a raccoon character that jumps across rocks arranged in a circle, simulating the modulo operation.
- Tower of Blocks is an interactive version of the task in which the user adds or removes blocks from towers to equalize their heights with the minimum number of moves. The game automatically counts the moves and offers a reset option.
- Tree Sudoku requires the user to manipulate the heights of trees in a 4×4 grid, with the goal of placing the trees so that each row and column contains exactly one tree of each height, with an additional condition of tree visibility from all the edges of the grid.
- Elevator presents an optimization task in which the user distributes penguins of different weights between two elevators with limited capacity, attempting to transport as many characters as possible at once.

The main advantage of ThinkLand compared to the original 2D tasks is the ability to actively manipulate objects, visually track changes, and receive immediate feedback, which encourages experimentation and reflection on problem-solving strategies. These differences, for example on the Tree Sudoku task, can be seen by comparing the original Bebras 2D task, shown in Fig. 1, with the 3D version of the same task in ThinkLand shown in Fig. 2.

*Research is supported by the project 'Hidden curriculum in higher education' (IP-UNIST-16) approved for funding under the University of Split's call for institutional research projects financed through the National Recovery and Resilience Plan 2021–2026.

Lada Maleš, University of Split, Faculty of Humanities and Social Sciences, Split, Croatia; (e-mail: lada.males@ffst.hr).

Jelena Nakić, University of Split, Faculty of Science, Split, Croatia; (e-mail: jelena.nakic@pmfst.hr).

Marko Rosić, University of Split, Faculty of Science, Split, Croatia; (e-mail: marko.rosic@pmfst.hr)

Figure 1. Tree Sudoku task in Bebras challenge [9]

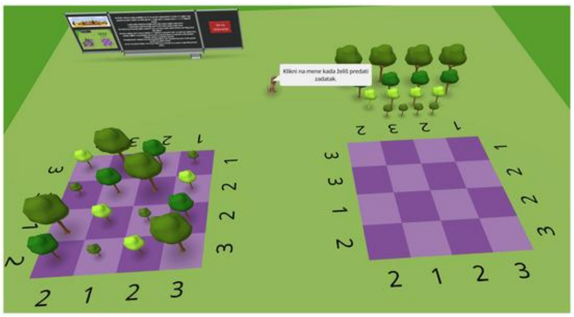

Figure 2. Tree Sudoku task in ThinLand [9]

A. Usability study of ThinkLand

The usability of the interface and user interaction with 3D objects in ThinkLand on different devices (desktop and mobile) were evaluated using a mixed-methods research approach [9]. The study involved 100 participants, including high school students and university students, who solved the tasks using two types of interfaces: mobile devices and desktop computers. User perception was assessed using the standardized System Usability Scale (SUS) [10] and a specially developed custom questionnaire, named Virtual Environment Usability Questionnaire (VEUQ).

The primary objectives of the research were as follows: i) to determine the impact of device type on task performance efficiency and user experience, ii) to identify specific interaction barriers within the virtual space on each interface iii) to analyze the relationship between demographic characteristics (age and gender) and the perceived usability.

The empirical findings demonstrate that the ThinkLand reached a 'good' overall usability threshold and there was no statistically significant difference in SUS scores between desktop (72.73) and mobile users (73.17). However, interaction analysis has showed statistically significant differences related to interface type. Desktop users found selecting and moving objects significantly easier than mobile users. The primary obstacles for mobile interface users were camera rotation and precise object selection, while desktop users experienced difficulties with more complex functions such as zooming. The time to complete the game correlated with participants' age, i.e., university students required more time to complete the challenges than high school students. Mobile users reported on screen feedback information as subjectively more useful compared to the desktop users.

III. MULTIAGENT ARCHITECTURE FOR THINKLAND

An agent can be any entity which is autonomous, can act reactively and proactively and possesses the property of social ability that enables interaction with other agents [11]. An multiagent system (MAS) architecture presented in this paper supports data interpretation, manages predictions, and makes decision on suitable interface for educational VR game ThinkLand. The MAS approach was chosen due to its ability to distribute responsibilities across specialized agents, each performing a different type of task. This architecture enables a clear separation between sub-symbolic artificial intelligence, represented by the neural network component, and symbolic artificial intelligence, which governs rule-based decision logic, each encapsulated within an independent, specialized agent integrated into the broader multi-agent system.

The MAS architecture for ThinkLand consists of three modules: Analysis Module, Reasoning & Decision Module and Learning Module. Communication takes place through a shared blackboard memory. All agents can write data on blackboard and read data from blackboard. The ThinkLand MAS architecture is shown in Fig. 3.

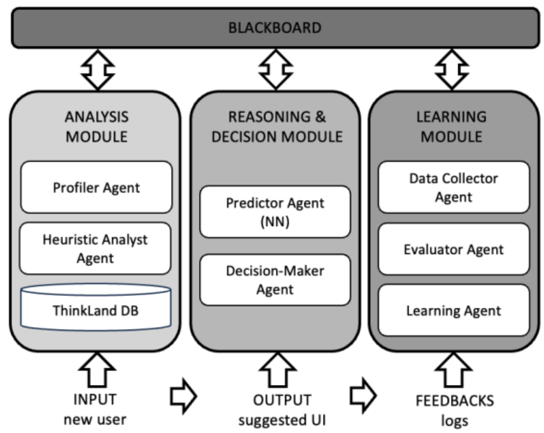

Figure 3. ThinkLand MAS architecture

A. Analysis module (Module 1)

The primary function of this module is to collect input data and establish the context for interface recommendation, i.e. it is responsible for contextual analytics. It receives input data from new users and contains data from 100 participants previously gathered from results of ThinkLand game. The Module 1 consists of two agents Profiler Agent, Heuristic Analyst Agent and database, i.e., ThinkLand database.

The task of the Profiler Agent is to collect the input data (age and gender) of each new user, assigns an identification number and write it to the blackboard.

The Heuristic Analyst Agent has access to a ThinkLand database that contains data from previous users who have played the game. The data includes age, gender, the type of used, game results i.e., the total score achieved in the game (referred to as final score) and the overall time required to complete the game (referred to as time), a completed VEUQ questionnaire, and SUS score. This agent implements a hybrid approach that combines Case-Based Reasoning (CBR) for identifying similar user profiles and rule-based logic for validating recommendations. By using standardized thresholds from the SUS and performance measures (final score and time), the agent ensures that the system does not generate recommendations based solely on predictive statistics, but on proven empirical data from the ThinkLand database. Its goal is to explain why a decision was made. The Heuristic Analyst

Agent serves as a knowledge-based expert that bridges the gap between data form the initial study and real-time decision-making.

The operation process of the Heuristic Analyst Agent begins with the application of CBR. To identify the most relevant reference profiles, the agent employs the k-Nearest Neighbours (*k*-NN) algorithm that search for the *k* most similar previous users to the current user. To mathematically determine the similarity between a new user U_{new} and user from database U_{db}, the agent calculates the Euclidian distance *d* using the equation Eq. (1):

$$d\,(U_{new}, U_{db}) = \sqrt{\sum_{i=1}^{n}(x_{i,new} - x_{i,db})^2} \tag{1}$$

where x_1 is age (14-22), x_2 is gender (binary coded: 0 for male and 1 for female, in initial study no one selected “prefer not to say” option) and n is a number of users in ThinkLand database. Therefore, the equation that agent computes is Eq. (2):

$$d\,(U_{new}, U_{db}) = \sqrt{\sum_{i=1}^{n}(age_{i,new} - age_{i,db})^2 + (gender_{i,new} - gender_{i,db})^2} \tag{2}$$

Since age and gender have different numerical scales, the agent applies Min-Max normalization to ensure that both features contribute equally to the distance calculation. Then agent selects the k-NN (e.g., k = 5) and retrieves from ThinkLand database SUS_scores, final score and time associated with these neighbors and computes average values. If the majority of these neighbors' results have low usability or long game duration on specific interface the agent writes recommendations and notifications on the blackboard to the Decision-Maker Agent. The next step is the application of rule-based logic, e.g., IF-THEN rules. For example, *IF avg_SUS_score < 68 THEN set_notification = RISK_HIGH* or *IF avg_time > 2 * global_median_time THEN set_recommendation = Desktop*. Moreover, agent calculates two interface suitability indices, one for desktop interface $I_{Desktop}$ and other for mobile interface I_{Mobile} using the equation Eq. (3):

$$I = w_1 \cdot avg\_result + w_2 \cdot \frac{1}{avg\_time} + w_3 \cdot avg\_SUS\_score \tag{3}$$

where w_1, w_2 and w_3 are weight coefficients, *avg_result*, *avg_time* and *avg_SUS_score* represent average values for k-nearest neighbors. The system enables flexible configuration of the weight coefficients according to their specific pedagogical goals.

B. *Reasoning and Decision-Making Module (Module 2)*

Reasoning and Decision-Making Module (Module 2) is responsible for reasoning and decision-making. This module is the central part of the architecture, which uses artificial intelligence to generate personalized decisions. To achieve this, it is necessary to include two agents Predictor Agent and Decision-Maker Agent.

Predictor Agent contains a neural network trained on data collected after 100 participants have used ThinkLand. Based on the age and gender that Profiler Agent has written on the blackboad, the Predictor Agent predicts SUS scores for each interface (desktop and mobile).

The Decision-Maker Agent reads the predictions of the Predictor Agent, the suitability indices of the Heuristic Analyst Agent, heuristic recommendation and notifications and apply predefined IF-THEN rules to create final decision on which type of interface will be better to a new user. For example, rule can be IF set_notification = RISK_HIGH THEN Recommend Desktop. This approach enables to prioritize recommendations and notifications created by the Heuristic Analyst Agent over Predictor Agent. That enables system to reduce potential AI hallucinations by favoring the experience of the actual users and tune desirable pedagogical outcomes.

C. *Learning module (Module 3)*

This module enables the MAS to learn from the data of each new user after they complete the game and fill out the questionnaires. There are three agents in this module: Data Collector Agent, Evaluator Agent and Learning Agent.

The Data Collector Agent reads game data from logs and writes results of played ThinkLand game, i.e., final score and time, on blackboard.

The Evaluator Agent collects responses from the SUS filled in after the game, calculates the SUS score, and compares it with the assessment made by the Predictor Agent. All the necessary data for the evaluation are available on the blackboard, and the Evaluator Agent writes the new user SUS score on the blackboard after the evaluation.

The Learning Agent compares what the system predicted with what actually happened. If the difference is too large (threshold must be set up), the new user data are saved in the ThinkLand database, and a signal is sent to the neural network indicating that retraining is needed.

D. *Execution cycle of the ThinkLand MAS*

The execution cycle carried out by the MAS consists of four phases.

- Phase 1 – **Initialization and Contextualization**. In this phase, the user inputs data (age and gender), the Profiler Agent takes this data, assigns an ID number, and write all on the blackboard. Then Heuristic Analyst Agent searches the ThinkLand database and writes recommendations and notifications on the blackboard, which can also be interpreted as risk warnings, for example: “this profile often has low SUS score on mobile interface” or “this profile has completion time twice longer than average”.
- Phase 2 – **Reasoning and Decision-Making**. In this phase, based on all available data on the blackboard and using the neural network, the SUS score is predicted for both interfaces (desktop and mobile). The predicted SUS scores and suitability indices from Module 1 are compared and rules are applied to reach final decision. The rules are designed to prioritize heuristic decision rather than predicted decision, if the results are different. The system is designed so that the rules can be written in different ways to accommodate different logic models.
- Phase 3 - **Execution and log collection**. In this phase, the user plays the game on the interface suggested by the system. During the game, logs are generated (score and time for each task in the game, but only the final score and time are used for the prediction). After the user finishes the game, the SUS and VEUQ questionnaires need to be completed.
- Phase 4 - **Evaluation and learning**. In this phase, data from the logs are already written on the blackboard, the SUS score is calculated and also recorded on the

blackboard. At the end of this phase, the system prediction obtained by the neural network output is compared with the actual results. If there is a difference, the Learning Agent updates the ThinkLand database with the user data and starts retraining of the neural network.

Table I. shows the overview of agent interactions and communication through blackboard.

TABLE I. OVERVIEW OF AGENT INTERACTIONS

Agent	*Function*	*Action*	*Blackboard*
Profiler	collects user demographics	writes	age, gender, user ID
Heuristic Analyst	searches the N=100 ThinLand database for similar profiles	reads	age, gender
		writes	suitability index for interfaces, heuristic context (recommendations and notifications), time and final score
Predictor (NN)	AI prediction of usability	reads	age, gender
		writes	predicted SUS score for interfaces
Decision Maker	makes the final interface decision	reads	predicted SUS, suitability index for interfaces, heuristic context
		writes	final decision on UI (desktop/mobile)
Data Collector	processes performance logs (raw data)	writes	final score and time
Evaluator	calculates SUS score after completed game	writes	actual session SUS score
Learning Agent	compares predicted and real results from the game	read	all data from blackboard
	updates the ThinkLand database and starts NN retraining if necessary	write	info about new entries to ThinkLand and retraining the NN

IV. CONCLUSION

This paper presents a comprehensive framework for developing the MAS aimed at providing personalized decisions on interface type within an educational context. By designing MAS architecture for integration into the ThinkLand educational game, it is demonstrated that the user experience can be dynamically optimized through a hybrid intelligence approach.

The primary strength of the system lies in its dual-layered reasoning. The application of the k-NN algorithm, based on Euclidean distance metrics, enables the Heuristic Analyst Agent to utilize empirical data from a ThinkLand database to calculate a personalized interface suitability index. In combination with predictive capabilities of the neural network, which estimates SUS score, the system achieves a critical balance between advanced AI prediction and evidence based on previous user data. The final decision is determined by the Decision Maker Agent, which acts as the system central reasoning engine. By employing rule-based logic, this agent synthesizes the outputs from both the Heuristic Analyst Agent and the Predictor Agent. A crucial aspect of its operation is the integration of heuristic recommendations and notifications; if data indicates user discomfort or task failure, the Decision-Maker Agent will override the neural network prediction.

A key feature of this framework is its adaptability. By allowing game creators to modify the weighting factors *w*, the system can be tailored to various pedagogical goals, i.e., whether the focus is on maximizing task performance (Bebras scores or time) or ensuring high subjective usability (SUS scores). Furthermore, the inclusion of a Learning Agent ensures that the system is not static. As more users engage with the game, their performance and feedback are autonomously integrated back into the architecture, continuously refining the accuracy of future decisions on interface type.

The reliance on age and gender as primary inputs for new users addresses the initial cold start problem. These variables represent fundamental, universally available characteristics that provide an immediate entry point without requiring intrusive pre-testing or long psychometric evaluations that could disrupt the initial immersion into the game. Furthermore, this choice is empirically grounded in our previous usability testing dataset. The initial demographic profile serves merely as a temporary starting point. As soon as a user engages with the ThinkLand game, the Learning Agent continuously captures real-time performance data, such as Bebras scores, and subjective feedback via SUS scores. This mechanism ensures that the system rapidly shifts from demographic classification to individualized, performance-based personalization.

A current limitation of this framework is the exclusion of higher-order individual differences, such as learning styles, prior knowledge, or cognitive styles, which are well-known to significantly influence the quality and depth of educational personalization [12]. In future work, these psychological and pedagogical dimensions will be integrated into the MAS for ThinkLand educational game. Specifically, established adaptive e-learning methodologies, such as the user-modeling approaches proposed in [12], will be applied to enable the Learning Agent to dynamically deduce cognitive and learning styles from implicit in-game behavior and navigation patterns rather than static questionnaires. This extension is expected to improve the accuracy of the suitability index and support a more holistic approach to real-time interface and content adaptation.

This paper establishes a detailed theoretically grounded framework for a MAS and elaborates the conceptual model which is ready for technical execution. Future research will focus on the full-scale implementation of this MAS within the ThinkLand environment, including the development and training of the Multilayer Perceptron (MLP) as part of the Predictor Agent, fine-tuned through k-fold cross-validation on the existing dataset of 100 participants. Subsequent phases will involve empirical testing with new users to validate the accuracy of the Decision Maker logic and the effectiveness of the calculated suitability index in real-time scenarios. These tests will provide the necessary data to refine the agent functions and further enhance the personalization of adaptive learning experiences.

REFERENCES

[1] S. W. Gundersen and G. Lampropoulos, "Using Serious Games and Digital Games to Improve Students' Computational Thinking and Programming

Skills in K-12 Education: A Systematic Literature Review," *Technologies*, vol. 13, p. 113, 2025.

[2] J. M. Wing, "Computational Thinking," *Communications of the ACM*, vol. 49, pp. 33–35, 2006.

[3] Bebras International. Available: https://www.bebras.org/ (accessed March 2, 2026).

[4] A. S. Oyelere, F. J. Agbo, and S. S. Oyelere, "Formative Evaluation of Immersive Virtual Reality Expedition Mini-Games to Facilitate Computational Thinking," *Comput. Educ. X Real.*, vol. 2, p. 100016, 2023.

[5] ISO 9241-11:2018(En), Ergonomics of Human-System Interaction—Part 11: Usability: Definitions and Concepts, ISO, 2018.

[6] A. Granić, J. Nakić, and N. Marangunić, "Scenario-Based Group Usability Testing as a Mixed Methods Approach to the Evaluation of Three-Dimensional Virtual Learning Environments," *J. Educ. Comput. Res.*, vol. 58, pp. 616–639, 2020.

[7] S. Sukirman, L. F. M. Ibharim, C. S. Said, and B. Murtiyasa, "Development and Usability Testing of a Virtual Reality Game for Learning Computational Thinking," *Int. J. Serious Games*, vol. 11, pp. 19–43, 2024.

[8] F. J. Agbo, S. S. Oyelere, J. Suhonen, and M. Tukiainen, "Design, Development, and Evaluation of a Virtual Reality Game-Based Application to Support Computational Thinking," *Educ. Technol. Res. Dev.*, vol. 71, pp. 505–537, 2023.

[9] J. Nakić, I. Rogulj, and L. Maleš, "Rethinking the Bebras Challenge in Virtual Reality: Implementation and Usability Study of a Computational Thinking Game," *Multimodal Technologies and Interaction*, vol. 9, no. 6, p. 60, 2025. Available: https://doi.org/10.3390/mti9060060.

[10] J. Brooke, "SUS: A 'Quick and Dirty' Usability Scale," in *Usability Evaluation in Industry*, London, U.K.: CRC Press, 1996.

[11] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995. Available: http://dx.doi.org/10.1017/S0269888900008122.

[12] J. Nakić, A. Granić, and V. Glavinić, "Anatomy of student models in adaptive learning systems: A systematic literature review of individual differences from 2001 to 2013," *J. Educ. Comput. Res.*, vol. 51, no. 4, pp. 459–489, 2015. Available: https://doi.org/10.2190/EC.51.4.e