Overcoming Student Passivity with Automatic Item Generation

Sebastian Kucharski[1][a], Florian Stahr[1], Iris Braun[1] and Gregor Damnik[2][b]
[1]Chair of Distributed and Networked Systems, TUD Dresden University of Technology, Dresden, Germany
[2]Chair of Didactics of Computer Science, TUD Dresden University of Technology, Dresden, Germany
{sebastian.kucharski, florian.stahr, iris.braun, gregor.damnik}@tu-dresden.de

Keywords: Automatic Item Generation, AIG, Assessment, Cognitive Model, Item Model.

Abstract: Studying at German universities is often associated with a passive mode of learning. Using learning tasks and (self-)test items is an effective way to address this issue. However, due to the high cost of creation, these materials are rarely provided to learners. The approach of Automatic Item Generation (AIG) allows for the resource-efficient generation of learning tasks and (self-)test items. This paper demonstrates, after presenting general ideas of AIG, how tasks or items can be automatically generated using the AIG Model Editor designed at TUD Dresden University of Technology. Subsequently, items generated using the AIG approach are compared with items created in a traditional manner. The results show that automatically generated items have comparable properties to traditionally created items, but their generation requires much less effort than the traditional creation, thus making AIG appear as a promising alternative for supporting active learning at universities.

1 INTRODUCTION

Studying at German universities is often characterized by attending lectures, reading and processing book chapters and journal articles, or watching instructional videos. However, listening, reading, and watching is associated with a mainly passive mode of learning, which can lead to insufficiently interconnected or sustainable knowledge (Chi and Boucher, 2023; Chi and Wylie, 2014).

One effective way to address this problem is the use of learning tasks and (self-)test items. Tasks and items consist of at least a) a question that prompts learners to reflect on the content, b) a response area that allows learners to represent the results of their thinking processes, and c) a responding component that depends on the type of the task or item (Körndle et al., 2004). If the goal, on the one hand, is to encourage learners to engage in deeper processing of the subject matter and, for example, to make comparisons between different content areas or to summarize various aspects of the content, learning tasks are used, which often include elaborated feedback. If the goal, on the other hand, is to enable learners to evaluate their learning process or to encourage them to reflect on the quality of their own learning methods, (self-)test items are used. These typically contain only brief informational feedback such as *correct* or *incorrect*. Regardless of whether it is a learning task or a (self-)test item, it can help to overcome a passive mode of learning and encourage learners to actively process a subject area (Körndle et al., 2004; Kapp et al., 2015).

1.1 Reasons for the Infrequent Use of Tasks and Items at Universities

Despite the described benefits, learners are hardly offered learning tasks or (self-)test items to support an active mode of learning because of two reasons. First, creating a necessary task or item pool is costly (Kerres, 2002). It is estimated that creating a single written task or item costs several hundred euros (Gierl and Lai, 2013b). Second, tasks or items would need to be created by university instructors who are content experts but often lack the know-how for traditional task or item creation (Li et al., 2021). Therefore, a task or item pool for learning purposes seems to be unrealistic at the moment (Damnik et al., 2018). However, the digitization or widespread use of computers in higher education has changed this assessment in recent years. It is now possible to generate tasks or items easily and inexpensively using software.

This paper demonstrates in Section 2 how the Au-

[a] https://orcid.org/0009-0003-4210-5281
[b] https://orcid.org/0000-0001-9829-6994

Kucharski, S., Stahr, F., Braun, I. and Damnik, G.
Overcoming Student Passivity with Automatic Item Generation.
DOI: 10.5220/0012747500003693

In *Proceedings of the 16th International Conference on Computer Supported Education (CSEDU 2024) - Volume 1*, pages 789-798
ISBN: 978-989-758-697-2; ISSN: 2184-5026

tomatic Item Generation (AIG) approach (Embretson, 2002; Embretson and Yang, 2007; Gierl et al., 2012; Gierl and Lai, 2013b; Damnik et al., 2018; Kosh et al., 2019) can be used to efficiently generate learning tasks and (self-)test items. Subsequently, it will be explained in Section 3 how the AIG approach can be applied using an editor designed at the TUD Dresden University of Technology[1], which has been evaluated multiple times with the help of instructors and revised in several iterations (Baum et al., 2021; Braun et al., 2022; Kucharski et al., 2023). Finally, an experiment is discussed in Section 4 in which learners were asked to compare and evaluate traditionally created and automatically generated items, aiming to assess the differences between items generated using the AIG approach and those created in a traditional manner.

2 AUTOMATIC ITEM GENERATION

Automatic Item Generation (AIG) describes a technology, methodology, or process by which learning tasks or (self-)test items are generated automatically (Embretson, 2002; Embretson and Yang, 2007; Gierl et al., 2012; Gierl and Lai, 2013b; Damnik et al., 2018; Kosh et al., 2019). AIG does not require experts anymore that individually write, review, and revise tasks or items, but rather operates with systematic representations of the subject matter (i.e., *cognitive models*), systematic representations that describe the type and form of the intended tasks or items (i.e., *item models*), and software that automatically generates a task or item pool from these models (i.e., *item generator*). The overall process of AIG is divided into the following four broad stages (Gierl and Lai, 2016; Damnik et al., 2018; Kosh et al., 2019).

1. The development of a cognitive model
2. The development of an item model
3. The generation of a task or item pool
4. The evaluation of the tasks or items and models

These four stages, in turn, encompass ten specific steps, as illustrated in Figure 2. The stages and steps are explained in detail below.

2.1 Development of a Cognitive Model

A cognitive model contains the information that an expert in a particular content area needs to answer a question, the information that helps to make

[1]https://tu-dresden.de, accessed February 13, 2024

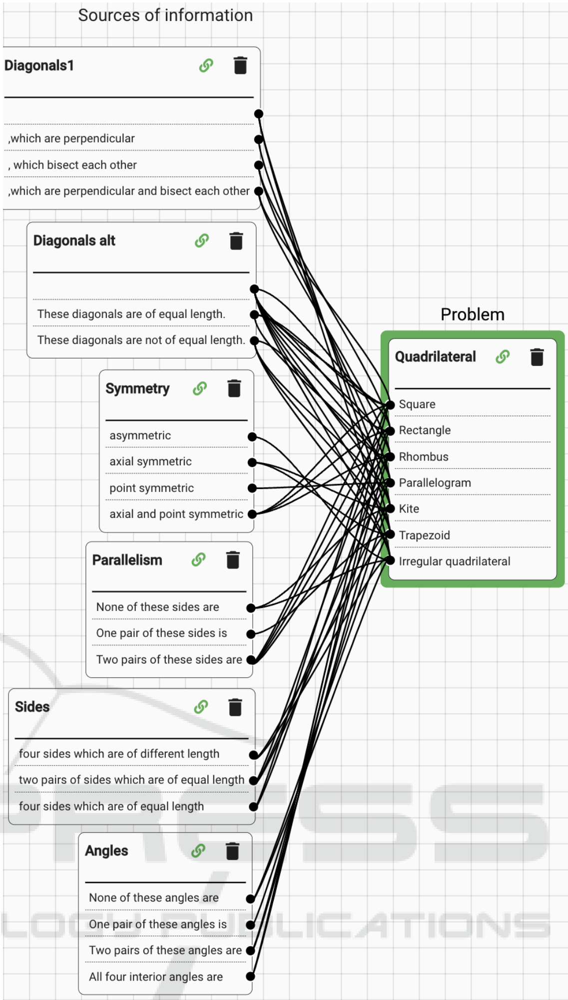

Figure 1: Example of a cognitive model of the AIG Model Editor related to different types of quadrilaterals and their differentiation.

proper decisions, or the information that best fits a given problem among various options (Gierl and Lai, 2013a). Therefore, the development of the cognitive model begins with the search for a source of expert knowledge in a content area. The input to this process is typically the knowledge of a subject matter expert. However, the use of textbooks, videos, or other learning materials is also possible. The source is then initially examined in terms of the recurring issues or problems described within it (referred to as the *problem*). This is the core of the cognitive model. Subsequently, *scenarios* (i.e., examples with different characteristics) are identified using the source, which are linked to the problem. In the third step of the AIG process, based on these scenarios, *sources of information* from the text, video, or other learning materials are extracted that can define the scenarios with their char-

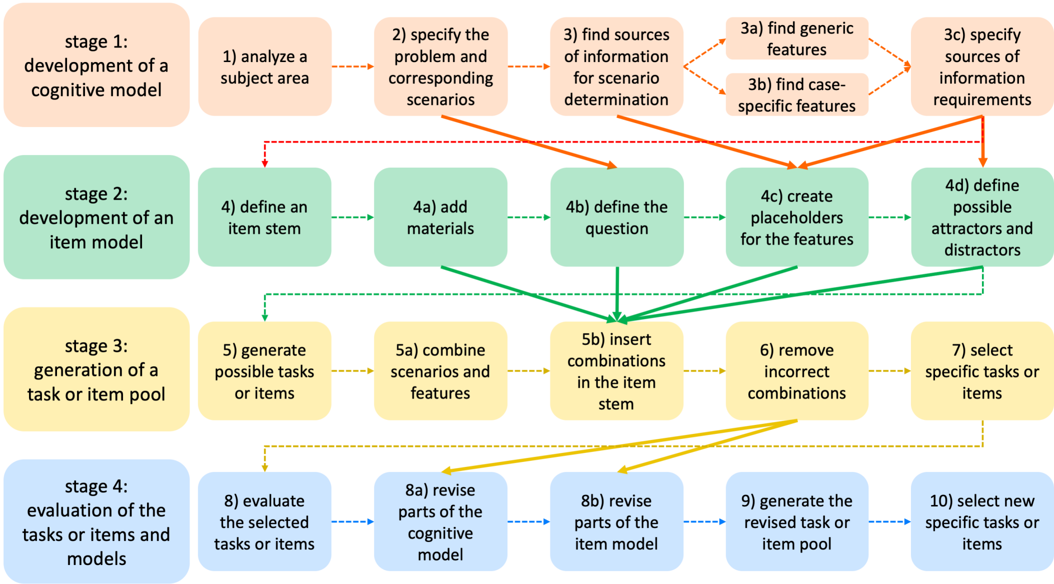

Figure 2: The stages of the traditional AIG process based on (Kosh et al., 2019) and (Gierl and Lai, 2016).

acteristics. A distinction is made between general and specific occurrences (i.e., *features*) of these sources of information. General occurrences of sources of information relate to all or at least a majority of the scenarios (i.e., *generic features*). Specific occurrences (i.e., *case-specific features*), on the other hand, are only related to particular scenarios, which must also be specified as a prerequisite in the cognitive model.

The following example related to different types of quadrilaterals (i.e., scenarios such as squares, rectangles, trapezoids, etc.) and their differentiation illustrates this idea in Figure 1. Initially, various mathematical textbook chapters were examined to identify sources of information typically related to these types of shapes, such as length of sides, perimeter and area, angles within the quadrilateral, and the parallelism of sides. Some of these sources of information are more restrictive than others. For example, the perimeter of a quadrilateral does not provide any information about its type. In contrast, having equal or unequal lengths of sides is one of several specific prerequisites for distinguishing between a square and a rectangle. However, the topic of determining the types of quadrilaterals is only an illustrative example here. Our research group has generated multiple cognitive models on topics such as biology, psychology, medicine, computer science, and mathematics, illustrating that AIG can handle a wide range of content areas.

2.2 Development of an Item Model

The item model serves as the link between the information from the cognitive model and the context in

Question template:	In geometry, a polygon that has four sides and four corners is called quadrilateral. A present quadrilateral has area of `<Area>` cm². Moreover, this quadrilateral has `<Sides>`. `<Parallelism>` parallel. The interior angles of the quadrilateral add up to 360 degrees. `<Angles>` equal. The colour of the quadrilateral is `<Colour>`. Furthermore, the quadrilateral is `<Symmetry>`. Lastly, it has two diagonals `<Diagonals1>`. `<Diagonals alt>` How is the present quadrilateral called?
Answer Template:	`<Quadrilateral>`

Figure 3: Example of an item model of the AIG Model Editor related to different types of quadrilaterals and their differentiation.

which the tasks or items are intended to be used. This means that in addition to the information in *item stem* and *question*, the item model also includes the item format, response options, item materials, feedback on solutions, and other components or information that are relevant or necessary to the context in which the tasks or items will be used (Gierl et al., 2012).

The main step of developing the item model is formulating the item stem, which is summarized in the fourth AIG step. For this purpose, a so-called *mother item* (i.e., a manually created task or item) can be used. If such a mother item does not exist, or if the existing tasks or items do not consider enough sources of information, a formulation must be found that includes all the necessary sources of information from the cognitive model and the corresponding question.

Subsequently, the general or specific occurrences of the sources of information in the item stem are replaced with placeholders that the item generator software will fill in order to generate the pool of tasks or items. Then, the task or item format is determined, and the feedback and the materials are included if needed. An example of an item model that was created with the AIG Model Editor from the TUD Dresden University of Technology (see Section 3) is illustrated in Figure 3. It shows the information from the cognitive model as placeholders (represented as $<$ ... $>$), the item stem (the description of the context as text), the question (*How...?*), and the distractors (also represented as $<$ ... $>$ in our editor).

When examining the item model, some sentences in the item stem may seem quite cryptic. To prevent learners from identifying the correct solution based on unusual or grammatically incorrect sentence constructions, it is possible to define word endings, punctuation marks, or even entire subordinate clauses as placeholder content. The development of the item model should be understood as an iterative process, characterized by multiple evaluation (see also stage 4 of AIG: *Evaluation of tasks or items and models*) and revision steps.

2.3 Generation of a Task or Item Pool

The task or item pool is generated by the item generator (in our case, the *AIG Model Editor* (AME) (Kucharski et al., 2023) developed at TUD Dresden University of Technology, see Section 3) through the alternation of all combinations of generic and case-specific features that do not violate any conditions defined in the cognitive model and that have a distinct solution.

In the above-mentioned example of quadrilaterals, out of 1728 theoretically possible combinations, only 81 combinations can result in items that have a distinct solution and do not violate any of the defined conditions. In Figure 4, a randomly selected item is illustrated. As can be seen, the AIG Model Editor has combined generic features (e.g., area, color, or sum of interior angles) that are irrelevant to the solution as well as case-specific features (e.g., side length, parallelism of sides, or type of diagonals) that are crucial for the solution.

Finally, to use the generated tasks or items for a specific use case, a selection must be made. If the goal is to use the tasks or items to support learning, then a selection would include tasks in the pool that cover as much of the knowledge about the subject as possible, even if only a few tasks are answered by a user (i.e., the tasks should be substantially different). If the goal is to use the tasks or items within a test scenario, the selection should contain as many items as

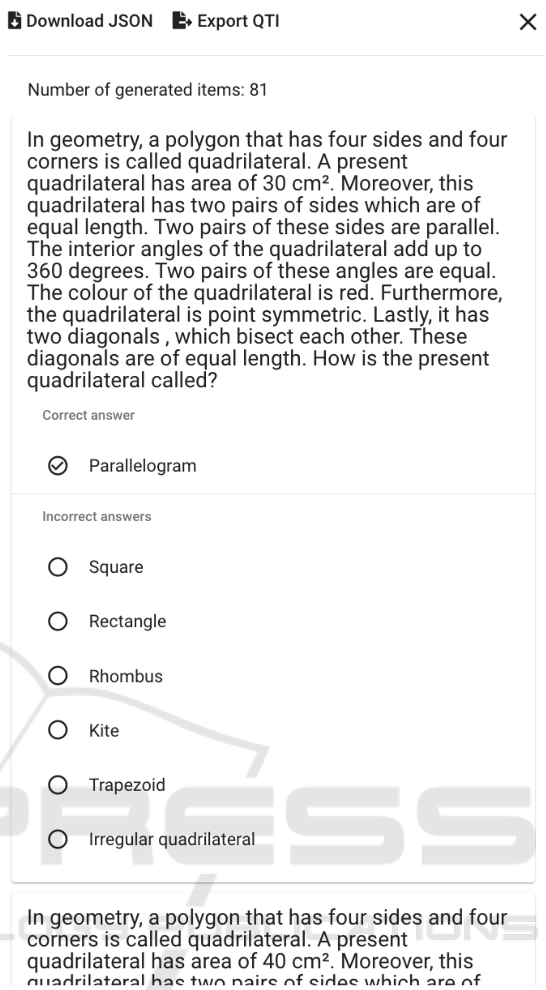

Figure 4: Example of one of the items generated by the AIG Model Editor from the above item model related to different types of quadrilaterals and their differentiation.

possible that are visually distinct from each other, but are comparable in terms of of difficulty. This ensures that each item must be answered individually, that students will require comparable knowledge to solve the test items, and that solutions cannot be copied or discussed among students. This selection of tasks or items can also be made automatically by entering specific parameters such as the number of tasks or items and their similarity. How this is done by the AIG Model Editor is described in Section 3.

2.4 Evaluation of the Tasks or Items and Models

In particular, if the provision of learning tasks or (self-)test items, the assessment, and the provision of feedback can be carried out in a computer-assisted manner (in terms of fully computer-based testing or assessment; see e.g., (Drasgow, 2016)), then it is advisable to also collect and analyze psychometric measures of the tasks or items (e.g., (Kosh et al., 2019)) automatically. This means that after the usage, characteristics such as the difficulty or the discrimination between tasks or items should be analyzed and compared with the predicted properties (Lienert and Raatz, 1998). For example, if it is observed that tasks or items derived from an item model are not approximately equally difficult (i.e., pseudo-parallel or even parallel items), or if tasks or items derived from a cognitive model do not discriminate between more and less successful learners (i.e., items with low discrimination), then the corresponding models from the AIG process should be revised.

It should be noted that these steps are optional and their requirements depend on the context in which the AIG tasks or items are used. For example, this approach is more necessary when building a test item pool for performance diagnostics than when tasks for supporting knowledge acquisition are to be offered to learners. It should also be noted that these steps do not only relate to AIG tasks or items. Even for manually created tasks or items, psychometric properties should be assessed when the context of task or item use requires it. These properties should then be used to guide the revision process.

2.5 Relation Between AIG and Adaptive Learning Approaches

Beyond the use of learning tasks and (self-)test items to encourage learners to actively process a subject area, student engagement and motivation is a focus of research related to Intelligent Tutoring Systems (ITS) as a means to provide customized tutoring (Alrakhawi et al., 2023) and research related to personalized learning in general (Ochukut et al., 2023).

With a history of more than 50 years, ITS aim to optimize the learning process in terms of various metrics by providing personalized instruction using intelligent functionalities and methods (Kurni et al., 2023). Over the years, a number of such intelligent functionalities and methods have been conceptualized, implemented, and evaluated (Mousavinasab et al., 2021). Some of these approaches also work with learning tasks or (self-)test items, such as (Pardos et al., 2023), which automatically generates problem steps for the learner to answer based on user-defined templates through variabilization, or (Yilmaz et al., 2022), which uses a variable number of items in Adaptive Mastery Tests (AMT) to test the learner's mastery of a particular subject and adjust the proposed learning path accordingly. In addition, research aimed at personalizing learning in general also uses test items, such as (Arsovic and Stefanovic, 2020), which uses pre-tests with relevant items prior to course study to identify prior knowledge and adapt course content and learning paths.

These few examples suggest that AIG, as a mechanism with the primary goal of generating materials to improve the learning process, and research related to ITS or personalized learning in general, with the primary goal of influencing the learning process itself to make it more effective, can not only be used side by side to achieve their respective goals more quickly, but can also benefit from each other conceptually by combining certain parts of both approaches.

3 AIG MODEL EDITOR

The *AIG Model Editor* (AME)[2], developed at TUD Dresden University of Technology, is the result of a collaboration between the departments of Computer Science and Psychology. Its origins can be traced back to a student project jointly supervised by these departments (Baum et al., 2021; Braun et al., 2022). After initial test runs, it was fundamentally revised and then continuously evaluated and optimized in multiple iteration loops. The current state of development is described in detail in (Kucharski et al., 2023) and can be explored at https://ame.aig4all.org.

3.1 General Structure

The latest version of the editor is illustrated in Figure 5, it consists of different visually separated areas. On the upper left side, there is a graphical representation of the cognitive model. In the center, the problem and its associated scenarios are visually highlighted in a box with a green border (i.e., *quadrilaterals* with *square*, *rectangle*, *trapezoid*, etc.). All case-specific features of the sources of information are linked to the scenarios in terms of their prerequisites. On the lower left side, all sources of information are defined with their generic and case-specific features (e.g., the source of information *symmetry* with *axis symmetry*, *point symmetry*, and *axis and point symmetry*). The clear structure and several easy-to-use modeling functionalities, such as the ability to drag and drop sources of information into the cognitive model, support the users during the first stage of the AIG process (see Section 2.1). The second stage (see Section 2.2) is supported on the right. There

[2]https://ame.aig4all.org, accessed February 13, 2024

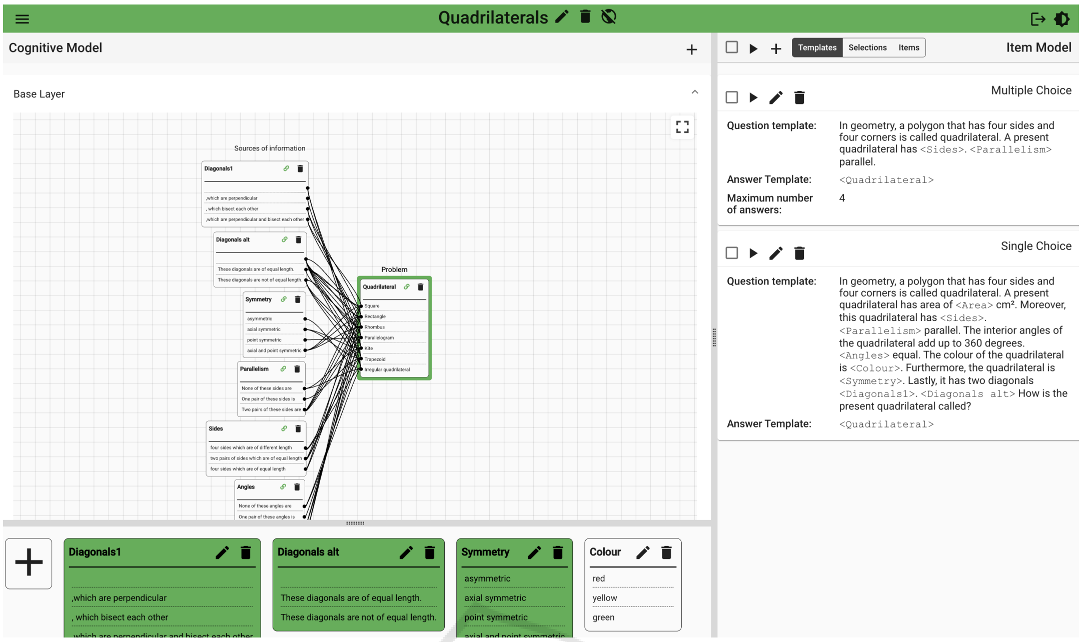

Figure 5: The latest version of the AIG Model Editor.

different item models can be defined with their item stems, questions, formats, and distractors. Pressing the *Play* button triggers the generation included in the third stage (see Section 2.3). After completion, an additional window opens showing all generated tasks or items with their distinct solutions for the corresponding cognitive model. From there, they can be exported to different output formats for import into learning platforms such as Moodle[3] or Audience Response Systems (ARS) such as AMCS[4] (Braun et al., 2018).

3.2 Features

The AIG Model Editor has two unique features compared to the small number of other editors that have been developed for the implementation of the AIG approach. First, a cognitive model created in this editor can contain multiple layers. Second, the editor provides the ability to perform a rule-based selection of AIG tasks or items to assist the user in the seventh step in the third stage (see Section 2.3).

Figure 1 shows the so-called *base layer* of an example of a cognitive model about the above-mentioned different types of quadrilaterals and their differentiation. The base layer of a cognitive model contains the simple relationships between features and scenarios that do not require loops or if-then propositions. However, to represent certain subject matters, such more complex conditional constructs are required to represent the necessary relationships. For this purpose, the AIG Model Editor provides the ability to define *condition layers*. These layers allow to elaborate conditions and conclusions between different sources of information and their features. In summary, the AIG Model Editor allows the generation of more complex learning tasks and (self-)test items than other editors.

To choose a subset of generated tasks and items, the AIG Model Editor provides the ability to automatically determine a random or rule-based selection. By having selected *Selections* and using the nearby button with the *plus* sign (see Figure 5), a dialogue window opens where various features of the selection can be adjusted. For example, tasks or items can be selected that differ from each other either extensively or only superficially. This option reduces the number of steps required after the generation and also allows to make a large set of generated items manageable again.

3.3 Evaluation Results

As previously mentioned, the AIG Model Editor was evaluated and revised multiple times. For the evaluation of the latest version of the editor, 12 participants were asked to create a textually described model in the editor, generate some items, and then share their experiences using a questionnaire. In order to ensure that persons with a background and without a background in computer science participated in that exper-

[3]https://moodle.com, accessed February 13, 2024
[4]https://amcs.website, accessed February 13, 2024

iment, persons from different subject domains were asked to join. Furthermore, also some people were included who had never created learning tasks or test items before. All participants were able to successfully complete the assigned task, and the overall feedback was consistently positive.

To quantitatively determine the user-friendliness of the developed editor, the *System Usability Scale* - SUS (Brooke, 1996) was used as a reference. Participants were asked to agree or disagree with a series of predefined negative and positive statements regarding the usability of the developed editor using a 5-point Likert scale. On average, the SUS score was 81. While the threshold for a user-friendly system is 68, the evaluation thus indicated that the average of the surveyed participants found the editor to be well usable. Moreover, as indicated by their statements, participants also understood the AIG use case.

4 EVALUATION OF GENERATED ITEMS

In order to assess the quality of items generated automatically using the AIG Model Editor, students were given such items alongside manually created items. In addition, they received an evaluation questionnaire containing eight statements. Five of the eight statements (e.g., *"In order to solve the item, knowledge in the area of '...' have to be applied."* or *"The correct option cannot be identified through grammatical peculiarities or unfamiliar phrasing."*) were based on the evaluation questionnaire by (Gierl and Lai, 2013a), which the authors also used to evaluate AIG items and manually created items. In contrast, three of the eight statements (e.g., *"The item is formulated simply and clearly."*) were based on the *Hamburger Verständlichkeitsmodell* (Langer, von Thun and Tausch, 2019) to compare the tasks and items based on general criteria of clarity. All statements had a six-point rating scale ranging from *strongly disagree* to *strongly agree*.

The questionnaire has been used in two rounds of evaluation. However, between these two rounds, the content of the items, the students who evaluated them, and the people who generated and created the items differed. This approach was chosen in order to ensure that the results could be interpreted independently from the specific content or the individuals involved.

4.1 Evaluation One

The first evaluation took place in February 2023. Twenty-four students attending a lecture in the field of computer science were presented with a total of eight items on the topic of *Service and Cloud Computing*. Five out of the eight items were generated using AIG, and three out of the eight items were manually created by experts in the field. The students were unaware of which items were automatically generated and which were manually created and were given the aforementioned evaluation questionnaire. For each item, they rated the eight statements. The evaluation results were then aggregated per item and method (i.e., automatically generated or manually created). Table 1 shows the results of the first evaluation.

Table 1: Results of the first evaluation.

Item	AIG	Manual	MV	SD
1	x		4.60	0.55
2		x	4.59	0.59
3	x		4.98	0.72
4	x		4.96	0.64
5	x		4.96	0.65
6		x	4.79	0.46
7		x	5.07	0.71
8	x		4.75	0.58
Total	4.85	4.81		

Firstly, the results indicate that both methods have led to good items, as the students fairly agreed with all quality criteria for each individual item (approximately 4.8 out of a maximum of 6 points, or in other words, they selected the response *agree with the criterion* most frequently). Furthermore, it becomes clear that AIG items hardly differ in their quality from manually created items, or that the students could not detect any quality differences between the items. An analysis using a t-test also did not find any differences between the results of the automatic item generation and the manual item creation.

4.2 Evaluation Two

The second evaluation took place in July 2023. This time, a total of eight items on the topic of *Computer Networks* were presented to 90 students in a computer science lecture. For this evaluation, four items were generated using AIG and four items were created manually. The students were given the same evaluation questionnaire described above. Once again, they could not distinguish which items were automatically generated and which were manually created. Table 2 shows the results of the second evaluation.

Table 2: Results of the second evaluation.

Item	AIG	Manual	MV	SD
1		x	5.09	0.67
2	x		4.96	0.71
3	x		4.70	0.89
4		x	5.15	0.66
5	x		4.98	0.69
6		x	4.86	0.77
7		x	4.94	0.63
8	x		4.99	0.72
Total	4.91	5.01		

The results are almost identical to the first evaluation, although the content of the items, the students asked to evaluate the items, and the people who generated or created the items were different from the first evaluation. Once again, the students positively assessed each individual item, regardless of whether it was generated using AIG or created in a traditional manner. Again, an analysis using a t-test did not find any differences between the results of the automatic item generation and the manual item creation.

5 DISCUSSION

The results of the evaluations can be discussed on different levels, such as the quality of the items, the quantity of the items, or the simplicity of the process to the final items (see also the review on AIG items in the field of medicine by (Falcão et al., 2022)). In terms of clarity, these discussion points are addressed separately below, although some aspects may overlap. We hypothesize that the evaluation results also apply to learning tasks and plan to test this hypothesis in the future.

5.1 Quality of the Items

Both evaluations have shown that items generated through AIG are of high quality and comparable to manually created items. However, when considering that the AIG process can be significantly more time and resource efficient than manual item creation, the value of AIG becomes clear. This is especially true when items are needed regularly or in large quantities. This significant advantage of AIG is in line with the findings of other research groups. Until above mentioned evaluations, our own research group has reported this result anecdotally. The results presented in this paper now empirically confirm this view.

5.2 Quantity of the Items

It should be noted that once concepts are incorporated into a cognitive model, they can be used repeatedly for new tasks and items. This aspect further increases the difference between the number of tasks and items generated by the AIG approach and those created manually. Thus, while the evaluations compared, for example, four AIG items against four manually created items, in reality, the four AIG items were just a random selection from several hundred items that could have been generated based on the predefined cognitive model, as shown in the following Table 3.

Table 3: Number of items per subject matter.

# Evaluation/# Item/Subject Matter	# Items
1 / 1 / Web Services Extensions	620
1 / 3 / Vertical Web Service Scaling	5
1 / 4 / Security goals and encryption methods	184
1 / 4 / Application encryption methods	12
1 / 8 / Web Service Scaling	1064
2 / 2 / Network Technologies	350
2 / 3 / Behavior of Data Transfer Protocols	2
2 / 5 / Addressing in Computer Networks	385
2 / 8 / TCP/IP Properties	24
Total	2646

5.3 Ease of Providing Learners with Items

Another advantage of the AIG approach and the developed editor is that even participants with little experience in creating learning tasks or test items can easily create them for a specific subject area. In contrast to manual item creation, where lack of knowledge or skills is a major hindrance to task or item creation (Li et al., 2021), AIG requires minimal knowledge of the task or item creation process since this process is guided continuously by the sequence of steps predetermined by AIG (Gierl and Lai, 2016; Damnik et al., 2018; Kosh et al., 2019) and the editor (Kucharski et al., 2023). The editor supports not only the generation of a large number of tasks or items in a short time compared to manual creation but also the structuring of learning content and the related knowledge, which is a prerequisite for both manual item creation and automatic item generation. The fact that 66.7% of the participants fully agree and 33.3%

agree with the statement *"I understand the concept of AIG and know what corresponding software can be used for.*", indicates that this understanding of the AIG process was also conveyed to the participants through their work with the editor.

6 FUTURE WORK

Currently, the AIG Model Editor only generates tasks and items through the alternation of all valid combinations of generic and case-specific features (see Section 2.3). Others proposed to use large language models (Sayin and Gierl, 2024; Kıyak, 2023). This approach allows to generate tasks and items whose wording and content is not explicitly given as generation input. But like the AIG Model Editor, it still requires describing the topic and the form of target tasks and items, and evaluating the generation result.

In further development of the AIG Model Editor, large language models could be used for the generation, similar to the other works. Besides, it is believed that the integration of artificial intelligence into the modeling process would further reduce the cognitive effort required during modeling. Therefore, it is currently examined in which steps and how users could best be supported by this new technology. At the moment, analyzing the subject area, creating the cognitive model, and formulating item stems seem to be possible candidates.

REFERENCES

Alrakhawi, H. A., Jamiat, N., & Abu-Naser, S. S. (2023). Intelligent Tutoring Systems In Education: A Systematic Review Of Usage, Tools, Effects And Evaluation. Journal of Theoretical and Applied Information Technology. 101. 1205 - 1226.

Arsovic, B., & Stefanovic, N. (2020). E-learning based on the adaptive learning model: case study in Serbia. Sādhanā, 45(1), 266.

Baum, H., Damnik, G., Gierl, M. & Braun, I. (2021). A Shift in automatic Item Generation towards more complex Tasks, INTED2021 Proceedings, pp. 3235-3241.

Braun, I., Kapp, F., Hara, T., Kubica, T., & Schill, A. (2018). AMCS (Auditorium Mobile Classroom Service)–an ARS with Learning Questions, Push Notifications, and extensive Means of Evaluation. In CEUR Workshop Proceedings (Vol. 2092).

Braun, I., Damnik, G., & Baum, H. (2022). Rethinking Assessment with Automatic Item Generation, Inted2022 Proceedings, pp. 1176-1180.

Brooke, J. (1996). SUS: a "quick and dirty" usability scale. In P. Jordan, B. Thomas, & B. Weerdmeester (Eds.), Usability Evaluation in Industry (pp. 189–194). London, UK: Taylor & Francis.

Chi, M., & Boucher, N. (2023). Applying the ICAP Framework to Improve Classroom Learning. In C. E. Overson, C. M. Hakala, L. L. Kordonowy, & V. A. Benassi (Eds.), In their own words: What scholars and teachers want you to know about why and how to apply the science of learning in your academic setting (pp. 94-110). Society for the Teaching of Psychology.

Chi, M., & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. Educational psychologist, 49(4), 219-243.

Damnik, G., Gierl, M., Proske, A., Körndle, H., & Narciss, S. (2018). Automatic Item Generation as a Means to Increase Interactivity and Adaptivity in Digital Learning Resources [Automatische Erzeugung von Aufgaben als Mittel zur Erhöhung von Interaktivität und Adaptivität in digitalen Lernressourcen]. In E-Learning Symposium 2018 (pp. 5-16). Universitätsverlag Potsdam.

Drasgow, F. (2016). Technology and testing: Improving educational and psychological measurement. New York: Routledge.

Embretson, S. E. (2002). Generating abstract reasoning items with cognitive theory. In S. H. Irvine and P. C. Kyllonen (Eds.), Item generation for test development (pp. 219–250). Mahwah, NJ: Lawrence Erlbaum Associates.

Embretson, S. E., & Yang, X. (2007). Automatic item generation and cognitive psychology. In C. R. Rao & S. Sinharay (Eds.), Handbook of statistics: Psychometrics, Volume 26 (pp. 747–768). Amsterdam, The Netherlands: Elsevier.

Falcão, F., Costa, P., & Pêgo, J. M. (2022). Feasibility assurance: a review of automatic item generation in medical assessment. Advances in Health Sciences Education, 27(2), 405-425.

Gierl, M. J., & Lai, H. (2013a). Evaluating the quality of medical multiple-choice items created with automated processes. Medical education, 47(7), 726-733.

Gierl, M. J., & Lai, H. (2013b). Instructional topics in educational measurement (ITEMS) module: Using automated processes to generate test items. Educational Measurement: Issues and Practice, 32(3), 36-50.

Gierl, M. J., & Lai, H. (2013c). Using weak and strong theory to create item models for automatic item generation: Some practical guidelines with examples. In M. J. Gierl & T. Haladyna (Eds.), Automatic item generation: Theory and practice (pp. 26–39). New York, NY: Routledge.

Gierl, M. J., & Lai, H. (2016). Automatic item generation. In S. Lane, M. R. Raymond, & T.M. Haladyna (Eds.), Handbook of test development (2nd ed., pp. 410–429). New York, NY: Routledge.

Gierl, M. J., Lai, H., & Turner, S. (2012). Using automatic item generation to create multiple-choice items for assessments in medical education. Medical Education,46, 757–765.

Kapp, F., Proske, A., Narciss, S., & Körndle, H. (2015). Distributing vs. blocking learning questions in a web-based learning environment. Journal of Educational Computing Research, 51(4), 397-416.

Kerres, M. (2002). Combining Online and Onsite Elements in Hybrid Learning Settings [Online- und Präsenzelemente in hybriden Lernarrangements kombinieren]. In A. Hohenstein & K. Wilbers (Hrsg.), Handbuch E-Learning (S. 1-19). Köln: Deutscher Wirtschaftsdienst.

Kıyak, Y. S. (2023). A ChatGPT prompt for writing case-based multiple-choice questions. Revista Española de Educación Médica, 4(3).

Kosh, A. E., Simpson, M. A., Bickel, L., Kellogg, M., & Sanford-Moore, E. (2019). A cost–benefit analysis of automatic item generation. Educational Measurement: Issues and Practice, 38(1), 48-53.

Körndle, H., Narciss, S., Proske, A., 2004. Creation of Interactive Learning Tasks for University Teaching [Konstruktion interaktiver Lernaufgaben für die universitäre Lehre]. In D. Carstensen & B. Barrios (Eds.), Campus 2004. Kommen die digitalen Medien an den Hochschulen in die Jahre? (pp. 57-67). Münster: Waxmann.

Kucharski, S., Damnik, G., Stahr, F., & Braun, I. (2023). Revision of the AIG Software Toolkit: A Contribute to More User Friendliness and Algorithmic Efficiency. In J. Jovanovic, I.-A. Chounta, J. Uhomoibhi, & B. McLaren: Proceedings of the 15th International Conference on Computer Supported Education - Volume 2: CSEDU. SciTePress, pages 410-417.

Kurni, M., Mohammed, M. S., & Srinivasa, K. G. (2023). Intelligent tutoring systems. In A beginner's guide to introduce artificial intelligence in teaching and learning (pp. 29-44). Cham: Springer International Publishing.

Langer, I., von Thun, F. S., & Tausch, R. (2019). Express Yourself Clearly [Sich verständlich ausdrücken]. München: Ernst Reinhardt Verlag.

Li, J., Pilz, M., & Gronowski, C. (2021). Learning Tasks in Higher Education Pedagogy: An Investigation Into the Use of Learning Tasks [Lernaufgaben in der Hochschuldidaktik: Eine Untersuchung zum Einsatz von Lernaufgaben]. Bildung und Erziehung, 74(1), 31-50.

Lienert, G., & Raatz, U. (1998). Test Construction and Test Analysis [Testaufbau und Testanalyse]. Weinheim: Beltz Psychologie Verlags Union.

Mousavinasab, E., Zarifsanaiey, N., R. Niakan Kalhori, S., Rakhshan, M., Keikha, L., & Ghazi Saeedi, M. (2021). Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. Interactive Learning Environments, 29(1), 142-163.

Ochukut, S. A., Oboko, R. O., Miriti, E., & Maina, E. (2023). Research Trends in Adaptive Online Learning: Systematic Literature Review (2011–2020). Technology, Knowledge and Learning, 28(2), 431-448.

Pardos, Z. A., Tang, M., Anastasopoulos, I., Sheel, S. K., & Zhang, E. (2023, April). Oatutor: An open-source adaptive tutoring system and curated content library for learning sciences research. In Proceedings of the 2023 chi conference on human factors in computing systems (pp. 1-17).

Pugh, D., De Champlain, A., Gierl, M., Lai, H., & Touchie, C. (2020). Can automated item generation be used to develop high quality MCQs that assess application of knowledge? Research and Practice in Technology Enhanced Learning, 15, 12 (2020).

Sayin, A., & Gierl, M. (2024). Using OpenAI GPT to Generate Reading Comprehension Items. Educational Measurement: Issues and Practice.

Shappell, E., Podolej, G., Ahn, J., Tekian, A., Park, Y. (2021). Notes From the Field: Automatic Item Generation, Standard Setting, and Learner Performance in Mastery Multiple-Choice Tests. Evaluation & the Health Professions, 44(3), 315-318.

Yilmaz, R., Yurdugül, H., Yilmaz, F. G. K., Şahiṅ, M., Sulak, S., Aydin, F., ... & Ömer, O. R. A. L. (2022). Smart MOOC integrated with intelligent tutoring: A system architecture and framework model proposal. Computers and Education: Artificial Intelligence, 3, 100092.