Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 7952816, 8 pages
https://doi.org/10.1155/2021/7952816

Research Article

Development of AHMES (Automatical Higher Mathematics Examination System) Using Rational Unified Process

Le Chen and JeongYoung Song

Department of Computer Science, Paichai University, West District, Daejeon 35345, Republic of Korea

Correspondence should be addressed to JeongYoung Song; jysong@pcu.ac.kr

Received 10 September 2021; Revised 11 October 2021; Accepted 15 October 2021; Published 2 November 2021

Academic Editor: Sang-Bing Tsai

This study was aimed at developing an automatic exam system as an intelligent and high-quality higher math examination solution for Department of Computer Engineering at Pai Chai University (PCU) based on S/W Engineering. As a research and development (R&D) project, the study used the rational unified process (RUP) method for software development. RUP describes how to effectively use commercial and reliable methods to develop and deploy software system. It is a heavyweight process; therefore, it is particularly suitable for large teams to develop large projects. This paper describes the RUP process of this R&D project what we named it as Automatic Higher Mathematically Exam System (AHMES). AHMES provides a new way to automatically generate exam questions. In the study, our team and the requesting department collected the requirements and selected methods and tools. Then, our team designed and implemented the framework of the system, completed the development of some main functions, tested and summarized the system, and planned the future work.

1. Introduction

A system named "Automatic Higher Mathematic Examination System (AHMES)" was developed based on RUP process in this study. The linear algebra subject of AHMES has been equipped with 11 question-generating engines and 181 mathematical models. Random questions can be generated automatically by the system based on the mathematical model. AHMES could collect the participants' answers and adjust the difficulty level of the mathematical model based on the correct rate of these answers. The linear algebra function produced by AHMES has been tested by teachers and students. We will improve the linear algebra subject in the subsequent phase and gradually develop functions for other mathematics subjects.

2. Background

The role of exam question-generating software in the current educational environment is becoming more and more important. It can save a lot of time and energy for educators, reduce the artificial uncertain factors in exam paper questions, and prevent cheating.

The previous examination system had a question database consisting of more than 100 fixed questions. When users use this system to generate an exam paper, the system will randomly select a certain number of questions from the question database to form the exam paper. In this mode, although mathematics teachers can add, delete, and modify questions in the question database at will, it will undoubtedly require a lot of efforts, and the exact same question may appear repeatedly in each exam paper generated. This system is undoubtedly unacceptable for the scientific discipline of mathematics. Therefore, a new examination system that is different from the old one, a smarter, scientific, and controllable software system, needs to be developed.

3. Methods

First, the requirements of the new exam system need to be collated before choosing the appropriate approach and tools.

3.1. Requirement Collation. The most difficult part of building a software system is to determine what to build [1]. The work clarifying requirements laid the foundation for construction of the system. Management of requirements will run through the entire development life cycle of RUP so that the analysis of requirements in each phase of RUP is very important.

In the preliminary communication with the department that proposed the requirements, the problems that need to be solved were collected and listed as follows:

(1) The system needs to create exam papers based on the difficulty level and number of questions specified by teachers.

(2) The system needs to turn mathematical problems into software models (mathematical models) and to create exam questions based on the mathematical models.

(3) Each model has an attribute of the difficulty level.

(4) In the initial phase, the system is designed to handle the linear algebra subject. Other higher mathematics subjects, such as calculus, may be added in the subsequent phases.

(5) The software can adjust the difficulty level of the mathematical model according to the historical data of the exam results.

3.2. Approach and Tools

3.2.1. Software Engineering Approach. Since the requirements' analysis suggested that new mathematical models could be added to the initial functions, and the system should be flexible and easy to scale up. The Rational Unified Process (RUP) is an iterative software development process framework created by the Rational Software Corporation, a division of IBM since 2003 [2, 3]. RUP was selected as the tool for the AHMES project because it is use-case driven and provides a process framework that can be adapted and extended to meet the growing demands.

3.2.2. Development Tools. Python 3.8 was chosen as the programming language for AHMES project because *Python* is a multiparadigm programming language and has filter, map, and reduce functions, list comprehensions, dictionaries, sets, and generator expressions [4].

In addition, due to *Python*'s extensive mathematics library and the third-party library NumPy that further extends the native capabilities, it is frequently used as a scientific scripting language to aid in problems such as numerical data processing and manipulation [5, 6].

For these reasons, choosing *Python* means that, in the AHMES project, we could focus more on design rather than programming. In addition, if a future version requires high performance, *Python* code can be easily converted to C code.

3.2.3. Text Formatting Tools. We have two solutions to export the exam paper. The first solution is to export as a file in Docx (Microsoft Word) format. The second solution is to export as a PDF file using LaTeX. The benefits of the first solution include that it is easy to implement and files in Docx format can be easily modified. The advantage of the second solution is it can output almost perfect mathematical formulas. In the initial phase, the first option is used temporarily. The second solution may be used in future iterations.

3.3. Workflow Design. RUP is a general process template that contains many development guides, products, and role descriptions involved in the development process. Because it is very large, it needs to be tailored when using RUP for specific development institutions and projects, that is, RUP must be configured. RUP is like a metaprocess. Many different development processes can be obtained by tailoring RUP. These software development processes can be regarded as specific examples of RUP. Reference [7] shows a simplified RUP model.

Similarly, AHMES as a R&D project does not require a complete RUP model. After selection and configuration, the RUP includes these workflows: Business Modeling, Requirements, Analysis and Design, Implementation, Test, and Project Management. The reason why Deployment, Configuration and Change MGMT, and Environment are discarded is that their usage scenarios are usually commercial and large-scale development projects and are not suitable for this project.

The modified RUP model used in this project is shown in Figure 1.

3.4. Mathematical Modeling. A mathematical model is a description of a system using mathematical concepts and language. The process of developing a mathematical model is termed mathematical modeling.

The one characteristic that differs AHMES from traditional examination systems is to generate questions based on mathematical models. The mathematical models are implemented with *Python* code based on the known topic types.

We use a simple example and a complex example to illustrate the mathematical models of AHMES.

3.4.1. A Simple Example. Figure 2 shows the most basic matrix addition model.

Matrix addition requires that the two matrices have the same shape, that is, have the same number of rows and columns. The number of rows and columns and the matrix values are generated based on the difficulty level input by the teacher. As the difficulty level increases, the amount of calculation will increase.

3.4.2. A Complex Example. The model uses the known matrices A and Y and other fixed conditions to calculate the nth power of the unknown matrix X. The relationship between A, Y, and X is $\mathrm{XA} = \mathrm{AY}$. The number of columns, rows, and values of matrices A and Y are determined according to the difficulty level provided by user. When the

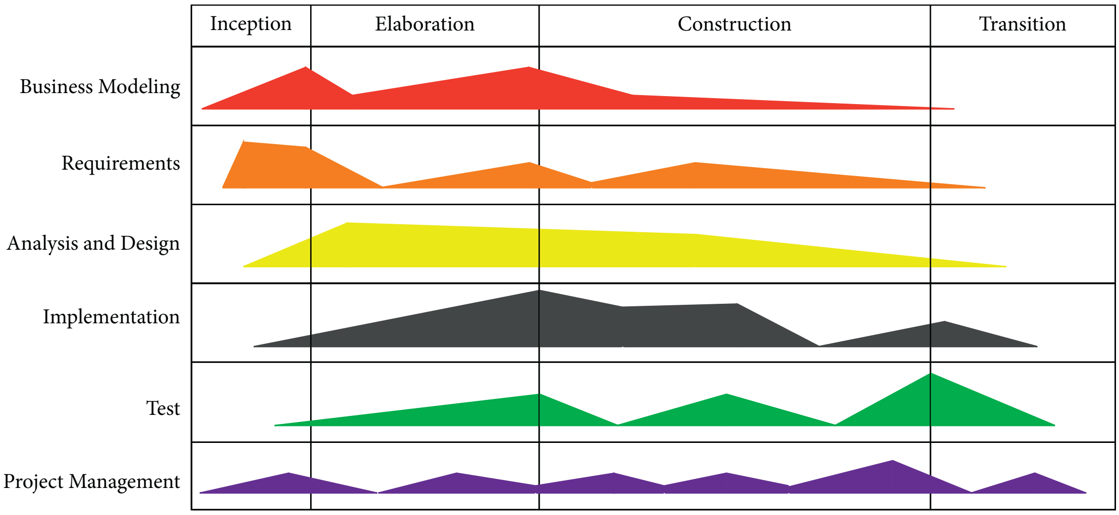


Figure 1: The modified RUP model used in this project.

$$\begin{pmatrix} A_{11} & A_{12}\ldots & A_{1n} \\ A_{21} & A_{22}\ldots & A_{2n} \\ \ldots & & \\ A_{m1} & A_{m2}\ldots & A_{mn} \end{pmatrix} + \begin{pmatrix} B_{11} & B_{12}\ldots & B_{1n} \\ B_{21} & B_{22}\ldots & B_{2n} \\ \ldots & & \\ B_{m1} & B_{m2}\ldots & B_{mn} \end{pmatrix}$$

Figure 2: Matrix addition model.

difficulty level is less than a certain value, n will be a known small integer. When the difficulty level is greater than a certain value, n will be an unknown number.

For students, the questions generated by this model are interesting and challenging. In a later section, we will list an example question generated by this model at the highest difficulty level.

3.5. Difficulty Control. Referring to Qiong Chen's difficulty setting method in his exam system [8], we subdivided the AHMES difficulty degree into 5 levels instead of 4 levels used by Qiong Chen.

In AHMES, each mathematical model has an attribute of difficulty degree. The difficulty range of a question generating engine is determined by the mathematical models it contains.

Define D as the difficulty reference value of a certain mathematical model:

$$D = 1 - \frac{\overline{x}}{100}, \tag{1}$$

where $\overline{x}$ is the average score out of 100 for all the students on this model.

The five levels of difficulty are shown in Table 1.

Assuming a question-generating engine has 3 mathematical models and the difficulty levels of these mathematical models are 2, 2, and 4, then the difficulty range of this question generating engine is 2 to 4. AHMES will not call this engine if exam papers with difficulty level equal to 1 and 5 are needed.

Table 1: Difficulty level of the models.

Value D	Level	Description
$D \leq 0.15$	1	Very easy
$0.15 \leq D < 0.4$	2	Easy
$0.4 \leq D < 0.65$	3	Common
$0.65 \leq D < 0.85$	4	Difficult
$D \geq 0.85$	5	Very difficult

4. Results

Currently, AHMES is at the beginning of construction phase. We had first implemented the AHMES framework and some mathematical models, successfully generated exam papers and tested them and then designed a solution for the current problems encountered by AHMES.

At each phase, our team will go through all workflows, but the main tasks and goals are different at each phase, so we focused different processes (see Figure 3, for details). Brief introductions were provided to the phases of AHMES in this section, and then, two instances are explained.

4.1. Inception. In beginning of inception phase, we constructed the main business model and sorted out the needs of the entire project. Then, we used *Python* to code a simple mathematical model (see 3.4.1) and generated a question. In this phase, we focused on the two workflows with Business Modeling and Requirements.

4.1.1. Business Modeling. In AHMES, teachers input parameters such as exam subject, number of exam questions, and difficulty level. The system calls the mathematical model to generate exam papers and the keys based on the parameters entered by the teacher. After importing the exam

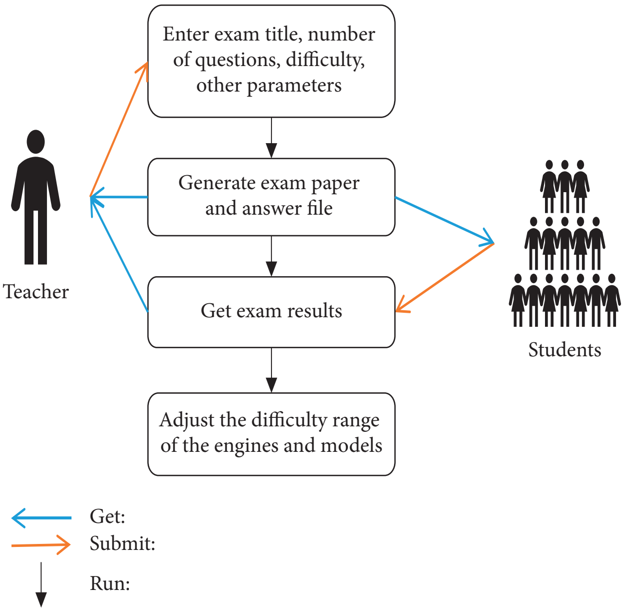

Enter exam title, number of questions, difficulty, other parameters
Generate exam paper and answer file
Get exam results
Adjust the difficulty range of the engines and models
Teacher
Students
Get:
Submit:
Run:

Figure 3: Main business modeling by use-case diagram.

results, AHMES could adjust the difficulty of the mathematical model based on historical data of students' scores.

In the RUP, use-case diagrams are often used to describe business modeling. Due to their simplistic nature, use-case diagram can be a good communication tool for stakeholders. The drawings in Figure 3 were made to mimic the real world and provide a view for the stakeholder to understand how the system is going to be designed.

4.1.2. Requirements. The requirements of AHMES could be described from three dimensions (Table 2): roles (who), works (what), and tasks (how).

4.2. Elaboration. The main task of elaboration phase was to initially build AHMES. In this phase, we designed the overall framework and constructed the main functions of AHMES. Then, we designed and constructed some mathematical models. Finally, we tested the complete AHMES and planned the next phase of tasks. In this phase, we focus on the four workflows of Analysis and Design, Implementation, Test, and Project Management.

4.2.1. Analysis and Design. The system structure is designed based on the results from analysis of requirement documents. The main flow controller of AHMES controls the operation of the software and calls various functional modules. The system includes four main functional modules: exam paper design module, question generating module, difficulty controller, and UI. A question generating engine contains one or more mathematical models with a certain commonality. Figure 4 shows the structure of AHMES.

The main functions of each module were listed in Table 3, and their interfaces were summarized in Table 4.

The UI module contains data input and output, exam paper file generating, and export.

Exam paper design module designs exam paper structure and question type of appropriate difficulty level according to the exam paper parameters input by teachers.

Question generating module contains various types of exam question-generating engines, and each exam question-generating engine contains several mathematical models. The exam question-generating module selects the appropriate exam question-generating engine according to the design by the exam paper design module. After the exam question-generating engine selects the appropriate mathematical model, the mathematical model generates exam questions based on the difficulty level and other parameters input by the teacher.

Difficulty controller includes two main functions: storage and adjustment of the difficulty level. The difficulty storage function is responsible for storing the difficulty range of each exam question-generating engine and mathematical model. The difficulty adjustment function can adjust the difficulty degree of the exam question-generating engines and the mathematical models according to historical data.

Figure 5 shows the flowchart.

4.2.2. Implementation. In this study, the framework and functions of AHMES were developed, and some mathematical models of linear algebra were implemented. The mathematical models of linear algebra subjects implemented in AHMES were listed in Table 5.

AHMES has been uploaded to GitHub, but it will not be open accessible until the completion of all work. The structure of code files for the project is illustrated in Figure 6.

4.2.3. Test. In the test, 100 exam papers with different difficulty levels were generated with AHMES and sent to teachers and students taking the exam. Teachers selected a part of these exam papers for students to finish. The feedbacks from the requesting department and students were collected and summarized in Table 6.

After receiving the answers from the students, we imported the data into AHMES and tested the difficulty adjustment function. Table 7 shows the adjustment results. Difficulty 1 is the easiest and difficulty 5 is the most difficult.

Negative points 1 and 2 among the feedbacks on AHMES were because no members of our team were experienced in teaching mathematics. Negative point 3 was because some algorithms of the exam paper design engines are not smart enough. The fourth negative point was because there is not enough historical data to adjust the difficulty levels.

4.2.4. Project Management. The AHMES project is currently at the beginning of the construction phase. The reason is that currently we have only achieved the followings:

(1) Verify the feasibility of the mathematical models

(2) Implement the framework and interface of the entire system

Table 2: Three dimensions of requirements.

Roles	Works	Tasks
Teacher	Generate exam papers	Input the title
		Input the number of exam questions
		Input the difficulty lever
	Adjust the difficulty range of the mathematical model	Collect and organize the results of the exam
		Import exam results into the system
Student	Perform exam	Obtain exam papers from the teacher
		Complete the exam
		Submit the exam paper to the teacher

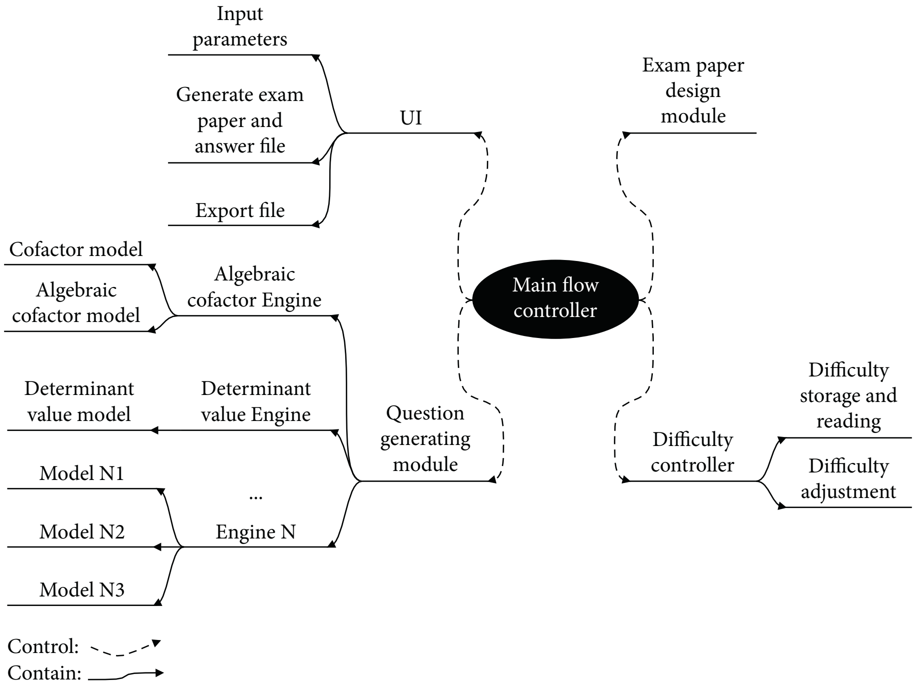


Figure 4: The structure of AHMES.

Table 3: Main functions of AHMES.

Function name	Description
Exam paper design module	This module designs the entire plan of the exam paper according to the parameters entered by the user (such as difficulty levels)
Question generating module	Based on the results designed by the exam paper design module, this module generates questions with the corresponding engines and mathematical models
Difficulty controller	Difficulty controller could store the engine's difficulty range; in addition, the engine's difficulty range can be adjusted based on historical data
UI	1. Acquire user input parameters 2. Hint information 3. Create exam papers and answer files 4. Export files
Main flow controller	It controls the flow of the software system and calls the functions

Table 4: The basic interfaces of main functions.

Interface name	Input	Output
Exam paper design module	1. Difficulty lever 2. Number of problems	Exam paper design
Engine interface	1. Difficulty lever 2. Question types	List of suitable math models
Math model	Difficulty lever	Questions and answers
Difficulty reading	Difficulty lever	List of matching engines
Difficulty adjustment	History data	Operation result

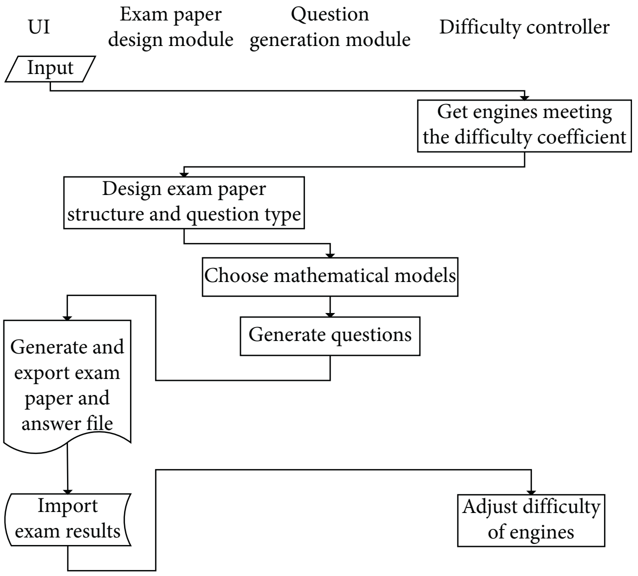

UI
Exam paper design module
Question generation module
Difficulty controller
Input
Get engines meeting the difficulty coefficient
Design exam paper structure and question type
Choose mathematical models
Generate questions
Generate and export exam paper and answer file
Import exam results
Adjust difficulty of engines

Figure 5: Flowchart.

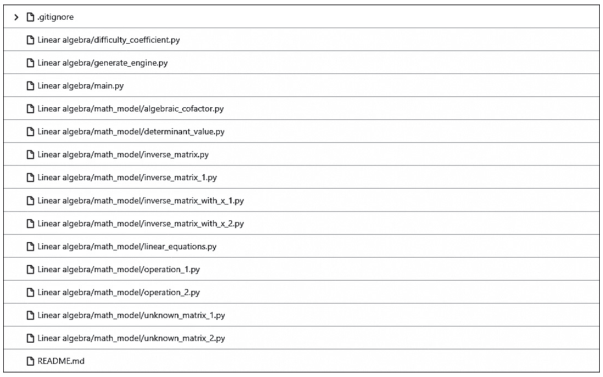
Table 5: Implementation of the inception phase.

Engine name	Number of models
Algebraic cofactor	40
Determinant value	32
Inverse matrix	17
Inverse matrix_1	12
Inverse matrix with x1	8
Inverse matrix with x2	5
Linear equations	16
Operation1	23
Operation2	14
Unknown matrix_1	4
Unknown matrix_2	10

(3) A version for internal testing is completed

(4) Plan the improvement direction of the release version

In the current version, we encountered two problems that need to be resolved. These problems and solutions are discussed in Section 4.4.



```
> .gitignore
Linear algebra/difficulty_coefficient.py
Linear algebra/generate_engine.py
Linear algebra/main.py
Linear algebra/math_model/algebraic_cofactor.py
Linear algebra/math_model/determinant_value.py
Linear algebra/math_model/inverse_matrix.py
Linear algebra/math_model/inverse_matrix_1.py
Linear algebra/math_model/inverse_matrix_with_x_1.py
Linear algebra/math_model/inverse_matrix_with_x_2.py
Linear algebra/math_model/linear_equations.py
Linear algebra/math_model/operation_1.py
Linear algebra/math_model/operation_2.py
Linear algebra/math_model/unknown_matrix_1.py
Linear algebra/math_model/unknown_matrix_2.py
README.md
```

Figure 6: Code files' structure of AHMES.

4.3. Examples. We choose the simplest matrix addition as an example to illustrate. Section 3.4.1 described the mathematical model of matrix addition. Matrix addition adapts the difficulty levels within 2.

Figure 7 shows a question generated by the model of matrix addition with difficulty level of 1.

First, the number of rows and columns of the two matrices needs to be generated. The number of rows and columns are generated according to the degree of difficulty with the help of a random function.

(1) row = random.randint (2, 2 + diff_coeff)

(2) col = random.randint (2, 2 + diff_coeff)

Then, generate two matrices based on the number of rows and columns:

(1) array_a = numpy.random.randint (–10*diff_coeff,

(2) 10*diff_coeff,

(3) size = [row, col])

(4) array_b = numpy.random.randint (–10*diff_coeff,

(5) 10*diff_coeff,

(6) size = [row, col])

Finally, add the two matrices to get the answer.

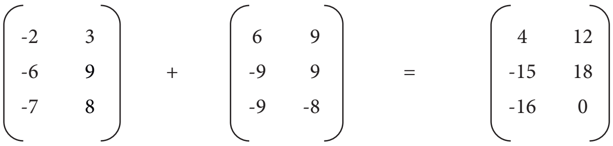
Figure 8 shows a more complicated question generated by AHMES. Section 3.4.2 describes its model. It is a problem with a difficulty level of 5 (the difficulty level of 5 is currently

Table 6: Feedbacks on the exam papers generated by AHMES.

Positive	Negative
1. Compared with the traditional way, it saves more than 90% of the time for teachers to write exam papers	1. The calculation of some questions is complicated
2. It is perfectly possible for each student to get an exam paper with completely different topics but the similar difficulty level	2. The number of question types is somewhat insufficient
3. The general difficulty of the exam paper is as expected	3. There is some confusion in the questions of exam papers
4. Engines' difficulty range could be adjusted based on historical data	4. The difficulty of some questions is not suitable

Table 7: Difficulty adjustment result.

Engine name	Before adjustment	After adjustment
Algebraic cofactor	1–5	1–3
Determinant value	1–5	3–5
Inverse matrix	1–5	3–5
Inverse matrix_1	1–5	3–4
Inverse matrix with x1	1–3	1–2
Inverse matrix with x2	1–4	2–5
Linear equations	1–5	1–4
Operation1	1–2	1–1
Operation2	2–5	1–5
Unknown matrix_1	1–5	5–5
Unknown matrix_2	2–5	5–5



$$\begin{pmatrix} -2 & 3 \\ -6 & 9 \\ -7 & 8 \end{pmatrix} + \begin{pmatrix} 6 & 9 \\ -9 & 9 \\ -9 & -8 \end{pmatrix} = \begin{pmatrix} 4 & 12 \\ -15 & 18 \\ -16 & 0 \end{pmatrix}$$

Figure 7: Result generated by matrix addition.

the highest difficulty of AHMES) generated by a mathematical model in Unknown matrix 2. The code of this model is more complicated and will be provided after we decide to open the public permissions of AHMES on Github.

4.4. Existing Problems. In the current version, our team encountered two problems that need to be resolved:

(1) The exam paper generating module in the current version is a completely random selection of mathematical models under the specified difficulty. The difficulty degrees of the questions generated by this method are fixed, and this cannot satisfy the needs in the actual situation. We hope that, in future versions, teachers using AHMES can control the proportions of questions with high and low difficulty degrees in an exam paper under the premise of controlling the overall difficulty degree of the exam paper (controlling the average score of the exam paper).

(2) The difficulty levels of mathematical models in the current version were calculated based on the correct rate of answers to the questions generated by them in the historical data. Although this method is intuitive, teachers need to manually or semimanually record and input the average correct rate of each question in exam papers for offline exams, which is almost impractical.

To solve these two problems, we plan to use some AI algorithms to improve AHMES and release the relevant research results in time.

5. Discussion

After this study, AHMES was proved enable to generate exam questions through mathematical models in linear algebra subjects, but the number and types of existing mathematical models are still not enough. At the same time, other mathematical subjects such as calculus and discrete mathematics need to be added into AHMES. AHMES has high requirements for developers' mathematical knowledge reserves and teaching experience in mathematical disciplines. To improve the rigor and practicality of AHMES, we suggest inviting math teachers to join our team.

In addition, the traditional and fixed methods currently used for designing exam papers and adjusting the difficulty degree of models are not perfect. In the next step, we plan to implement these functions using A.I. algorithms such as reinforcement learning and optimization algorithms.

There are three matrices A, X and Y satisfying $XA = AY$, where A and T are as follows:

$$A:\begin{pmatrix} 3 & 1 & 3 \\ 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix} \qquad Y:\begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

Calculate the X^n.

Answer :

$$\begin{pmatrix} 3*2^{n/4} + 3^{n/4} & -3*2^{n/2} + 3*3^{n/2} & 9*2^{n/4} - 9*3^{n/4} \\ 2^{n/4} - 3^{n/4} & -2^{n/2} + 3*3^{n/2} & 3*2^{n/4} - 3*3^{n/4} \\ 2^{n/4} - 3^{n/4} & -2^{n/2} + 3^{n/2} & 3*2^{n/4} + 3^{n/4} \end{pmatrix}$$

Figure 8: Result generated by another model in unknown matrix 2.

6. Conclusion

AHMES is an automatic generating system of mathematics questions developed for university mathematics subjects. As of now, AHMES is the first software system to generate exam questions through mathematical models in linear algebra subjects. Through this study, we confirmed that, in linear algebra subject, the idea of generating exam questions through mathematical models is feasible. The variety, scalability, and practicality of the questions generated in this way are better than the fixed questions randomly picked from a database. AHMES can provide practical help for teachers in their teaching work and solve some objective problems encountered by some students in the exam.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] P. Brooks Frederick, *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley, Reading, MA, USA, 1975.

[2] K. Darryl, *Taft (December 06, 2002)*, IBM Acquires Rational, Cupertino, CA, USA, 2002.

[3] Philippe Kruchten (January 2000), The rational unified process--an introduction, https://www.researchgate.net/publication/220018149_The_Rational_Unified_Process--An_Introduction.

[4] R. Hettinger, *PEP 289 – Generator Expressions*, Python Enhancement Proposals, Python Software Foundation, Wilmington, DE, USA, 2002.

[5] H. Koepke (3 February 2019), 10 Reasons Python Rocks for Research (And a Few Reasons it Doesn't), https://www.stat.washington.edu Retrieved.

[6] Shell, S. 2014), An Introduction to Python for Scientific Computing.

[7] Rational Unified Process, Wikipedia, Retrieved.

[8] Q. Chen, "An application of online exam in discrete mathematics course," in *Proceedings of the ACM Turing Celebration Conference-China (TURC '18)*, New York; NY, USA, May 2018.