Banking Strategies and Software Solutions for Generated Test Items

Tahereh Firoozi[1][a] and Mark J. Gierl[2][b]
[1]School of Dentistry, Faculty of Medicine and Dentistry, University of Alberta, Edmonton, Alberta, Canada
[2]Measurement, Evaluation, and Data Alberta, Edmonton, Alberta, Canada
{tahereh.firoozi, mark.gierl}@ualberta.ca

Keywords: Automatic Item Generation, Item Banking, Content Coding.

Abstract: Automatic item generation is a scalable item development approach that can be used to produce large numbers of test items. Banking strategies and software solutions are required to organize and manage large numbers of generated items. The purpose of our paper is to describe an organizational structure and different management strategies that rely on appending items with descriptive information using content codes. Content codes contain descriptive data that can be used to identify and differentiate generated items. We present a modern approach to banking that allows the generated items to be managed at both the item and model level. We also demonstrate how a content coding system at both the item and model level provides the user with the ability to execute many different types of searches including accessing one content-specific item from one model, multiple content-specific items from one model, one content-specific item from multiple models, and multiple content-specific items from multiple models. These examples help demonstrate that content coding is a fundamental concept that must be implemented when attempting to organize and manage generated test items.

1 INTRODUCTION

Automatic item generation (AIG) is a research area where cognitive theories, psychometric practices, and computational methods guide the creation of items that are produced with computer-assisted processes. AIG is an important educational testing innovation that can be used to overcome the scalability problem inherent to the traditional item development approach where subject-matter experts (SMEs) produce items sequentially on an individual basis. Gierl et al., (2021) described a three-step method for template-based AIG. The content for item generation is identified in step 1. SMEs identify and structure the content necessary to generate new test items using a cognitive model. A cognitive model is used to identify the knowledge and skills required by the examinee to solve a test item in a designated content area. An item model is created in step 2. An item specifies where the content from the cognitive model must be placed in a predefined rendering to generate new items thereby functioning as a blueprint for the item production process. The item model is a template (hence the phrase template-based AIG) because it identifies which parts of the test item can be systematically manipulated and which parts remain static when producing new test items. Computer-based algorithms then place the content from the cognitive model into the item model as part of step 3. Content assembly is a combinatorial task conducted with a computer algorithm that identifies content combinations that meet the constraints defined in the cognitive model in order to produce plausible test items. The outcome from step 3 is the production of plausible and coherent test items aligned with the parameters and constraints initially specified in the cognitive model.

Template-based AIG relies on the coordinated activities and outcomes produced by human expertise and computer technology. SMEs create cognitive and item models to structure the content required to create new items. Computer programs then implement algorithms to amalgamate the content in the cognitive and item models using constraints identified by the SMEs specifying which content combinations are feasible. A distinctive and critical feature of template-based AIG is that it relies on the model as the unit of

[a] https://orcid.org/0000-0002-6947-0516
[b] https://orcid.org/0000-0002-2653-1761

Firoozi, T. and Gierl, M. J.
Banking Strategies and Software Solutions for Generated Test Items.
DOI: 10.5220/0013491000003932

In *Proceedings of the 17th International Conference on Computer Supported Education (CSEDU 2025) - Volume 1*, pages 779-784
ISBN: 978-989-758-746-7; ISSN: 2184-5026

analysis. This shift in the unit of analysis means that the model is used to generate many items compared with the traditional item development approach, where each item is written individually by one SME. This important shift also means that the number of items is not dependent on the number of SMEs. Instead, item development is linked to the number of available models, where one SME can create a small number of models that produce large numbers of items thereby scaling the item development process.

The purpose of our paper is to describe and illustrate strategies that rely on appending items with descriptive information using content codes so the items can be identified and differentiated in a bank. Strategies that rely on content coding allow the user to monitor, organize, and manage the generated items in his, her, or their bank.

2 AIG AND CONTENT CODING

AIG is a scalable item development approach capable of producing large numbers of test items when correctly implemented. However, this influx of new items must be organized and managed if this resource is to be used effectively (Cole et al., 2020; Lane et al., 2016). One strategy for organizing items is to append each item with psychometric data (e.g., item difficulty) so the item can be identified and differentiated from other items in the bank. Typically, generated items do not contain psychometric data because the item sets are large. As an alternative, content coding is a method that can be used to describe generated items. Two different methods can be used to describe items and models using content codes. The first method relies on *posthoc* coding. With this method, content codes are created by reviewing the items and the models in order to identify relevant content descriptors. The advantage of posthoc content coding is its flexibility. This method can be used to create new and novel content coding systems for any item and model combination because it does not require an existing content coding system or taxonomy. However, this method often suffers from a lack of specificity within a code, it is often inconsistently applied across codes because the content descriptions tend to be overly general, and because of this generality, the codes are often difficult to interpret (Gierl et al, 2022). In addition, posthoc coding is time-consuming to implement, particularly if large numbers of items and models are created and, hence, need appended content codes after generation.

The second method relies on predefined (also called *a priori*) coding. With this method, existing content codes are drawn from a taxonomy or established coding system and then applied to the items and models as the content descriptors. Predefined coding uses existing content codes derived from established taxonomies or coding systems thereby providing specific, consistent, and interpretable descriptors. In addition, a content coding taxonomy often contains data descriptions that are related to one another because of their position with other data descriptors in a hierarchy or system (Gartner, 2016). As a result, the content codes in one system or taxonomy can be linked to other content codes and data descriptors both in the existing system or taxonomy and to other related systems and taxonomies. The disadvantage of predefined content coding is that this method is prescriptive meaning that coding is limited to the content in the taxonomy and, hence, inflexible. Predefined content coding also relies on the availability of an appropriate taxonomy to describe the items and models. This type of taxonomy is not available in some content areas.

One important benefit of using template-based AIG is that the item model created in step 2 encompasses all of the content that is required for generating items. Hence, the item model includes the content that will be used to generate all of the items specified in the cognitive model. Because content coding is included in the item modelling step, the content codes can also be integrated into the items using the same assembly logic. In other words, content codes can be used to create data to describe the generated items. Content codes can be added at three different levels in the item model. The first level is option-level coding. Option-level coding describes data in the multiple-choice response options or alternatives. Content coding, therefore, can be used to describe the item options when generating the selected-response item type. The second level is item-level coding. A cognitive model contains variables and values. Variables are elements in an item model that describe a particular outcome. Variables contain values that will be manipulated during item generation to create new items. Content coding can be used to describe the generated items at both the variable and value level. These variables and values can be denoted as string or integer values. The third level is model-level coding. Content codes can be applied at the model level. With model-level coding, specific codes that describe all of the generated items from a particular item model are used.

Content codes are implemented in the template-based AIG workflow during step 2. Because the unit of analysis has shifted from the item to the model, content coding is a straightforward process when predefined codes are available. Codes are placed into the item model in step 2 using both the item (i.e., variable, value, and/or options) and model-level (e.g., content classification) coding, and these codes, in turn, are included with each generated item in step 3. Recall that a combinatoric method is used in step 3 to assemble all permissible combinations of content specified in the cognitive model. The combinatoric method results in a new set of generated items. Because the content codes are included in the item model, the content coding list associated with each generated item is also produced using the same combinatoric method to create the generated items. Different item and model-level content is used to create the items and, hence, a different set of content codes will be used to describe each generated item. As a result, the content codes for each generated item will be unique and different from one another. In other words, the content codes produce a descriptive data list for each generated item in step 3 that includes all of the item and model codes as part of the item modelling process in step 2. Multiple content codes at different locations in the item and model provide data that can be used to describe each generated item uniquely.

2.1 Requirements and Applications of a Bank Structured Using Content Codes

Next, we describe and demonstrate how to create and access content from a bank specifically designed for generated items. The content codes structure the items in the bank. Hence, the bank's requests are entirely driven by item and model content codes. A content coding system at both the item and model level provides the user with the ability to execute many different types of searchers including accessing one content-specific item from one model, multiple content-specific items from one model, one content-specific item from multiple models, and multiple content-specific items from multiple models.

2.1.1 One Content-Specific Item from One Model

The bank contains a list of models. The models can be filtered and searched by text or can be sorted and searched by any of the model categories, including the date the model was placed into the bank, the title of the model, a description of the model and item, and the generation capacity of the model.

To access the items, the user must begin by entering one content code from a drop-down list of available content contents that describes the items and the models in the bank, as shown in Figure 1.

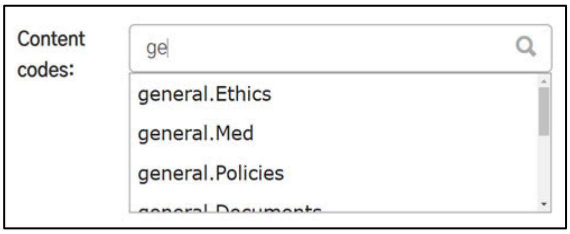

Figure 1. Content code lookup table.

Next, the user must specify the exact number of items that will be requested from the model. In the Figure 2 example, the goal is to identify 10 items that contain one specific content code from one specific model. Since only one model is being used for the item request, the number in the "Total" box (bottom) will equal the requested number of items (top). Selection of the requested items occurs in one of two ways. If the user requests, for example, 10 items and 25 items are available, then 10 items from the set of 25 items are selected randomly. Alternatively, if the user requests 10 items and 5 items are available, then all 5 items are accessed and presented to the user (see Section 2.2 for details on the Item Selection Process).

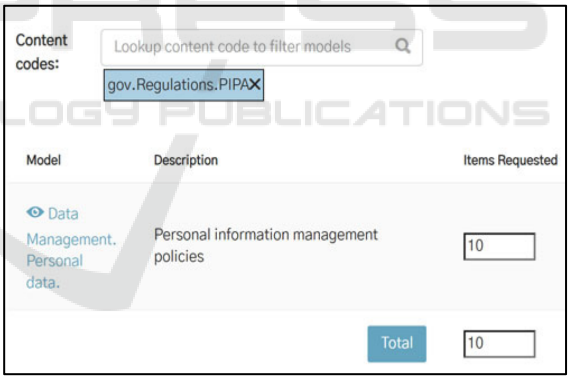

Figure 2. One content code from one model.

2.1.2 Multiple Content-Specific Items from One Model

To access multiple content-specific items from one model, the user must enter multiple content codes (see Figure 3). Then, the user must specify the exact item request from the model. In Figure 3, two content codes are specified from one model for a 10-item request.

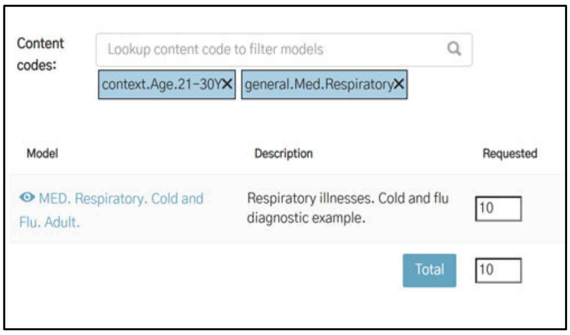

Figure 3. Multiple content codes from one model.

2.1.3 One Content-Specific Item from Multiple Models

The previous two examples focus on item requests from a single model. In operational testing situations with large numbers of models, item requests are typically conducted across multiple models. For example, the user can specify one content code in order to access items across multiple models. The search can be conducted using either a subset of the models in the bank or all of the models. In the example presented in Figure 4, the goal is to select 30 items using one content code which can be accessed from three different models. The user also specifies the exact number of items that will be requested from each model (which, in this example, is 10 from three models for a total of 30 items). Hence, the task is to identify the items that meet the content-coding requirement and then draw a specific number of items that meet these requirements.

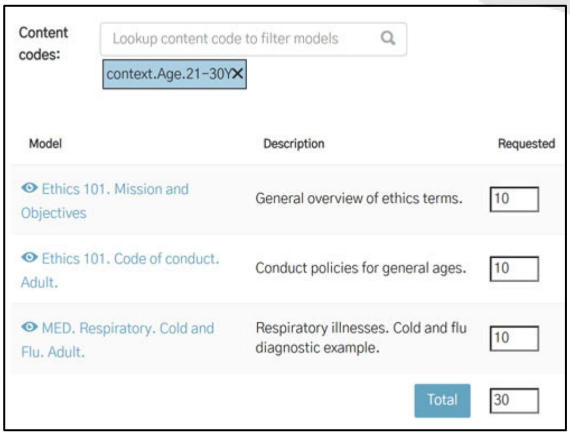

Figure 4. One content code from three models.

2.1.4 Multiple Content-Specific Items from Multiple Models

The final scenario includes items with multiple content codes that must accessed from multiple models. This scenario is common when a large number of models are available. This scenario also helps ensure that the user is identifying and accessing diverse items from the bank because the items must satisfy multiple content coding requirements and these requirements must occur across multiple models. In the example shown in Figure 5, the user lists two content codes that must be satisfied across two models, where the model 1 request includes 5 items and the model 2 request includes 10 items for a total of 15 items.

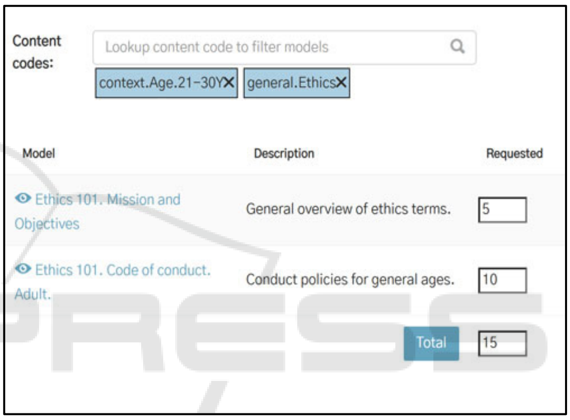

Figure 5. Multiple content codes from multiple models.

2.2 Selecting Items Using Content Codes

We have described and illustrated how a content coding system at the item and model level provides the user with the ability to access items using different numbers of content codes across different numbers of models. The items that meet the content coding specification are presented to the user as a list. The curated items in the list can then be reviewed. The user can select a subset of items from the list, as shown in Figure 6, using a checkbox. The user can also select the entire list of items. The items are then exported into a content management system.

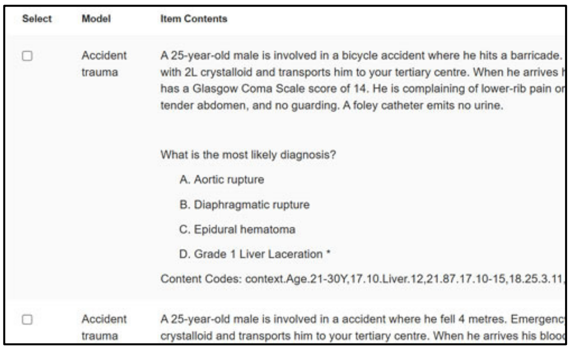

Figure 6. Selecting items from content codes.

The exported items have three key properties that allow users to monitor, organize, and manage the generated items exacted from the bank. First, the items from the bank are tracked so the user can easily see how many items have been extracted from a model. Second, the bank tracks the exact items that have been exported thereby preventing the same items from being selected as part of a future item request. Third, the items can be exported in a wide variety of formats (e.g., QTI, XLS) thereby ensuring the exported items are compatible with the requirements of any content management system.

3 CONCLUSIONS

AIG is an item development method that leverages cognitive theories, psychometric practices, and computational methods to produce test items with the assistance of computer technology. AIG can be used to create large numbers of items. Large item banks are needed to support testing innovation (Chen et al., 2022; Gordillo-Tenorio, 2023; OECD, 2024) and to enhance test security (Gierl et al., 2022). However, the generation of an abundant supply of items presents significant challenges in organizing and managing these resources. This issue is particularly pertinent for testing organizations that have traditionally handled relatively small numbers of items, as managing a bank containing millions of AIG-generated items introduces complexities far beyond those associated with maintaining hundreds of SME-created items.

A traditional item bank is a repository for organizing and managing information on each item. The maintenance task focuses on item-level information. However, with AIG, models rather than items serve as the unit of analysis. Testing organizations are familiar with developing, organizing, and managing items. But AIG creates another new challenge that many organizations have not experienced because the models must first be developed and then organized and managed. Hence, testing organizations must organize and manage the generating models in addition to the generated items. The purpose of our paper was to describe strategies that rely on appending items with descriptive information using content codes so the items can be identified and differentiated in a bank. We presented and illustrated a modern approach to banking that allows the generated items to be organized and managed at the model level in the bank. An AIG bank is an electronic repository for organizing and managing information on both the item and model using content codes. Because the model serves as the unit of analysis, the banks contain information on every model as well as every item.

The strategies we present are fundamentally dependent on content coding, a method used to describe data systematically. These descriptive codes must be applied to both items and models to effectively describe and locate specific generated items within the bank. Content codes can be developed *posthoc* by the user or accessed *a priori* through existing content coding systems. We demonstrate various applications of content coding, including how a single content code can locate ten generated items from one model (see Section 2.1.1), how two content codes can identify ten items from one model (see Section 2.1.2), how one content code can retrieve thirty items from three models (see Section 2.1.3), and how two content codes can locate fifteen items from two models (see Section 2.1.4). These examples illustrate that content coding is a robust method for appending descriptive data to generated items, thereby enabling the efficient and effective organization and retrieval of specific items from a bank.

AIG represents an important educational testing innovation that addresses the scalability issues inherent in traditional item development approaches. Content coding AIG items at the item and model level is an equally important innovation that must be implemented when attempting to organize and manage an abundant new supply of generated test items. This two-level content coding approach facilitates the maintenance of large item banks and ensures that research developments in AIG can be readily translated into operational testing practices.

REFERENCES

Chen, X.; Zou, D., Xie; H., Cheng, G., & Liu. C (2022). Two decades of artificial intelligence in education: Contributors, collaborations, research topics, challenges, and future directions. *Educational Technology & Society 25,* 28-47.

Cole, B. S., Lima-Walton, E., Brunnert, K., Vesey, W. B., & Raha, K. (2020). Taming the firehose: Unsupervised machine learning for syntactic partitioning of large volumes of automatically generated items to assist automated test assembly. *Journal of Applied Testing Technology, 21,* 1-11.

Gartner, R. (2016). *Metadata: Shaping knowledge from antiquity to the semantic web.* New York: Springer.

Gierl, M. J., & Haladyna, T. (2013). *Automatic Item Generation: Theory and Practice.* New York: Routledge.

Gierl, M. J., Lai, H., & Tanygin, V. (2021). *Advanced Methods in Automatic Item Generation.* New York: Routledge.

Gierl, M. J., Shin, J., Firoozi, T., & Lai, H. (2022). Using content coding and automatic item generation to improve test security. *Frontiers in Education* (Special Issue: Online Assessment for Humans—Advancements, Challenges, and Futures for Digital Assessment). 07:853578. doi: 10.3389/feduc.2022.853578

Gordillo-Tenorio, W. (2023). Information technologies that help improve academic performance, a review of the literature. *International Journal of Emerging Technologies in Learning, 18,* 262-279.

Lane, S., Raymond, M., & Haladyna, T. (2016.). *Handbook of Test Development* (2nd edition). New York: Routledge.

OECD iLibrary. (2024). *OECD Digital Economy Outlook 2024 (Volume 1)*. OECD. https://doi.org/10.1787/a1689dc5-en