2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)

2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET) | 979-8-3503-2259-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSE-SEET58685.2023.00020

BARA: A Dynamic State-based Serious Game for Teaching Requirements Elicitation

Yu Liu
Beijing University of Technology
Beijing, China
vmasks@outlook.com

Tong Li
Beijing University of Technology
Beijing, China
litong@bjut.edu.cn

Zheqing Huang
Beijing University of Technology
Beijing, China
zheqing.huang@ucdconnect.ie

Zhen Yang
Beijing University of Technology
Beijing, China
yangzhen@bjut.edu.cn

***Abstract*—Teaching requirements elicitation to students who do not have practical experience is challenging, as they usually cannot understand the difficulty. Several recent studies have reported their experience of teaching requirements elicitation with a serious game. However, in these games, the fictitious characters have not been carefully designed to reflect real scenarios. For example, they always respond the same no matter how many times a learner interacts with them. Moreover, most existing serious games contain only one specific scenario and cannot be easily extended to cover various cases. In this paper, we design and implement a dynamic state-based serious game (BARA) for teaching requirements elicitation, which can realistically simulate real-world scenarios and automatically record learners' actions for assessment. Specifically, we model fictitious characters' behaviors using finite-state machines in order to precisely characterize the dynamic states of stakeholders. We also developed an easy-to-use editor for non-programmers to design fictitious characters and thus construct various simulated scenarios. Finally, BARA records learners' actions during the game, based on which we can gain an in-depth understanding of learners' performance and our teaching effectiveness. We evaluated BARA with 60 participants using a simulated scenario. The result shows that most participants are immersed in BARA and can reasonably complete the requirements elicitation task within the simulated scenario.**

***Index Terms*—requirements elicitation, serious game, empirical study, goal-based design**

I. Introduction

Requirements elicitation, which is the first requirements engineering (RE) step, plays an important and fundamental role in RE [1]. Unfortunately, many learners (e.g., college students) cannot recognize the difficulties in requirements elicitation due to the lack of practical experience, and thus do not pay enough attention to learn it. Ideally, we want learners to participate in the requirements elicitation work, interact with stakeholders, and thus can identify the real challenges in requirements elicitation.

Because realistic projects and stakeholders are usually not easy to find, especially in university studies , serious games [2] have been identified as an alternative educational approach [3]. In RE education, several researchers have proposed to construct simulated requirements scenarios in a serious game [4]–[6]. The scenario's background is that an organization needs to develop a system to complete certain business objectives. There are multiple fictitious stakeholders in the scenario. A learner will play the role of a requirements analyst, interact with stakeholders, and elicit their requirements for the new system. By carefully designing characters' behavior and dialogue content, learners can conduct experiential learning and accumulate requirements elicitation experience.

In recent years, several works have proposed solutions for teaching requirements elicitation with a serious game [4]–[7]. Rusu et al. [7] constructed an earth defense scenario. Learners are required to conduct several decision-based conversations with fictitious characters within a limited number of times to elicit their requirements for the Earth's defense system. Garcia et al. [5], [6] respectively construct library scenarios with multiple fictitious characters and require learners to elicit and model the requirements of these characters. [4] is similar to the previous two works but provides several increasingly difficult scenarios.

However, the aforementioned studies have not fully unleashed the power of serious games in the following three aspects. Firstly, in their scenarios, the states of fictitious characters can not change, which would result in a fixed interaction. In other words, all the characters will always give the same response no matter how many times the learner interacts with them. This does not match reality and cannot effectively help learners to understand and experience real requirements elicitation scenarios. Instead, we argue that the dynamic states of fictitious characters should be designed and implemented to better reflect the real scenarios. For example, if a learner interacts with a character repeatedly, the character will become impatient and thus not respond to the learner. Secondly, most existing works only provide one particular scenario, which is far from enough regarding the teaching objective. In different scenarios, the requirements elicitation experience may vary greatly. Therefore, providing more high-quality scenarios can achieve better teaching effectiveness.

979-8-3503-2259-0/23/$31.00 ©2023 IEEE
DOI 10.1109/ICSE-SEET58685.2023.00020

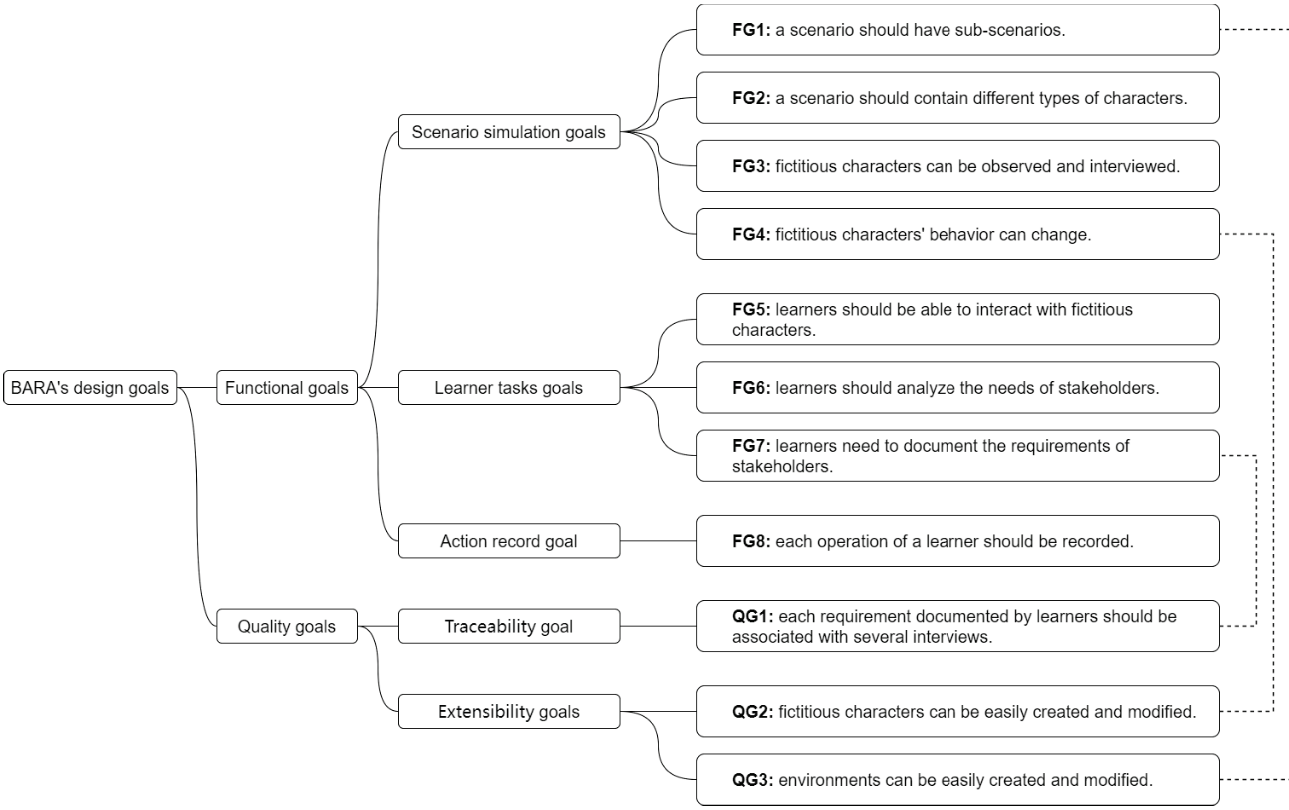

Fig. 1. The goal-based design model

Taking a pragmatic perspective, it is more important to address the challenge that who and how to produce such scenarios. We believe that crowdsourcing is an ideal solution to practically support non-programmers to construct scenarios. Finally, none of the existing studies deeply investigate learners' learning process, i.e., their interactions with the game, which we think can provide educators with valuable insight into their teaching effectiveness and support a more precise evaluation of learners' performance.

In this paper, we designed and developed a dynamic state-based serious game to teach requirements elicitation, called BARA (**B**e **A** **R**equirements **A**nalyst). In BARA, the behaviors of fictitious characters are modeled by finite-state machines and can change according to certain conditions (i.e., learners' interaction with fictitious characters). In different states, the response of characters will be different, which can precisely reflect the reality and thus promote teaching effectiveness. BARA also supports multiple scenarios and can add new scenarios by loading constructed scenarios. To facilitate the process of scenario construction, we also developed an easy-to-use scenario editor for creating and modifying fictitious characters by non-programmers. In addition, BARA can record learners' actions during the game with a timestamp. With these records, we can know how learners get their results and assess their learning effectiveness. In summary, we make the following contributions:

- We designed and developed a dynamic state-based serious game called BARA, which can realistically simulate real-world scenarios.
- We developed an easy-to-use scenario editor to facilitate the process of scenario construction so that non-programmers to construct scenarios.
- We record learners' actions during the game for a more precise evaluation of learners' performance.
- Our experiment shows that of 60 participants, more than 80% of participants can be immersed in BARA and reasonably complete the requirements elicitation task in the simulated scenario.

The rest of the paper is structured as follows: section 2 proposes a goal-based design model. Section 3 describes design details according to the goal model. In section 4, we construct a scenario in BARA and introduce details about the scenario. In section 5, we required 60 undergraduate students from a software engineering course to use BARA, then used collected data to evaluate BARA. Section 6 discusses related works. Section 7 provides discussion and future works. Finally, we conclude in section 8.

II. Design goals

The main goal we want to achieve is using a game to enable learners to learn to elicit and analyze the needs of fictitious stakeholders in simulated scenarios and document their needs as requirements with structured patterns. In order to reach the main goal, we decompose it as several sub-goals. The goal-based design model is shown in the Fig. 1.

We decompose the main goal into functional goals and quality goals. The functional goals point out the capabilities that BARA should have, while the quality goals put quality requirements on corresponding functional goals. In the Fig. 1, the quality goals and the corresponding functional goals are connected by dashed lines. In the rest of this section, we will elaborate on three functional goals and their corresponding quality goals.

A. Scenario simulation goals

In RE education, serious games can be an alternative way when authentic projects or stakeholders are not available [3]. We argue that it is practical only if the game can simulate reality, including the simulation of the environments (FG1, FG2) and the simulation of fictitious characters (FG3, FG4).

1) Environments simulation goals: (FG1) we believe that a scenario should have multiple sub-scenarios. On the one hand, this design simulates reality, and on the other hand, it can increase the complexity of the scenario, thereby stimulating the learners' desire to explore.

QG3 is the quality goal corresponding to FG1. BARA aims to teach RE by simulating realistic scenarios. We believe a learner can perform better after experiencing similar simulated scenarios in BARA. However, in reality, affected by various factors such as types of companies and types of stakeholders, there will be different experiences in the process of requirements engineering. In order to contain more scenarios, BARA first needs support loading scenarios from external files. Moreover, we need experienced requirements analysts to assist in creating more high-quality scenarios. Thus a low-cost creating and modifying environments approach is essential.

(FG2) In simulated scenarios, there should be multiple types of fictitious characters, including different stakeholders(e.g., users and managers) and unrelated characters. The learner needs to interact with them and identify whether they are stakeholders because only stakeholders' needs should be considered. Such a design can allow learners to realize that realistic scenarios are complex, and in addition to stakeholders, there will also be irrelevant persons in realistic scenarios.

2) Fictitious characters simulation goals: (FG3) there should be two interaction ways between a learner and fictitious characters, including observation and interview. These two interactions are the same as in reality. The interview is a direct conversation with a fictitious character, while the observation is a way to get information without disturbing a fictitious character. Through different interaction ways, the learner may get different information, which simulates reality and improves the interest of BARA.

(FG4) In addition, behaviors of fictitious characters should change over time or be affected by the learner's interaction. The learner can get different information when a character is in different states and need to decide what to do next according to the character's feedback. On the one hand, such a design simulates reality; on the other hand, it allows learners to gather more information from fictitious characters.

QG2 is the quality goal corresponding to FG4. A scenario consists of an environment and multiple fictitious characters. So for fictitious characters, we also need an easy way to create and modify fictitious characters.

B. Learner tasks goals

The learner tasks goals indicate the tasks that a learner must complete in a game session. Specifically, in BARA, a learner plays the role of a requirements analyst and is immersed in simulated scenarios consisting of environments and fictitious characters. (FG5) The learner can be familiar with scenarios by strolling in the environments and interacting with fictitious characters. (FG6) Next, the learner should comprehensively analyze the collected information, identify the stakeholders' needs, and discard unreasonable needs. (FG7) Finally, the learner should document the needs of stakeholders as requirements with structured patterns.

QG1 is the quality goal corresponding to FG7. In addition to being interested in the content of a requirement, we are also interested in which conversations the learner recalled when he documented this requirement. This information is helpful to the teaching effectiveness evaluation. So BARA should have the ability to trace this corresponding relationship.

C. Action record goals

(FG8) For a more precise evaluation of learners' performance, during a game session, learners' actions will be automatically recorded by BARA, such as "interview with a character" and "document a requirement for a character", and each record has a timestamp.

III. Design of BARA

In this section, we will give design details of BARA according to our proposed design goals, including scenario simulation design, learner tasks design, and action record design. The design for quality goals will follow the corresponding functional design.

A. Scenario simulation design

1) Scenario design (FG1, FG2): in our design, a scenario should have at least three sub-scenarios. Moreover, there should be multiple types of fictitious characters, including stakeholders and unrelated persons, distributed in scenarios. Stakeholders can provide helpful information, while unrelated persons can confuse learners with conflicting or useless information. To determine whether a character is a stakeholder or an unrelated person, a learner needs to interact with him and make a comprehensive analysis based on the scenario's background and the contents of the dialogue.

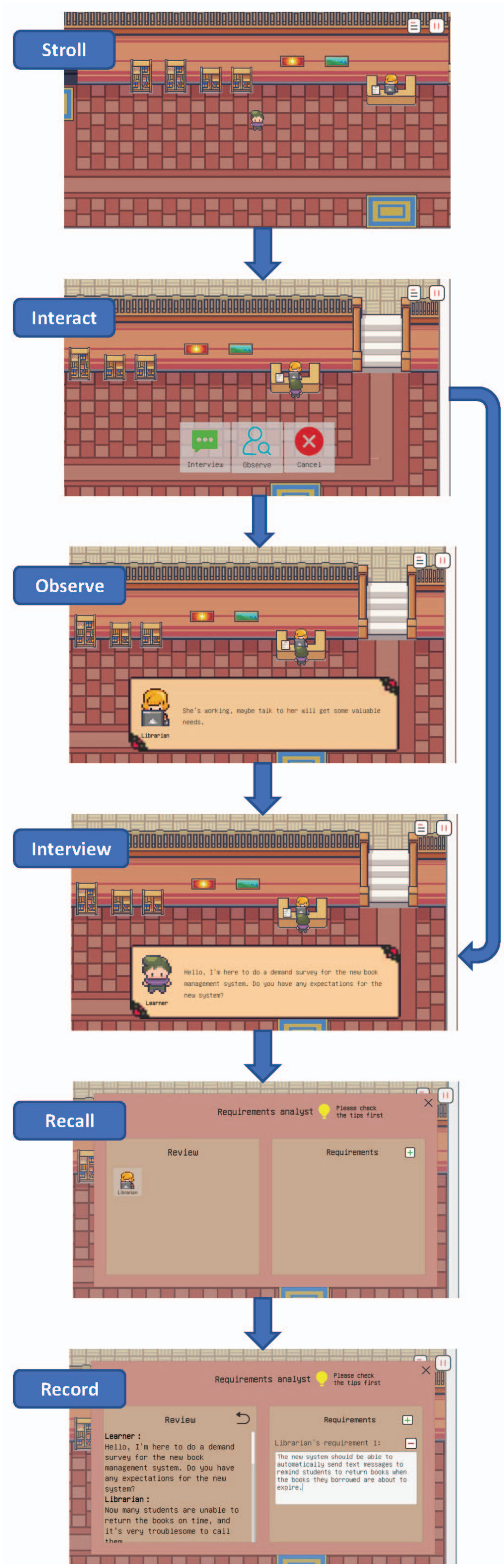

Fig. 2. The requirements elicitation process in BARA

2) Scenario extensibility design (QG3): for environments that can be easily built, we use a 2D tile map to create the environments. A tile map is a map in which each region is represented by a single tile of the same shape and size. Given appropriate tile materials, one can easily build a tile map using the map edit tool (e.g., Tiled[1]).

3) Fictitious characters interaction design (FG3): in BARA, a fictitious character can be observed and interviewed by the learner. The interview is a technique for identifying the needs of users and other stakeholders of the system through face-to-face conversation. The operation of the learner interviewing a character will trigger a pre-edited dialogue text. A pre-edited dialogue is a multi-line text. Each line of content represents either a learner or a fictitious character and is shown in turn, simulating the natural interview process. The Fig. 2 "Interview" part shows the interview operation. The analyst is asking about the character's needs for the new system. Through giving interviews with a fictitious character, the learner can collect his needs and identify whether he is a stakeholder or an unrelated person to the project.

We also simulated the observation elicitation technique in BARA. The observation is a technique that involves an investigation of the user's work and taking notes on the events. The learner observes a character that will trigger a pre-edited text, which describes what the fictitious character is currently doing. Compared to the interview operation, the observation operation may contain more content. By observing a character, the learner can not only know what he is doing to get helpful information about the project but also know his current state to decide whether to interview him further. For instance, when the learner observes a character and knows he is focusing on work, the learner should not further interview him until he has finished his work. Otherwise, the character may get angry, resulting in the learner can not get information from him in this game session. The Fig. 2 "Observe" part shows the observe operation. The analyst observes that the character is working. The change of characters' behavior is implemented by finite-state machines, which we will describe in the next paragraphs.

4) Fictitious characters behaviors design (FG4): in order to better simulate realistic scenarios, we use FSM (finite-state machine) to model fictitious characters' states. A finite-state machine is an abstract machine that can be in exactly one of a finite number of states at any given time. The FSM can change from one state to another in response to some inputs.

In BARA, each fictitious characters have an FSM to manage their respective states, and each state corresponds to a pre-edited dialogue text and a pre-edited observation text. Besides the "be observed" and the "be interviewed" operation are the inputs of an FSM, "after some time" is also an input, and the time can be specified. When the learner observes or interviews a character, the dialog box will show corresponding content. After that, The FSM will transfer to the next state based on the current state and input. The machine will keep the current state if there is no corresponding next state. Fig. 3 shows a

[1]https://www.mapeditor.org/

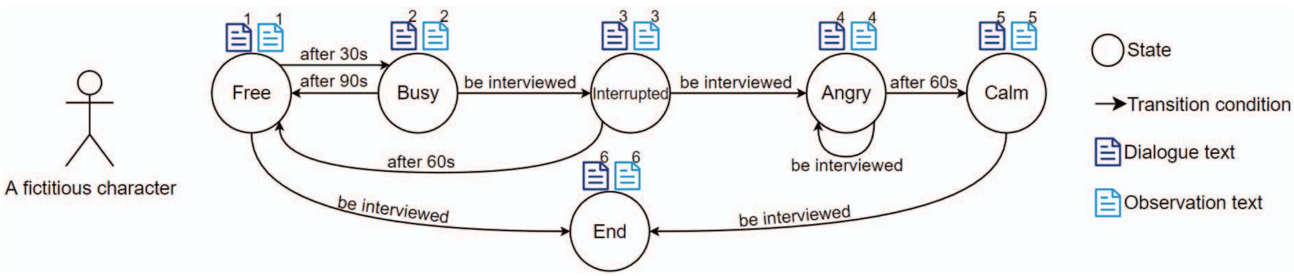

Fig. 3. A finite-state machine example

character's FSM instance. There are six states in the FSM, each with an independent dialogue text and an observation text. The character's default state is "Free" and will transfer between "Free" and "Busy" automatically after a specific time. If the learner interviews him at the "Free" state, he will tell the learner his needs and then transfer to the "Finished" state. If the learner interviews him at the "Busy" state, he will tell the learner he is busy with his own business and comes back to him when he is not busy for a while. The character's state is now "Interrupted". If the learner listens to him and interviews him when his state transfers to the "Free" after 60 seconds, the learner can also get the character's needs. However, if the learner does not listen to him and still interviews him, the character will be angry. When the character is in the "Angry" state, the learner can not get any helpful information from him. The character's state will repeatedly enter the "Angry" state if the learner repeatedly interviews him. The character's state will transfer to "Calm" after 60 seconds. At this state, the learner interviews him. He will also tell the learner his needs. Whether the character is in the "Free" or "Calm" state, the state will transfer to the "Finished" after the learner interview with him. In the "Finished" state, whether the learner observes or interviews him, his state will keep in this state, and the character will tell the learner that he has told the learner his needs and ask about others' needs regarding the new system.

5) Fictitious characters extensibility design (QG2): To facilitate the fictitious characters' design process, we developed a tool for editing fictitious characters' states and corresponding dialogue and observation text. The user interface of the tool is shown in the Fig. 4. It can be seen that the UI is horizontally dividend into three parts. The characters management panel (leftmost) manages all characters in the scenario. We can create a new character by clicking the "plus" button below or delete an existing character by clicking the "minus" button after the corresponding character's name. Once a character is selected, we can edit his states diagram in the states editing panel (rightmost) by dragging the state node or transition condition. Then select a state, and we can edit the dialogue text and observation text in the text editing panel (middle). Finally, we can export this scenario as files when all fictitious characters have been created. BARA can read these files and generate fictitious characters in scenarios.

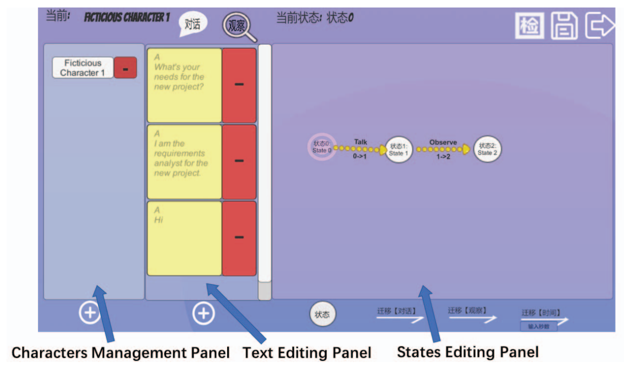

Fig. 4. The fictitious characters editor

B. Learner tasks design

The genre of BARA is the same as most 2D Role-playing games. There are simulated scenarios with multiple fictitious characters with the background of a company that will develop a software system to accomplish a business. A learner plays the role of a requirements analyst and is responsible for the requirements elicitation for this system. Specifically, throughout a game session, the learner needs to accomplish the following tasks:

- Explore scenarios to know the layout of the scenarios, the types of fictitious characters, and their locations.
- Observe and interview fictitious characters to collect their needs. (FG5)
- Comprehensively analyze based on collected information and identify whether the requirements are reasonable or conflicting. (FG6)
- Document the requirements with structured patterns. (FG7)

The primary interface and game flow of BARA can be seen in the Fig. 2. At a game session beginning, a learner plays the role of a requirements analyst and appears at a point in the scenario. The learner can control the analyst stroll in the scenario. During the exploration, the learner can discover some fictitious characters. When the learner approaches a character and presses the "Enter" key, a floating panel pops up, and the learner can select to observe or interview the fictitious character.

Once an interview is completed, the conversation content will be recorded automatically and can be recalled in the “requirements panel” at any time. The “requirements panel” can be seen in the Fig. 2 “Recall” part. It is divided into left and right two parts. The left part lists all fictitious characters interviewed, and the right part is the area where the learner documents the requirements. The learner can click the fictitious character’s avatar to recall the dialogue contents and as a reference to document corresponding requirements. Before the learner document the first requirements, BARA will give some tips. The tips include operation guides and two requirements templates helping the learner document requirements with structured patterns. Specifically, the two requirements templates guide learners in documenting requirements from a system perspective and a user perspective respectively. The template from the system perspective looks like ”[optional preconditions] [optional pre-events] the system should [expected system response]”. For example, if the requested book is available in the library, the system should return the location of the book in the library.

1) Requirements traceability design (QG1): We just mentioned that the learner needs to document fictitious characters’ requirements. We want to know each requirement belongs to which fictitious character, and when the learn document this requirement, which conversations were recalled. So in our design, when the learner adds a new requirement, he must specify for which character the requirement is documented. As for wanting to know which conversations were recalled, we can get answers from the learner’s actions record. The detailed design of the actions record will be introduced in the next paragraph.

C. Action record design

Besides being interested in a list of requirements written by learners, we are also interested in the process of learners getting these requirements. In other words, we care about which fictitious characters were interviewed by learners, how each fictitious character’s state changes, etc. In order to know these information, BARA records learners’ actions and each action with a timestamp. Besides the list of requirements written by the learner, a list of actions will also be included in the game save that the learner finally submits. With this additional action information, we can make more detailed assessments of learners’ learning effects. For example, with this information, we evaluated whether learners were attentive when playing BARA, which we will introduce in the evaluation section.

IV. A LIBRARY SCENARIO INSTANCE

In order to evaluate BARA, we implemented BARA in Unity and constructed a scenario according to the simulation goals. BARA can be download from this google drive link[2]. The link contains two versions of BARA for Windows and MacOS respectively. In the rest of this section, we will introduce the library scenario. The scenario is a university library. The scenario’s background is that a university library plans to develop a new library management system. A learner will play the role of a requirements analyst, explore the scenario, identify stakeholders, and elicit their needs. The scenario includes three sub-scenarios and 13 fictitious characters.

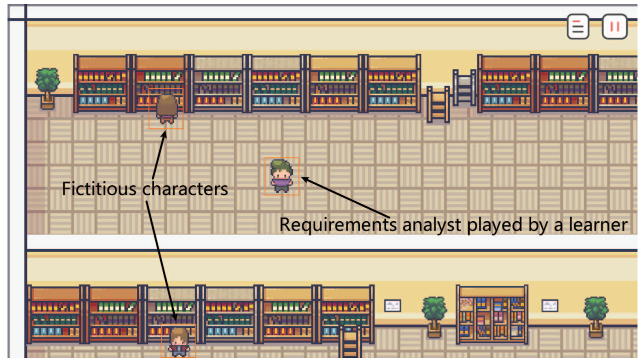

Fig. 5. The screenshot of library main hall

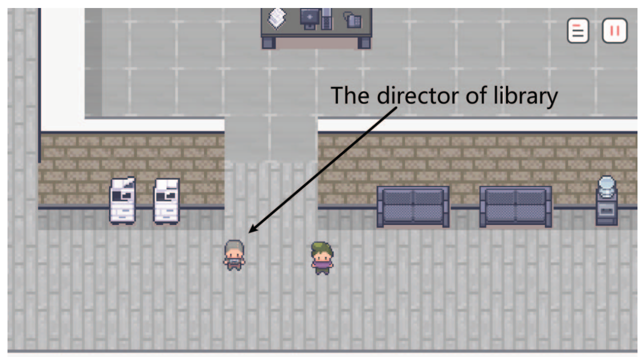

Fig. 6. The screenshot of the director’s office

The three sub-scenarios are the main hall, the director’s office, and the study room. The partial screenshots of the main hall and the director’s office are shown in the Fig. 5 and Fig. 6. There are fictitious characters in each sub-scenario, and each character represents a specific role:

- Director of library : he will provide top-level business requirements. When deciding whether a requirement is reasonable or resolving conflict requirements, the learner need to refer to these business requirements.
- Students: they are the primary user of the library, expect a series of new functions, such as online book borrowing, study room seat reservation, etc.
- Librarians: they expect managerial functions, such as automatically reminding students to return books.
- Others: they are off-campus persons and a cleaner, may also provide helpful information.

These fictitious characters will provide reasonable or unreasonable needs, and these needs may conflict. The learner needs to comprehensively analyze these needs based on collected infor-

[2]http://shorturl.at/eoxC7

mation and only document a list of reasonable requirements with no conflict as the output of BARA.

Specifically, we detail the design of three characters:

1) Student S: In the beginning, S focuses on his affairs. Through interviews with him, the learner will know this and be told to come back to him later. Now, the state of S is "Bother". If the learner ignores this message, interview him again, and his state will change to "Angry". At this time, the learner can not get information from him anymore, thus losing valuable needs. The appropriate approach is that the learner should wait for more than 20 seconds after the state of S changes to "Normal". Then, the learner can gather his needs. The setting of Student S allows learners to realize that the timing of requirements elicitation is important.

2) Librarian L: The initial state of L is "Normal", and the state will change periodically in "Angry" and "Normal". The reason for his anger is that the existing system would lag occasionally. Only in the "Angry" state, L will tell the learner this phenomenon and propose corresponding improvement expectations. The setting of Librarian L allows learners to recognize that stakeholders often do not have a clear idea of their needs.

3) Cleaner C: The need of C is that the system's mobile application can display the real-time number of people in the library. Thus it is convenient for him to pick a time when there are fewer people to clean up. This is a reasonable requirement because the director has a similar expectation. Such a setting allows learners to recognize that determining whether a requirement is reasonable should be based on a comprehensive analysis of business objectives rather than the occupation of the person who raised the requirement.

V. Evaluation

We evaluate BARA with the library scenario described in the previous section. The participants in the experiment are 60 computer science undergraduates. Specifically, in a software engineering classroom, the teacher first taught the knowledge of requirements elicitation. Then, the teacher introduced the functions and gameplay of BARA and assigned BARA as homework to the students. Students are required to submit the BARA exported results file one week later. Each file includes a list of requirements they elicit from fictitious characters. Besides a list of requirements, for each learner, BARA also records additional information, including game duration, access of each fictitious character, and a list of detailed learner's actions records.

For the following subsection, we first propose two research questions. Then answer these two questions by conducting experiments using the collected data.

A. Research questions

RQ1: what is the quality of requirements written by the learners?

RQ2: what is the learners' concentration when playing BARA?

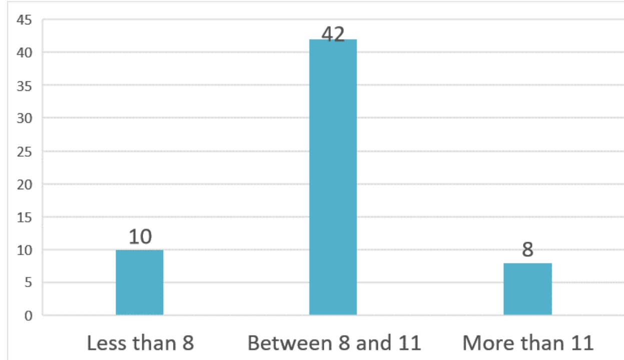

Fig. 7. Distribution of the number of requirements written by learners

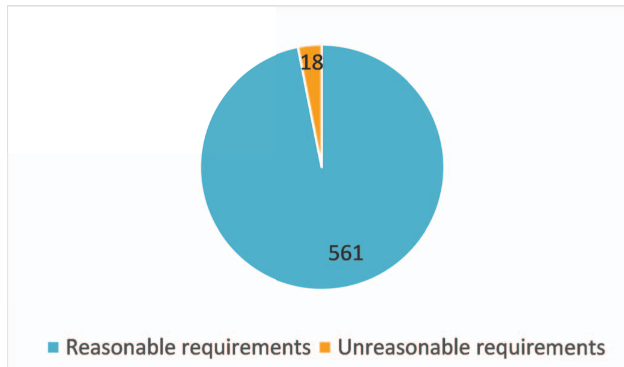

Fig. 8. Proportion of unreasonable requirements

B. RQ1: Quality of requirements written by learners

Each learner submitted a list of requirements they elicit from fictitious characters. We recall that fictitious characters in the scenario have their needs. These needs include reasonable, unreasonable, and conflicting. We ask the learners to document what they believe are correct and non-conflicting requirements as final submissions.

The Fig. 7 is the requirements number histogram of 60 learners. We consider there are 11 reasonable requirements in the scenario. As can be seen, most learners write at least eight requirements, which shows that they played BARA seriously. There are eight learners record requirements more than 11. We checked and found that they recorded additional requirements according to the project background. We think these requirements are also reasonable, so we also count them.

The Fig. 8 shows the proportion of unreasonable requirements in all requirements. We can see that in all 579 requirements, there are only about 3% unreasonable requirements. Specific to each learner, we can see from the Fig. 9, 80% of the learners identified unreasonable requirements and did not record them.

C. RQ2: learners' concentration

Besides a list of requirements in learners' submissions, there is also a list of actions. In the process of learners using BARA,

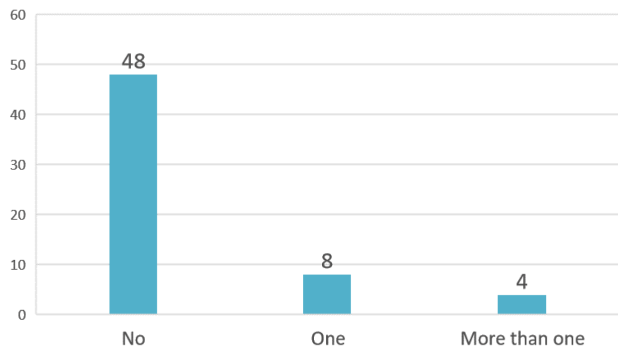

Fig. 9. Distribution of the unreasonable requirements written by learners

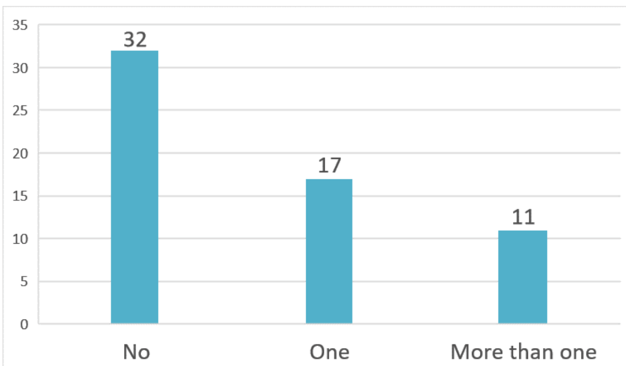

Fig. 10. Distribution of interruption times

every action will be pushed to this list with a timestamp. With these records and timestamps, we can check how focused learners are when using BARA.

Specifically, we calculate the time difference between adjacent action records for each learner's action list. If the time difference is more than 300 seconds, we will consider this an interruption. The number of interruptions histogram is shown in the Fig. 10. We can see nearly five in six learners experience BARA with no interruptions or only have an interruption which shows that they are focused when using BARA.

VI. Threat to Validity

There are several types of threats to validity. We follow the four categories detailed in [8], i.e., internal, construct, conclusion, and external validity for the discussion in the following four subsections.

A. *Internal validity*

Internal validity considers the relationship between the program and the outcome. It assesses whether the observed results were due to the program or other possible factors.

In simulated scenarios, stakeholders may have unreasonable and conflicting needs. BARA requires learners to discard the unreasonable and conflicting requirements elicited from different stakeholders and record a list of reasonable and non-conflicting requirements. This makes it hard to discriminate whether learners have identified unreasonable and conflicting requirements or just ignored them. To mitigate this threat, we can combine learners' action records for comprehensive analysis. Suppose a learner has a conversation with two stakeholders with conflicting needs but ends up recording only one's requirements. In that case, we consider he recognizes there is a conflict and records only the requirements that he thinks reasonable. By checking the action records of those learners who do not record conflicting requirements, 45 of them interviewed with all stakeholders. Therefore, we consider they indeed identified conflicting needs. Conflicting requirements are the most important starting point for further elicitation and negotiation, typically resulting in better and more requirements. Therefore, in future work, we will improve BARA so that learners can explicitly mark conflicting requirements.

B. *Construct validity*

Construct validity is related to generalizing. It refers to whether the program is constructed in a way that it successfully tests what it claims to test.

In this experiment, playing BARA is assigned to learners as an assignment. They are required to return the game save after a week. Learners can play BARA several times but only submit the final game save, so we may not be able to obtain the learner's complete learning process. To mitigate this threat, we have emphasized to the learners that the entire game process should be completed in one game save. In future work, we will improve BARA so that the complete learning process can be recorded.

C. *Conclusion validity*

Conclusion validity concerns the ability to draw correct conclusions from the results obtained through experiments. In section 5, we evaluate the quality of requirements written by learners. From the three statistical indicators, i.e., distribution of the number of requirements, proportion of unreasonable requirements, and distribution of the unreasonable requirements written by learners, we concluded that most learners recorded high quality requirements. Judging whether a requirement is reasonable and its quality is done manually. This could be a threat to conclusion validity. To mitigate this threat, we designed a scoring standard, considering the word count of each requirement and whether the sentence structure is complete. Each requirement will be scored 1-4. Only requirements with a score of 2 and above will be considered a valid requirement and counted. We also asked two people to score each requirement separately. For the requirements with controversial scores, the two people gave the final scores through discussion.

Another threat to conclusion validity is the complexity of simulated scenarios. If the simulated scenarios are too simple, the communications between the learner and fictitious characters are only a few linear conversations. Then the learner can easily identify conflicting and unreasonable requirements.

While in reality, communications between requirements analyst and stakeholders can be more time-consuming and complex. To mitigate this threat, we designed fictitious characters based on real requirements elicitation scenarios and modeled their behaviors using finite-state machines. In future work, we will design more complex simulated scenarios so that learners can recognize the difficulties in the requirements elicitation process.

D. External validity

External validity refers to the generalization of conclusions from the experimental setting to other situations.

The learners' lack of motivation can be a threat to external validity. There is no subsequent use for the requirements recorded by learners, so the learners have nothing to gain or lose by performing better or worse. We assign playing BARA as an assignment in this experiment, and the game results are part of the course grade. In future work, we plan to incorporate more gamification elements, such as badges, to improve learners' motivation.

VII. Related work

In this section, we separately focus on RE teaching with and without real stakeholders.

A. RE teaching with real stakeholders

There are some works [9]–[11] focusing on teaching requirements engineering through cooperating with real organization. These works show us that through careful curriculum design, learners can participate in the work of requirements engineering under the guidance of teachers, which can effectively allow learners to understand RE-related work and the connections between various RE stages. In this process, the output of learners can also be utilized, which is a win-win on both sides.

Penzenstadler et al. [9] argue that although simulating stakeholders is an alternative in requirements engineering teaching, only real stakeholders can provide a real experience. This project is divided into two parts. In the first part, nine graduate students elicit requirements through interviews with some different stakeholders. In a subsequent part, four undergraduates went on to implement software systems based on these requirements. Since requirements elicitation and the implementation of software are guided by the same teacher, the quality of the requirements elicited and the impact on subsequent implementation of the software can be evaluated.

A subsequent work [10] by Penzenstadler et al. focuses on improving the engineering motivation of undergraduate students to learn needs. The author argues that one way to let the learners work with partners is to develop case studies of socially relevant systems. Experiments have proved that this method can indeed increase interest and increase learners' understanding of requirements engineering.

Gabrysiak et al. [11] also hopes to improve the teaching effectiveness of requirements engineering through Cooperating with real stakeholders. This work describes their experience teaching RE through case in two universities. The results show that a study case-based approach to teaching requirements engineering greatly enhances skills valued by the industry and is enjoyable for students, teachers, and stakeholders.

B. RE teaching without real stakeholders

In reality, realistic projects and stakeholders are usually not easy to find. So more works were proposed solutions for teaching requirements engineering when authentic projects and stakeholders are not available. The main challenge is to simulate real-world scenarios so that learners can recognize the problems that may arise in the practice of requirements engineering.

Some works [12]–[15] applied role-playing as the teaching approach for learners. In their approach, they divided learners into several groups, and each group represents a specific role in the requirements engineering (e.g., client, builder, and auditor). Each client team is to propose an idea and aims to turn it into a software. Each builder team is responsible for requirements elicitation for the client team. Each auditor team needs to validate requirements modeled by the builder team. The three groups communicate with each other under the teacher's guidance and work together to complete a project-based requirements elicitation learning task. This teaching approach can cover a broader range of requirements engineering rather than only focusing on requirements elicitation. Moreover, each client team will propose a different idea, which can give other builder and auditor teams a different experience. Although the quality of the scenarios can be guaranteed under the control of the teacher, such a scenario cannot be preserved for future learners to experience. In our approach, the scenario can be constructed into BARA so that it can be experienced by future learners while ensuring the scenario's quality.

There are also some works attempts to use digital serious games to construct simulated scenarios to address the challenge. Knauss et al. [16] designed a game and visualized a process of building the right system within available time. In the game, a learner needs to make a series of decisions to complete the project with a balance of speed and quality, and the game will give the learner feedback on every decision. Through this process, the game helps learners understand and take requirements engineering seriously. Its basic concept, "Software Quantum", is abstract and unsuitable for beginners to understand the difficulties of the requirement elicitation. In BARA, learners do not need to understand additional concepts and just need to follow the guidance to complete the requirements elicitation task.

Similar to the above work, Rusu [7] designed an interactive, decision-based game. The scenario's background is to develop a defense system to protect the earth. A learner needs to elicit requirements from fictitious characters through a limited number of decision-based conversations. The fictitious characters may lie, deceive and contradict one another, so the learner must make decisions after understanding and analyzing. Finally, the game will show the learner whether the defense system based on his decision can resist the alien invasion, which makes the learner realize that requirements elicitation is not easy. This approach is similar to ours. The design that allows learners to select questions to ask fictitious characters can allow learners to obtain better learning effects. Nevertheless, the order in which questions are asked does not seem to affect the characters' responses, which is addressed in our work by modeling characters' behavior using finite state machines.

Ye et al. [17] and Vega et al. [18] constructed scenarios in Second Life[3] using its building and scripting capabilities and simulated various activities in software requirements. The former focus on simulated software specification activities, while the latter aims to simulate various activities in software requirements workshop. Building a scene based on Second Life facilitates cooperation between learners, but this is a threshold for learners who have not played Second Life. While BARA is a cross-platform game. Learners can use BARA intuitively without any complex process.

Hainey [19] developed a games-based learning application to teach requirements elicitation and analysis. Learners move through the game and "talk" to the non-player characters in the game. The control experiments show the game-based approach are better for teaching requirements elicitation than role-playing approach.

Similar to the above work, Ibrahim et al. [4] designed and developed a digital serious game for requirements elicitation and analysis following the ADDIE model. The game provides several increasingly difficult scenarios. BARA is similar to this game but pursues multiple scenarios in different fields.

Laiq et al. [20] designed and developed an AI-based interview simulator to help learners improve their interview skills. A learner can interview with the simulator with natural language and document a set of requirements. The experiment results show that the learners were able to create a requirements specification using the simulator, and the feedback is generally positive.

Garcia designed and developed two serious games [5], [6] that are similar in form but with different focuses. They all provides a library scenario so that learners can explore, analyze, and elicit requirements from fictitious characters. The former tends to exercise learners' ability to identify stakeholders, while the latter aims to supports the teaching the fundamentals of the ISO/IEC/IEEE 29148:2011 Systems and Software Engineering. However, they only construct a library scenario respectively, which results in some modules cannot be used multiple times (i.e., the data flow diagram module). In contrast, BARA has good extensibility, and new scenarios can be added at any time by loading external files.

C. Serious games in other teaching areas

As well as in the area of RE teaching, serious games are also used in a wide range of other teaching areas.

Evangelou et al. [21] developed an educational game called "My life as a software engineer", which focuses on simulating the technical difficulties or ethical dilemmas that software engineers may encounter in real work experiences. Through this game, learners can experience situations and challenges associated with the software engineering work environment without risk. The feasibility of this game has been verified by the analysis before and after the experimental test, and it has a statistically significant learning effect.

Intending to combine education and evaluation, Calder et al. [22] designed a serious game for training and evaluating students in software project management. This game offers two game modes. In the basic game mode, the game is divided into three stages: onset, execution, and end, which respectively correspond to the initiation and planning process, the cycle of executing, controlling and planning, and the closing process group in the project management process. It achieves the purpose of education through the simulation of reality at the process level. In the final stage, it evaluates the skills acquired by learners through the recording of the game process and the evaluation standards formulated by teachers. BARA similarly records learners' behavior, and we evaluate the learners' concentration.

In the area of programming teaching, Muratet et al. [23] used Real-Time Strategy Game (RTS) as the basis to produce a serious game dedicated to strengthening programming skills. The game uses an open-source real-time strategy game as its base framework and changes the way the game is played through the creation of an Applicative Programming Interface (API) named "Prog&Play". Through this interface, learners can interact with the game using commands similar to programming languages. The game uses a series of campaign to guide learners and motivate them through the background setting and the sense of involvement of being the protagonist. BARA uses a similar mode to motivate learners by designing a series of fictitious characters.

Beckers et al. [24] proposed a card based serious game to elicit social engineering security requirements. The card based game can reduce communication barriers between learners and can arise learners' security awareness of social engineering attacks. The game have been tested with several groups of researchers, IT administrators, and professionals from industry. The results show that the learners' security awareness can be effectively enhanced through playing the game.

VIII. Discussion

In this section, we discuss the potential value of using serious games for teaching requirements elicitation.

A. Crowdsourcing-based scenario building

The primary advantage of our proposed serious game is to realistically model and simulate stakeholders' behaviors so that learners can experience and understand the practical challenges during requirements elicitation. In order to pragmatically promote the usage of our proposal and unleash its full power, it is essential to establish a repository of realistic scenarios. We have so far developed two scenarios based on our project

[3]https://secondlife.com/

experience, which certainly cannot cover all the challenges of requirements elicitation.

Although we can step-by-step establish a scenario repository by ourselves, we believe a crowdsourcing approach would be very helpful and pragmatic. Especially, we aim at inviting domain experts who are knowledgeable in certain domains to develop realistic scenarios. To this end, we designed and developed an easy-to-use tool to facilitate the process of scenario construction so that non-programmers, e.g., domain experts, can easily construct scenarios with their working experience.

In addition to inviting domain experts to construct the scenarios, we also plan to incorporate the scenario development task with the requirements elicitation teaching. Specifically, with the help of our developed scenario editor, we plan to ask learners to construct a "reachable" scenario based on their practical investigation. We first ask learners to empirically investigate a particular scenario that they have access to, based on which they need to construct the scenario using the scenario editor. When designing fictitious characters within the scenario, a learner will further consider the problems that may arise in the process of actual requirements elicitation. We think this design process is of great benefit to the learner.

By considering the above two alternatives of crowdsourcing tasks, the next step of our research is to establish a scenario repository that covers a broad spectrum of practical challenges in requirements elicitation.

B. Learners' behaviors pattern mining

BARA records every important action of the learner during a game session. Each learner will generate a sequence of actions after playing BARA. In this work, the sequence data is used to analyze the learners' concentration during playing BARA. Another intuitive usage is to check whether a learner directly copied someone else's results to complete the task. Because it is obvious that learners who have fully experienced within BARA and learners who have directly copied the results of others.

However, we can further exploit it with sequential pattern mining approaches [25]. After using these pattern mining approaches to the sequence data, we can get a series of frequent patterns.

These frequent patterns have many uses. One use of these patterns is to use them to cluster the learners. The behavior patterns of learners in different clusters will be different. For example, the patterns of learners who document requirements while interviewing and the learners who document requirements after interviewing all fictitious characters are different and will be clustered into two distinct clusters. We can first discover this interesting phenomenon by clustering and then select 1-2 learners from each cluster to conduct individual interviews, asking them some thoughts on playing BARA and their opinions and suggestions on BARA. This process contributes to both the assessment of teaching effectiveness and the improvement of BARA.

Moreover, the frequent patterns can be used to discover BARA's flaws. We argue that the design flaws will cause learners to repeat the same actions so that they can be captured by pattern mining algorithms [26]. In our work's experiment, from the learners' actions record, we observed that many learners interview a fictitious character repetitively. After investigation, it was found that, because of a bug, the fictitious character was supposed to tell the learner some information but did not.

In addition, we can use these frequent patterns as reference to define a series of normal behaviors. Then, we can use these normal behaviors as filters to filter anomalous behaviors from the actions sequence. With these anomalous behaviors, we can make a more detailed assessment of each learner's learning effectiveness.

IX. Conclusion and future work

In this paper, we have presented BARA, a dynamic state-based requirements engineering serious game that models fictitious characters by finite-state machines and can automatically record learners' actions. 60 participants have already played BARA and submitted their game saves. We analyzed action records and found that most learners completed the game with immersion and concentration. From the analysis of requirements written by learners, most learners can identify unreasonable requirements and resolve conflicting requirements based on top-level business goals. In addition, some learners record additional reasonable requirements based on their own experience, which further reflects that these learners understand the context of the scenario. These results illustrate that learners are interested in this instructional approach and can complete the requirement elicitation task well in the simulated scenario.

In the future, we plan to explore the role of serious game from pedagogical aspects. Such as how to incorporate creating scenarios with the scenario editor and playing BARA into the RE course teaching plan to mitigate the impact of missing realistic stakeholders' involvement. Furthermore, we will improve the experiments' design and conduct more experiments, such as doing a comparative study in which some learners play BARA and others learn requirements elicitation through traditional ways. Through comparative experiments, we can further evaluate applicability and effectiveness of BARA.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (No.62162051), the Major Research Plan of the National Natural Science Foundation of China (92167102), the Project of Beijing Municipal Education Commission (No.KM202110005025).

References

[1] B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap," in *Proceedings of the Conference on the Future of Software Engineering*, 2000, pp. 35–46.

[2] C. C. Abt, *Serious games*. University press of America, 1987.

[3] M. Daun, A. M. Grubb, and B. Tenbergen, “A survey of instructional approaches in the requirements engineering education literature,” in *2021 IEEE 29th International Requirements Engineering Conference (RE)*. IEEE, 2021, pp. 257–268.

[4] Z. Ibrahim, M. C. Soo, M. T. Soo, and H. Aris, “Design and development of a serious game for the teaching of requirements elicitation and analysis,” in *2019 IEEE international conference on engineering, technology and education (TALE)*. IEEE, 2019, pp. 1–8.

[5] I. Garcia, C. Pacheco, A. León, and J. A. Calvo-Manzano, “Experiences of using a game for improving learning in software requirements elicitation,” *Computer Applications in Engineering Education*, vol. 27, no. 1, pp. 249–265, 2019.

[6] I. García, C. Pacheco, A. León, and J. A. Calvo-Manzano, “A serious game for teaching the fundamentals of iso/iec/ieee 29148 systems and software engineering–lifecycle processes–requirements engineering at undergraduate level,” *Computer Standards & Interfaces*, vol. 67, p. 103377, 2020.

[7] A. Rusu, R. Russell, and R. Cocco, “Simulating the software engineering interview process using a decision-based serious computer game,” in *2011 16th International Conference on Computer Games (CGAMES)*. IEEE, 2011, pp. 235–239.

[8] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[9] B. Penzenstadler, M. Mahaux, and P. Heymans, “University meets industry: Calling in real stakeholders,” in *2013 26th International Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 2013, pp. 1–10.

[10] B. Penzenstadler, D. J. Richardson, B. Karlin, A. Cook, D. Callele, and K. Wnuk, “Using non-profit partners to engage students in re.” in *REET*, 2014, pp. 1–10.

[11] G. Gabrysiak, R. Hebig, L. Pirl, and H. Giese, “Cooperating with a non-governmental organization to teach gathering and implementation of requirements,” in *2013 26th International Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 2013, pp. 11–20.

[12] D. Zowghi and S. Paryani, “Teaching requirements engineering through role playing: Lessons learnt,” in *Proceedings. 11th IEEE International Requirements Engineering Conference, 2003.* IEEE, 2003, pp. 233–241.

[13] P. Liang and O. De Graaf, “Experiences of using role playing andwiki in requirements engineering course projects,” in *2010 5th International Workshop on Requirements Engineering Education and Training*. IEEE, 2010, pp. 1–6.

[14] R. L. Quintanilla Portugal, P. Engiel, J. Pivatelli, and J. C. S. do Prado Leite, “Facing the challenges of teaching requirements engineering,” in *Proceedings of the 38th International Conference on Software Engineering Companion*, 2016, pp. 461–470.

[15] T. Nakamura, U. Kai, and Y. Tachikawa, “Requirements engineering education using expert system and role-play training,” in *2014 IEEE international conference on teaching, assessment and learning for engineering (TALE)*. IEEE, 2014, pp. 375–382.

[16] E. Knauss, K. Schneider, and K. Stapel, “A game for taking requirements engineering more seriously,” in *2008 Third International Workshop on Multimedia and Enjoyable Requirements Engineering-Beyond Mere Descriptions and with More Fun and Games*. IEEE, 2008, pp. 22–26.

[17] E. Ye, C. Liu, and J. A. Polack-Wahl, “Enhancing software engineering education using teaching aids in 3-d online virtual worlds,” in *2007 37th Annual Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports*. IEEE, 2007, pp. T1E–8.

[18] K. Vega, H. Fuks, and G. Carvalho, “Training in requirements by collaboration: Branching stories in second life,” in *2009 Simposio Brasileiro de Sistemas Colaborativos*. IEEE, 2009, pp. 116–122.

[19] T. Hainey, T. M. Connolly, M. Stansfield, and E. A. Boyle, “Evaluation of a game to teach requirements collection and analysis in software engineering at tertiary education level,” *Computers & Education*, vol. 56, no. 1, pp. 21–35, 2011.

[20] M. Laiq and O. Dieste, “Chatbot-based interview simulator: a feasible approach to train novice requirements engineers,” in *2020 10th International workshop on requirements engineering education and training (REET)*. IEEE, 2020, pp. 1–8.

[21] S. M. Evangelou, G. Stamoulakatou, and M. Xenos, “A serious game for mobile phones used in a software engineering course: Usability evaluation and educational effectiveness,” in *2021 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2021, pp. 219–225.

[22] A. Calderón and M. Ruiz, “Prodec: a serious game for software project management training,” in *Proceedings of the 8th International Conference on Software Engineering Advances, Venice, Italy*, 2013.

[23] M. Muratet, P. Torguet, F. Viallet, and J.-P. Jessel, “Experimental feedback on prog&play: a serious game for programming practice,” in *Computer graphics forum*, vol. 30, no. 1. Wiley Online Library, 2011, pp. 61–73.

[24] K. Beckers and S. Pape, “A serious game for eliciting social engineering security requirements,” in *2016 IEEE 24th International Requirements Engineering Conference (RE)*. IEEE, 2016, pp. 16–25.

[25] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, “A survey of sequential pattern mining,” *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 54–77, 2017.

[26] L. Tong, W. Yiting, and G. Congkai, “Detection of anomalous modeling behavior: a goal-driven data mining approach,” in *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, 2022, pp. 142–145.