e-ISSN 2963-2870
Vol 3 2022

DEVELOPING AUTOMATIC ITEM GENERATION

Heru Widiatmo[1], Adi Suryanto[2], Kani[3], Mawardi[4]

[1, 2, 3, 4]*Universitas Terbuka (INDONESIA)*

Abstract

Developing and administering new test items each semester are important tasks for every teacher, as well as educational institutions, so that the assessment of student learning outcomes is meaningful and reliable. On the other hand, the time, effort, and competencies of teachers in particular are limited for frequently developing new items in a short time period. Hence, they need assistance to do so.

One method to produce new and good test items on a large scale and quickly is with the help of Automatic Item Generation (AIG). The potential benefits of AIG are promising. Through this research, the development of AIG software has been completed. The purpose of this paper is to describe and illustrate a method of AIG for generating a huge number of calculus test items that are closely aligned to the test blueprint.

Keywords: Automatic item generation, Computer-assisted assessment, Calculus 1.

1 INTRODUCTION

Two important tasks that should be conducted by every teacher in elementary, middle schools, or universities is first, developing and second, administering new test items each semester (whether for quizzes, midterm, or final exams). Administering test items, that have never been exposed to students, increases the test security and the validity level of the test results (Emberston, S.E, & Kinston, N.M; 2018). However, developing good items is not a simple task, but takes a long time and is expensive. The task includes constructing a test- blueprint, writing items, trying them out, revising them, and forming the test packages for administration. Rudner (2010) reported that to develop one test form of multiple-choice (MC) items in a high-stake testing program takes no less than one year. With an estimated cost of $1,500 to 2,000 for each item, a test form of 50 items can cost up to $100,000.

Lack of time, effort, expertise, and resources to develop test items drives some teachers to take a shortcut. Due to fear of the answer-key-leak, they administer the same items or the same test forms but by changing the order of the items and/or the order of the options. These test constructions are not considered good test practices per the Standards for Educational and Psychological Tests guidelines (AERA/NCME/APA, 2014).

In fact, there is a method to construct the MC items without violating the Standards guidelines, namely Automatic Item Generation (AIG). This method includes a two-stage process. First, subject experts or test item developers prepare test templates or parent item models. Then, based

on the parent item models, a computer algorithm is used to generate several possible items that can be obtained from the models. Therefore, instead of the test developers writing each individual item, AIG helps the test developers to get many similar items (a family of items) quickly from one parent item model.

Many benefits of AIG have been mentioned by Gierl, M. J., & Haladyna, T.M. (2013). First, because AIG can produce many items in a matter of seconds, of course, the costs of producing test instruments are lower (Ruder, 2010). Second, parallel test packages are two or more test packages that have different items or questions, but with the same level of complexity and difficulty. Because AIG can create many parallel test packages, it is possible for each test taker to see different test questions, thereby increasing test security (Bejar, I., 2013). Third, AIG can also produce items with varying degrees of difficulty, small error rates, and higher comparability levels than items made by test writers (Embretson, S.E.; 2016). Fourth, when combined with a computerized adaptive test (CAT), it allows AIG to generate new questions at the time of the exam (on the fly) based on the individual abilities of the examinees (Blum, Diego, B., & Heinz, H.; 2018).

1.1 THE PURPOSE

Little research has been proposed in Indonesia to explore the benefits of AIG. None of the research addressed software for generating comprehensive test items. The purpose of this study is to explore the benefits of AIG by introducing a method for generating calculus test items used in the Universitas Terbuka (UT) or Indonesia Open University. A prototype software has also been developed based on this method. Hopefully, the described method can be applied to any math item. We use calculus because many freshman students at UT enroll in this general class. In addition, calculus test items are more comprehensive and complex, including graphics, pictures, diagrams, numbers, and mathematical symbols. We introduce the basic logic needed to generate items with what is called a template-based method. Using this method, the AIG can generate items that are similar to the model. We use real test items to ensure that the illustration is understandable and concrete.

1.2 LITERATURE REVIEW

Automatic Item Generation (AIG) or the creation of test questions automatically is a way to produce many good items in a short time. AIG is a method for developing test items by combining educational measurement theories or psychometrics and computer programming (Gierl et al.,

2013). Computer algorithms are used to automate item construction according to the expected test blueprint or test characteristics. This method was first developed by Bormuth, J. R. (1962).

Currently, several methods of AIG have been successfully developed for various subjects and purposes. MathGen is a math problem generator engine for students in primary and secondary education in the USA. This engine is now in its the third version, MathGen 3.0 (Wilson, J., Morrison, K., & Embretson, S.E; 2014). Embretson, S.E., & Kinston, N.M. (2018) conducted qualitative research and empirical trials to evaluate the quality and psychometric results of the items made by MathGen 3.0, and they concluded that those items met the expected psychometric characteristics.

Ferreyra, M.F., & Backhoff-Escudero (2016) constructed two parallel test forms to measure students' competencies in basic education using AIG called GenerEx. They examined and compared the structure and psychometric characteristics of the items in the two forms. They concluded that GenerEx could be relied on to generate parallel test packets. Gierl and a number of other researchers (Gierl, M.J., Lai, H., & Turner, S.R., 2012; Gierl, M.J., Lai, H., 2013; Lai, H., Girl, M. J., Byrne, B.E., Spielman, A.I., & Waldschmidt, D.M., 2016) used an AIG engine called the Item Generator or IGOR (Gierl, M.J., Zhou, J., & Alves, C., 2008) to create multiple-choice items that measure the cognitive abilities of health workers. The items were compared with the items that were created manually by the expert item writers, and the results showed that the quality of the IGOR items was as good as the items made by the expert item writers.

Arendasy, M.E., Sommer, M., & Mayr, F. (2011) used AIG to create items that measure German and English-speaking fluency. The two tests were given to native English and German speakers, respectively. The results showed that the AIG questions were psychometrically acceptable. Holling, Berling, and Zeuch (2008) applied probability theory to automatically generate mathematical word problems with a predetermined level of difficulty. The Rasch model (1960) was used to test the fit of the model. The results obtained indicate that the level of difficulty produced by the items was as expected.

The original purpose of AIG was to create items that met the test-blueprint and the requirements of the item parameters. However, those two requirements were inadequate because the test developers needed an item model (Gitomer & Bennett, 2003). To satisfy the test blueprint and to confirm that the resulting items are intended as the item model, the item model can be developed from two measurement approaches (Drasgow et al., 2006). The first approach is based on a strong

e-ISSN 2963-2870
Vol 3 2022

theory (Irvine, 2002). Using this approach, the level difficulties of an item are estimated by making assumptions about the cognitive processes required to solve the item. All generated items in this template-based method are within the range of expected psychometric properties. When cost is the main factor, the strong theory may be useful, because it is not required for a field testing for the item calibrations. The strong theories usually are found on psychological tests. They have been applied mostly to tests requiring specific tasks such as mental rotation (Bejar, 1990) and spatial abilities (Embretson & Yang, 200&). On the other hand, most subjects or achievement tests (such as calculus) don't have any strong theory.

The second approach is based on a weak theory. Design guidelines produce calibrated items (Drasgow et al., 2006). An item model named a parent item model is used to generate items. The item features in the parent item model are manipulated to produce similar items with different components. The benefit of a weak theory for an automated item generation is in terms of its practicality. The parent model can usually be found in a previous administered exam. Also, a weak theory is useful for broad content domains in which few theoretical models exist on the cognitive knowledge and skills used by examinees to solve the items (Schmeiser & Welch, 2006). Due to calculus being a university class that doesn't have the strong theories being measured, this study used the weak theory. The test blueprint of the calculus class from the previous test administration was used to develop a set of items that consists of 30 MC items. Then, each item was implemented as the parent item model to generate new test items.

2 METHODOLOGY

2.1 Data: Item Models

This study used a set of items from the calculus final exam. The form consisted of 30 multiple-choice (MC) items with four options. Table 1 shows the distribution of the test contents. In UT, the topic or contents of calculus is divided into nine parts called modules that comprise one semester of lectures. Each of those 30 items was used as a parent item model for generating new test items.

Table 1.

Distribution of Items on the calculus test

No. Item	Contents
1, 2, 3, 4	Sets and Real Numbers
5, 6, 7, 8	Functions
9, 10	Different Types of Functions and the Graphics
11, 12, 13, 14	Limits and Continuity
15, 16, 17	Differentiation I
18, 19, 20, 21	Differentiation II
22, 23, 24, 25	Analytical Applications of Differentiation I
26, 27	Analytical Applications of Differentiation II
28, 29, 30	Limits of Indefinite Functions

2.2 Method

An item model is implemented in this study as a basis for AIG. The term item model was introduced by Bejar (1996). Other researchers used different terms: schema (Singley & Bennett, 2022), blueprint (Embretson, 2002), templet (Mislevy & Riconscente, 2006), form (Hively, Patterson, & Page, 1968), frame (Minsky, 1974), and shells (Haladyna & Shindoll, 1989). We have called it a template-based method.

The MC test item is divided into three parts: stem, options, and auxiliary information (Bejar, Lawless, Morley, Wagner, & Bennett, 2003). First, stem is part of the main item that provides information or context of the item, and gives questions or problems that must be answered by test takers. Second, options are answer choices consisting of one correct answer or key and several incorrect answers, which are often referred to as distractors. Third, auxiliary information is additional information or material that could be presented in the stem or options in the form of tables, pictures, or diagrams. As additional information, the auxiliary information may or may not be present in an item. Some elements in the stem or in the options of an item can be changed. These are called components. The components are used to produce several possible new test items.

Figure 1 shows an example of the MC item from a mathematics test of elementary schools. No auxiliary information is presented in this example. Components of the item which can be changed are “Dimas”, “pens”, “9”, and “his”. How to change these components is discussed in the following.

e-ISSN 2963-2870
Vol 3 2022

Dimas brought 80 pens shared with 9 of his friends. Each of his friends has the same number of pens. How many pens left over after Dimas handed them out to his friends?	Stem
A. 9 B. 8 *) C. 7. D. 6 Note: *) key answer.	Options

Figure 1 Item Model

Based on Figure 1, we can generate the item model as in Figure 2 with our template- based method. To generate new items, each of the five components (S_1, X_1, N_1, N_2 and gender) can be replaced with any other members of the component. Therefore, using all combinations, a total of 160 new items can be generated.

$\lceil S_1 \rceil$ brought $\lceil N_1 \rceil$ $\lceil X_1 \rceil$ shared with $\lceil N_2 \rceil$ of [gender] friends. Each of [gender] friends has the same number of $\lceil X_1 \rceil$. How many $\lceil X_1 \rceil$ left over after $\lceil S_1 \rceil$ handed them out to [gender] friends?	Stem
A. $\lceil N_2 \rceil$ B. $\lceil \mathrm{mod}\left(\frac{N_1}{N_2}\right) \rceil$ *) C. if$\left(\lceil \mathrm{mod}\left(\frac{N_1}{N_2}\right) \rceil > 2\right)$,then $\lceil \mathrm{mod}\left(\frac{N_1}{N_2}\right) - 1, else\ \mathrm{mod}\left(\frac{N_1}{N_2}\right) \rceil + 1$ D. if$\left(\lceil \mathrm{mod}\left(\frac{N_1}{N_2}\right) \rceil > 2\right)$, then $\lceil \mathrm{mod}\left(\frac{N_1}{N_2}\right) - 2, else\ \mathrm{mod}\left(\frac{N_1}{N_2}\right) \rceil + 2$	Options
• $\lceil (S_1 : Gender) : \{Toto : his, Tuti : her\} \rceil$ • $\lceil X_1 : \{pencils, books\} \rceil$ • $\lceil N_1 : \{50, 60, 70, 90\} \rceil$ • $\lceil N_2 : \{5, 6, 7, 8, 11\} \rceil$	Components
2 x 2 x 4 x 5 = 160	Possible new items

Figure 2 Generating New Items

Sets and Real Numbers. Three components of the items can be changed: the sets of A, B, and the operator ('-"). Accordingly, the set of A could be {1,2,3, …, 5}, {3,4,5, …,10}, {10,11,12,13, … ,17}, the set of B could be

$$\{x|-2 \le x \le 3, x \in Z\}, \{x|3 \le x \le 10, x \in Z\}, \{x|1 \le x \le 9, x \in Z\},$$

and the operator could be {–,+}. These possibilities are shown in Figure 4. The options for each item can be created as earlier in Figure 2).

e-ISSN 2963-2870
Vol 3 2022

Let $A = \{1,2,3,\ldots,7\}$ and $B = \{x \mid -3 \leq x \leq 4, x \in Z\}$. What is $A - B$ *adala* ?	Stem
A. {1,2,3,4} B. {5,6,7}. *) C. $\{-3,-2,-1,0\}$ D. $\{-3,-2,-1,0,1,2,3,4,5,6,7\}$	Options

Figure 3

Item Model of Calculus Test Item

Let $A = set1$ and $B = set2$. What is A *operator* B ?	Stem
A. The first four members of A B. A *operator* B. C. The first members of B D. A *opposite operator* B [*opposite has 2 p's*]	Options
• $Set1 : \{\{1,2,3,\ldots,5\},\{3,4,5,\ldots,10\},\{10,11,12,13,\ldots,17\}\}$ • $Set2 : \{\{x \mid -2 \leq x \leq 3, x \in Z\},\{x \mid 3 \leq x \leq 10, x \in Z\},$ $\{x \mid 1 \leq x \leq 9, x \in Z\}\}$ • *operator* : $\{-,+\}$	Components
2 x 3 x 2 =12	Possible new items

Figure 4

Generated Calculus Items

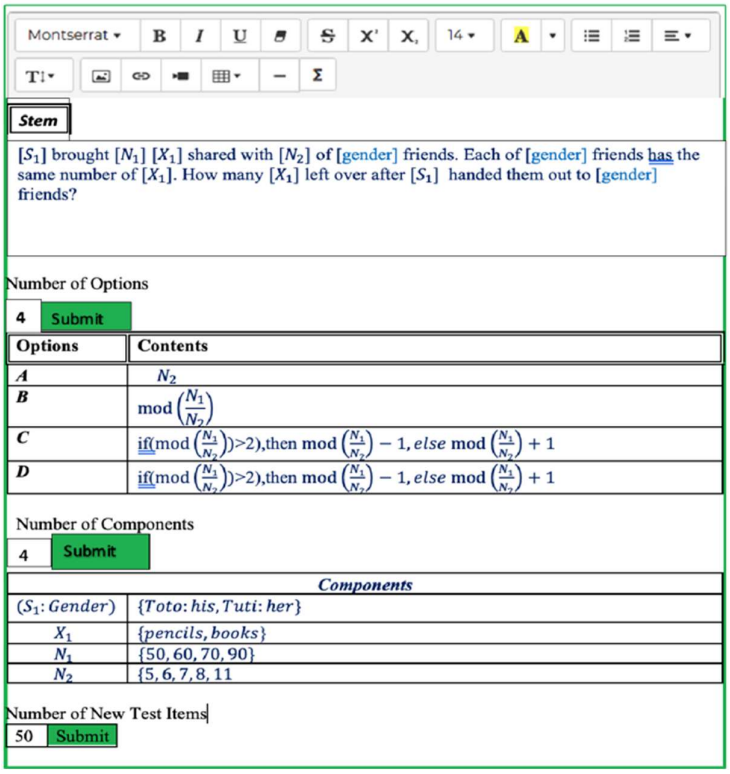
2.3 Results

Figure 5 shows the developed AIG dialog box developed as a template-based method. The dialog box shows how to generate new test items from the parent item model discussed in Figures 2 and 3. The dialog box is actually composed of three boxes. The first box is provided for the stem of the item model. The number of options needed for this item is typed in the provided box. After clicking the submit, the second box appears for users to input the information for each of the options. The inputs could be a math function, computation, graph, or any mathematical logic based on the input components

After inputting the information for each of the options, the number of components is recorded, and the third box of the components appears. After all contents of the components have been written, the last box is submitted to produce the new test items. In this example, the maximum number of new test items is 50. The user can provide how many items are needed. (There cannot be more than the possible number of the new test items.) Figure 6 shows the four new items generated from this AIG.

Table 2 displays the results of generating calculus test items used for the final exam in the IOU. Among of 30 items generated, 26 items could be generated successfully, while the three items (No. 7, 8, 9, and 10) could not. Those four items cannot be generated because they require graph manipulation.

e-ISSN 2963-2870
Vol 3 2022



Stem

[S_1] brought [N_1] [X_1] shared with [N_2] of [gender] friends. Each of [gender] friends has the same number of [X_1]. How many [X_1] left over after [S_1] handed them out to [gender] friends?

Number of Options

4 Submit

Options	Contents
A	N_2
B	$\text{mod}\left(\frac{N_1}{N_2}\right)$
C	if(mod $\left(\frac{N_1}{N_2}\right)$)>2),then mod $\left(\frac{N_1}{N_2}\right) - 1, else \text{ mod } \left(\frac{N_1}{N_2}\right) + 1$
D	if(mod $\left(\frac{N_1}{N_2}\right)$)>2),then mod $\left(\frac{N_1}{N_2}\right) - 1, else \text{ mod } \left(\frac{N_1}{N_2}\right) + 1$

Number of Components

4 Submit

Components	
$(S_1: Gender)$	$\{Toto: his, Tuti: her\}$
X_1	$\{pencils, books\}$
N_1	$\{50, 60, 70, 90\}$
N_2	$\{5, 6, 7, 8, 11$

Number of New Test Items

50 Submit

Figure 5 AIG Dialog Box

Soal No 1.

Toto brought 50 pencils shared with 5 of his friends. Each of his friends has the same number of pencils. How many pencils left over after Toto handed them out to his friends?

A. 0
B. 1
C. 2
D. 5

Soal No 2.

Toto brought 50 pencils shared with 6 of his friends. Each of his friends has the same number of pencils. How many pencils left over after Toto handed them out to his friends?

A. 2
B. 4
C. 3
D. 6

Soal No 5.

Toto brought 50 pencils shared with 11 of his friends. Each of his friends has the same number of pencils. How many pencils left over after Toto handed them out to his friends?

A. 11
B. 6
C. 5
D. 4

Soal No 6.

Toto brought 60 pencils shared with 5 of his friends. Each of his friends has the same number of pencils. How many pencils left over after Toto handed them out to his friends?

A. 0
B. 1
C. 2
D. 5

Figure 6 Four New Items

Table 2 AIG Result

No Item	Topic	Auto Generate	
		Able	Unable
1, 2, 3, 4	Sets and Real Numbers	1, 2, 3, 4	
5, 6, 7, 8	Functions	5, 6	7, 8
9, 10	Different Types of Functions and the Graphics		9, 10
11, 12, 13, 14	Limits and Continuity	11, 12, 13, 14	
15, 16, 17	Differentiation I	15, 16, 17	
18, 19, 20, 21	Differentiation II	18, 19, 20, 21	
22, 23, 24, 25	Analytical Applications of Differentiation I	22, 23, 24, 25	
26, 27	Analytical Applications of Differentiation II	26, 27	
28, 29, 30	Limits of Indefinite Functions	28, 29, 30	

3 FINDINGS AND DISCUSSION

The purpose of this study is to develop the AIG for generating new test items for the calculus final exam administered in UT The present study was able to develop successfully 87% of the items. One challenge for generating math items is graph manipulation. Table 2 shows the items that required graph, table, or image manipulation are troublesome being generated. Therefore, due to a lack of computer liberties that can create graphs, tables, or images dynamically, currently, we can only generate items that don't require graphs. Or, if it is an item with graphs, we can generate it traditionally without AIG.

Second, as described in figure 5 regarding generating options, it is required to write in computer programing language as a function or subroutine. Therefore, to effectively generate good items is expected that the test developers must have basic knowledge of a computer programing language. This condition is concerning for some test developers. They have required not only experts in the contents of the test, but also, are expected to know programing language. In this project, the basic knowledge of programming hypertext preprocessor (PHP) is necessitated for the test developers.

Third, the use of the item model changes the scope of how to construct test items. Traditionally, for a given constructed item the item writer only paid attention to that item. On the other hand, using the item model all items produced must meet the expected item quality.

Because of those limitations, AIG can be considered as a supplement tool where not all items can be created effectively and not all test developers know basic programing language. Nevertheless,

this study shows that many math items can be generated automatically. Some future research is needed to develop math libraries and graphical editors that are able to create graphs, tables, or images dynamically.

ACKNOWLEDGEMENTS

We like to thank Universitas Terbuka for the financial support for this automatic item generation research project.

REFERENCES

American Educational Research Association (AERA), the American Psychological Association (APA), and the National Council on Measurement in Education (NCME), 2014

Arendasy, M.E., Sommer, M., & Mayr, F. (2011). Using automatic item generation to simultaneously construct German and English versions on a Word Fluency Test. Journal of cross-cultural psychology, 43(3).

Bejar, I. (2013). Recent prospects and developments in item generation. I S.E. Embretson (E.), Measuring psychological constructs with model-based approaches (pp. 1-8). Washington, DC: American Psychological Association.

Blum, Diego; Holling, Heinz (6 August 2018). Automatic Generation of Figure Analogies With the IMak Package (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6087760).

Bormuth, J. R. Cloze tests as a measure of comprehension ability and readability.

Unpublished doctoral dissertation, Indi- ana University, 1962.

Embretson, S.E. (2016). Understanding examinees' responses to items: Implication for measurement. Educational Measurement: Issues and Practice, 35(3), 6-22.

Embretson, S.E. & Kingston, N.M. (2018). Automatic item generation: a more efficient process for developing mathematics achievement items ? Journal of Educational Measurement, 55 (1), 112-131.

Ferreyra, M.F., & Backhoff-Escudero, E. (2016). Validity of item generation on Basic competencies (Excoba). Relieve, 22(1), art. 2, 1-16.

Gierl, M. J., & Haladyna, T. M. (2013). Automatic item generation. New York, NY: Routledge.

Girl, M.J., Lai, H., Pugh, D., Touchie, C., Boulais, A.P., & De Champlain, A. (2016).

Evaluating the psychometric characteristics of generated multiple-choice test items.

Applied Measurement in Education, 29(3), 196-201.

Girl, M.J., & Lai, H. (2013). Evaluating the quality of medical multiple-choice items created

with automated processes. Medical Education, 47, 726-733.

Gierl, M.J., Lai, H., & Turner, S.R. (2012). Using automatic item generation to create multiple-choice test items. Medical education, 46(8), 757-765.

Gierl, M.J., Zhou, J., & Alves, C. (2008). Developing a taxonomy of item mode types to promote assessment engineering. J Technol Learn Assess, 7(2), 1-51.

Irvine, S. (2002). The foundations of item generation for mass testing. In S.H. Irvine, & P.C. Kyllonen (Eds.). Item Generation for Test Development (pp. 3-34). Mahwah: Lawrence Erlbaum Associates.

Lai, H., Gierl, M.J., Byrne, B.E., Spielman, A.I., & Waldschmidt, D.M. (2016). Three modeling applications to promote automatic item generation for examinations in dentistry. Journal of Dental Education, 80(3), 339-347.

Rasch, G. (1960). Probabilistic models for some intelligence and attainment tests. Chicago: University of Chicago Press.

Rudner, L. (2010). Implementing the Graduate Management Admission Test Computerized Adaptive Test. In W. van der Linden & C. Glas (Eds.), Elements of adaptive testing (p. 151-165), New York, NY: Springer.

Willson, J., Morrison, K., & Embretson, S.E. (2014). Automatic item generator for mathematical achievement items: MathGen3.0. Technical report IES1005A-2014 for the Institute of Educational Sciences Grant R305A100234. Atlanta, GA: Cognitive Measurement Laboratory, Georgia, Institute of Technology.

e-ISSN 2963-2870
Vol 3 2022