

A Method for Generating Nonverbal Reasoning Items Using n-Layer Modeling

Mark J. Gierl¹(✉), Marita MacMahon Ball², Veronica Vele²,
and Hollis Lai³

¹ Faculty of Education, University of Alberta, Edmonton, AB, Canada
Mark.Gierl@ualberta.ca

² Australian Council for Educational Research, Melbourne, Australia
{Marita.MacMahonBall, Veronica.Vele}@acer.edu.au

³ School of Dentistry, University of Alberta, Edmonton, AB, Canada
Hollis.Lai@ualberta.ca

Abstract. Automatic item generation is the process of using item models to produce assessment tasks using computer technology. An item model is comparable to a template that highlights the variables or elements in the task that must be manipulated to produce new items. When a small number of elements is manipulated in the item model, the generated items look similar to one another and are often referred to as clones. The purpose of our study is to describe a method for generating large numbers of diverse and heterogeneous items using a generalized approach called n-layer item modeling. When a large numbers of elements is manipulated in the n-layer item model, diverse items are generated. We demonstrate the method by generating 1,340 nonverbal reasoning items that would be appropriate for a high-stakes medical admission test.

Keywords: Test development · Automatic item generation · Item writing · Technology and assessment

1 Introduction

Automatic item generation (AIG) [1–4] is a rapidly evolving research area where cognitive theories, computer technologies, and psychometric practices establish a process that can be used to generate test items. AIG can be described as the process of using models to generate items with the aid of computer technology. It requires two general steps. First, content specialists create item models that highlight the elements in the assessment task that can be manipulated. An item model is similar to a template that specifies the variables or elements in the task that must be manipulated to produce new items. Second, the elements in the item model are varied using computer-based algorithms to generate new items. The purpose of this study is to describe and illustrate a method where one item model can be used to generate many test items. The focal content area for item generation in this study is nonverbal reasoning.

2 Item Modeling and the Problem with Cloning

Item modeling provides the foundation for AIG [5, 6]. An item model is comparable to a template, mould, rendering, or prototype that highlights the elements in an assessment task that must be manipulated to produce new items. Elements can be found in the stem, the options, and/or the auxiliary information. The stem is the part of an item model that contains the context, content, and/or the question the examinee is required to answer. The options include the alternative answers with one correct option and one or more incorrect options. For selected-response item models, both stem and options are required. For constructed-response item models, only the stem is created. Auxiliary information includes any additional content, in either the stem or option, required to generate an item. Auxiliary information can be expressed as images, tables, diagrams, sound, or video. The stem and options are further divided into elements. Elements are denoted as strings which are non-numeric content and integers which are numeric content. Often, the starting point is to use an existing test item. Existing items, also called *parent items*, can be found by reviewing previously administered tests, by drawing on existing items from a bank, or by creating the parent item directly. The parent item highlights the structure of the model, thereby providing a point-of-reference for creating alternative items. Then, content specialists identify elements in the parent that can be manipulated to produce new items. They also specify the content (i.e., string and integer values) for these elements.

One drawback of item modeling in the current application of AIG is that relatively few elements can be manipulated because the number of potential elements in any one item model is small. For example, if a parent item contains 16 words in the stem, then the maximum number of elements that can be manipulated is 16, assuming that all words in the stem can be made into elements. One important consequence of manipulating a small number of element is that the generated items may be overtly similar to one another. This type of item modeling can pose a problem in the current application of AIG because many content specialists view this process negatively and often refer to it pejoratively as “cloning”.

Cloning, in a biological sense, refers to any process where a population of identical units is derived from the same ancestral line. Cloning helps characterize item modeling if we consider it to be a process where specific content (e.g., nuclear DNA) in a parent item (e.g., currently or previously existing animal) is manipulated to generate a new item (e.g., new animal). Through this process, instances are created that are identical (or, at least, very similar) to the parent because information is purposefully transferred from the parent to the offspring. Our current approaches to item modeling yield outcomes that are described by content specialists as clones. Clones are perceived by content specialists to be generated items that are overly simplistic and easy to produce. More importantly, clones are believed to be readily recognized by coaching and test preparation companies which limits their usefulness in operational testing programs. Hence, cloned items has limited practical value.

3 n-Layer Item Modeling: A Method to Address the Limitations of Cloning

AIG is the process of using an item model to generate items by manipulating elements in the model. When a small number of elements is manipulated, the generated items look similar to one another and, hence, are referred to as clones. Cloning is synonymous with *1-layer item modeling*. The goal of item generation using the 1-layer model is to produce new test items by manipulating a relatively small number of elements at *one layer* in the model. A generalization of the 1-layer item model is the *n-layer item model* [7]. The goal of automatic item generation using the n-layer model is to produce items by manipulating a relatively large number of elements at *two or more layers* in the model. Much like the 1-layer item model, the starting point for the n-layer model is to use a parent item. But unlike the 1-layer model where the manipulations are constrained to a set of generative operations using a small number of elements at a single level, the n-layer model permits manipulations of a set of generative operations using elements at multiple levels. As a result, the generative capacity of the n-layer model is substantially increased and, in the process, the number of content combinations also increase thereby producing more diversity and heterogeneity among the generated items.

The concept of n-layer item generation is adapted from the literature on syntactic structures of language where researchers have reported that sentences are typically organized in a hierarchical manner [8, 9]. This hierarchical organization, where elements are embedded within one another, can also be used as a guiding principle to generate large numbers of diverse test items. n-layer modeling serves as a flexible method for expressing structures that permit many different but feasible combinations of embedded elements. The n-layer structure can be described as a model with multiple layers of elements, where each element can be varied at different levels to produce different combinations of content and, hence, items.

A comparison of the 1- and n-layer item model is presented in Fig. 1. For this example, the 1-layer model can provide a maximum of four different values for element A. Conversely, the n-layer model can provide up to 64 different values using the same

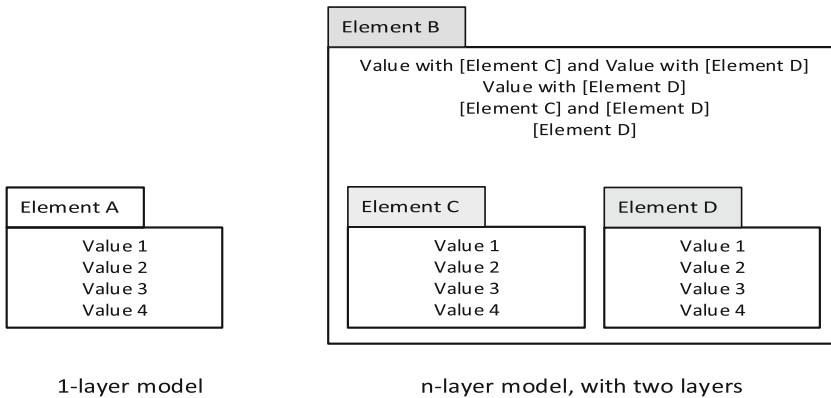


Fig. 1. A comparison of the elements in a 1-layer and n-layer item model.

four values for elements C and D embedded within element B. Because the maximum generative capacity of an item model is the product of the ranges in each element [10], the use of an n-layer item model will always increase the number of items that can be generated relative to the 1-layer structure.

The key advantage of using the n-layer structure is that more elements can be manipulated within the model resulting in generated items that appear to be different from one another. Hence, n-layer item modeling can be used to address the problem of cloning. The disadvantage of using an n-layer structure is that the models are challenging to create given the complexity of combining elements in an embedded fashion. Also, the effect of embedding elements in multiple levels, while useful for generating large numbers of diverse items, may make it challenging to consistently identify the correct solution for every generated item. Hence, constraints are required to ensure that content in the elements and layers are combined in a meaningful way so useful items can be generated. The importance of constraint programming will be illustrated later in our study.

4 Purpose of Study

The purpose of this study is to describe and illustrate a methodology for n-layer item modeling as it applies to generating nonverbal reasoning items. 1-layer item modeling dominates the current application of AIG. The n-layer item model serves as a generalization of the 1-layer approach. n-layer item modeling permits a large number of elements to be manipulated at multiple layers and, as a result, the generated items are more heterogeneous and, therefore, less susceptible to the limitations associated with cloning. We will also demonstrate how this method can be used to generate large numbers of diverse nonverbal reasoning items.

5 Method

The method section is presented in three parts. First, we describe the nonverbal reasoning item type. Second, we present the procedures used to implement the n-layer model. Third, we summarize the item generation process by describing the IGOR software program.

5.1 Nonverbal Reasoning Item Type

To demonstrate the application of the n-layer item modeling method, the nonverbal reasoning item format called “middle of the sequence” was used. A middle of the sequence parent item was selected because it is a format used by the Australian Council for Educational Research on an undergraduate admission test. Scores from the test are used in the selection of students for health science undergraduate programs. Middle of the sequence is one of three item formats used in the nonverbal reasoning section of the test. To solve this item type, examinees are required to reorder five figures to form the simplest and most logical sequence. Then, they select the alternative (A, B, C, D or E) that is in the

middle of the sequence. This task is based on sequences of shapes designed to assess examinees' ability to reason in the abstract and to solve problems in non-verbal contexts.

An example of a middle of the sequence nonverbal reasoning item is shown in Fig. 2. To solve this item, examinees are first required to rotate the subfigure from each corner or vertex of the triangle to the middle position in the base image. Then, examinees are required to identify the most systematic order for the figures so the middle of the sequence can be specified. For our example, this order follows a clockwise rotation beginning in the bottom left corner of the triangle. Therefore, the correct sequence is CADBE and the middle of the sequence is figure D. The correct answer is indicated with an asterisk.

Reorder the five figures to form the simplest and most logical sequence possible. Then, select the alternative that is in the **middle** of the sequence.

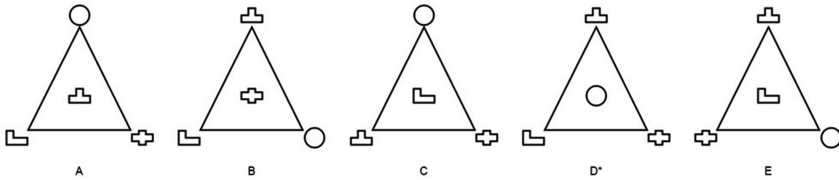


Fig. 2. A “middle of the sequence” nonverbal reasoning item.

5.2 n-Layer Item Modeling Procedure

The n-layer item model was created using the parent item presented in Fig. 2. Six layers were identified and manipulated to generate items. The layers are summarized in Fig. 3. Element 1 is the base image for the nonverbal reasoning item which corresponds to the central figure. Our example contains five base images (i.e., Element 1 = 5 values). Element 2 defines the number of positions for the subfigures located around the base image. Our example has two positions (Element 2 = 2 values). Element 3 specifies the number and shape of each subfigure. Our example has eight subfigures (Element 3 = 8 values). Element 4 specifies the type of rotation permitted by each subfigure around the base image. Our example allows for 12 rotational positions (Element 4 = 12 values). Element 5 highlights the shading pattern for the subfigures. We have nine shading patterns in our example (Element 5 = 9 values). Element 6 is the step logic required to rotate the subfigures from one base figure to the next in the sequence. Our example includes four different step logic sequences (Element 6 = 4 values). Taken together, our 6-layer item model has the element structure of $5 \times 2 \times 8 \times 12 \times 9 \times 4$.

5.3 Item Generation with IGOR

After the model is created, items were generated using IGOR [11]. IGOR, the acronym for **I**tem **G**enerat**OR**, is a software program written in JAVA that produces all possible combinations of elements based on the definitions within the model. To generate items, a model must be expressed in an XML format that IGOR can interpret. Once a model is

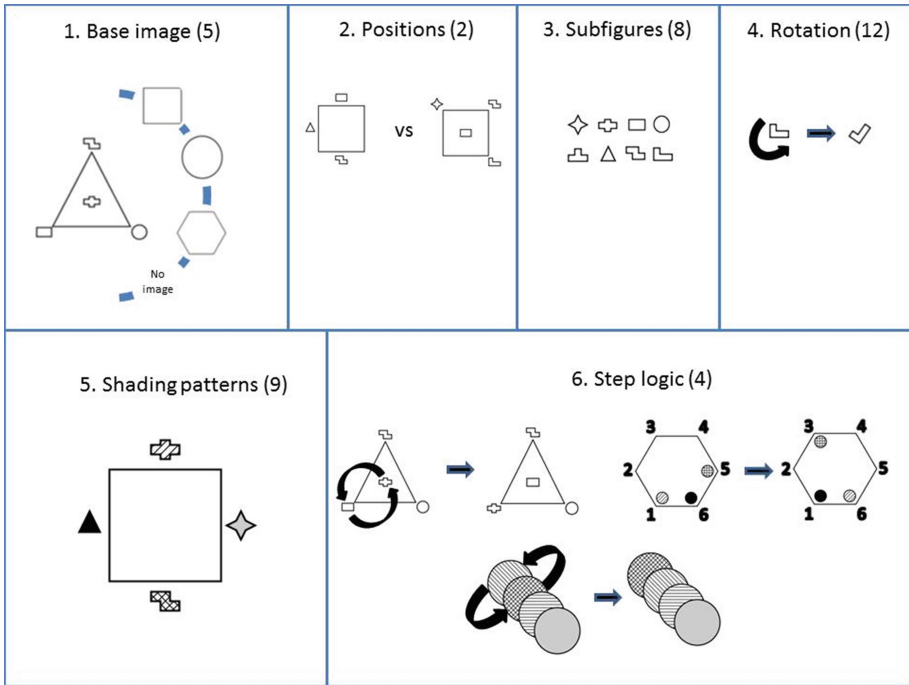


Fig. 3. A 6-layer nonverbal reasoning item model.

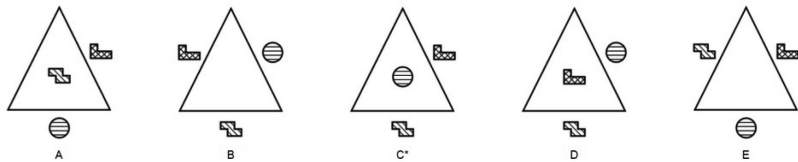
expressed in an XML form, IGOR computes the necessary information and outputs items in either a HTML or a Word format. Iterations are conducted in IGOR to assemble all possible combinations of elements subject to the constraints. Without the use of constraints, all of the elements would be systematically combined to create new items. For example, the 6-layer nonverbal reasoning item model in our example has $5 \times 2 \times 8 \times 12 \times 9 \times 4 = 34,560$ possible combinations. However, some of these items are not useful. Our goal was to generate middle of the sequence items that were comparable to the parent items found on the ACER admission test. As a result, constraints were used to ensure that the element and layer combinations only produced ACER-style nonverbal reasoning items. For instance, when the base image is a circle, the subfigure can only be a star. Or, when the base image is a polygon, the subfigure can only be a star or a circle. Constraints serve as restrictions that must be applied during the assembly task so that meaningful items are generated.

6 Results

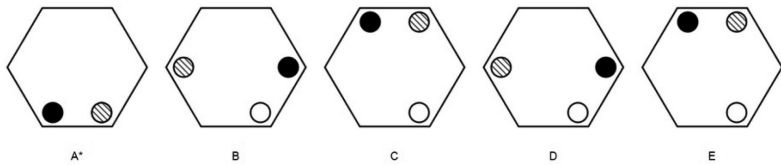
IGOR generated 1,340 items from the 6-layer item model. A sample of five items with different combinations of elements in each of the layers is presented in Fig. 3. To increase generative capacity and to expand item diversity, elements and layers can be added to the existing model or new models can be created with different elements and layers. Hence, n-layer item modeling serves as a generalizable method for creating

large numbers of diverse items and item types by manipulating the elements, layers, and models (Fig. 4).

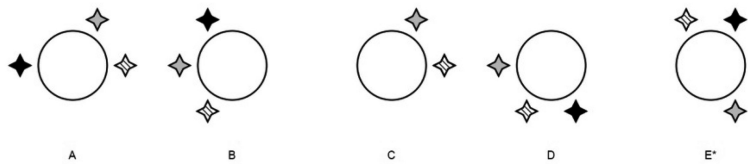
1. Reorder the five figures to form the simplest and most logical sequence possible. Then, select the alternative that is in the **middle** of the sequence.



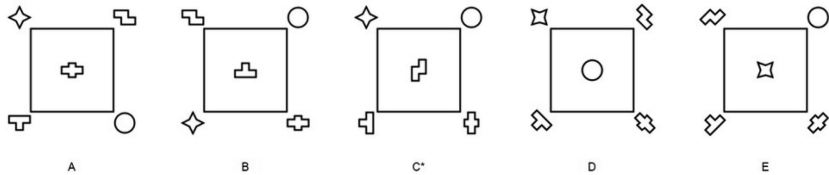
2. Reorder the five figures to form the simplest and most logical sequence possible. Then, select the alternative that is in the **middle** of the sequence.



3. Reorder the five figures to form the simplest and most logical sequence possible. Then, select the alternative that is in the **middle** of the sequence.



4. Reorder the five figures to form the simplest and most logical sequence possible. Then, select the alternative that is in the **middle** of the sequence.



5. Reorder the five figures to form the simplest and most logical sequence possible. Then, select the alternative that is in the **middle** of the sequence.

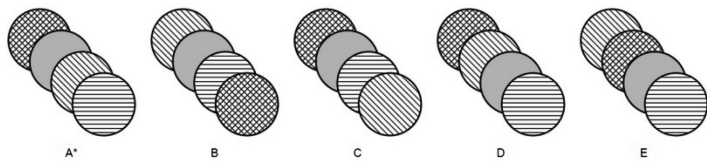


Fig. 4. A sample of five generated items from the 6-layer nonverbal reasoning item model.

7 Conclusions

Testing agencies like the Australian Council for Educational Research require large numbers of high-quality items that are produced in a timely and cost-effective manner. One approach that may help address these challenges is with automatic item generation. AIG is the process of using models to generate items using computer technology. It requires two steps. First, content specialists create item models. Second, the elements in the model are manipulated with computer-based algorithms. With this two-step process, thousands of new items can be created from a single item model, as we demonstrated in the current study. Not surprising, AIG is seen by many administrators in testing agencies as a “dream come true”, given the laborious processes and high costs required for traditional item development. Unfortunately, many content specialists in these same testing agencies are not so enthralled by this dream because they find the quality of the generated items is still lacking.

This study was motivated by our desire to improve the quality of generated items given our discussions with content specialists. Simply put, content specialists dislike cloning because the generated items are too similar to one another for any practical use. We used biological cloning as an analogy for 1-layer item modeling, particularly when the generated items are designed to emulate the statistical properties of the parent. While item cloning has an important role to play in some AIG research [e.g., 12, 13], it is also important to recognize that these types of generated items may have limited value in operational testing programs, according to many content specialists, because they are easily produced, overly simplistic, and readily detectable.

In the current study, we described and illustrated a generalized method called *n*-layer item modeling. The *n*-layer model is a flexible structure for item generation that permits many different but feasible combinations of embedded elements and results in a diverse and heterogeneous pool of generated items. It can be used with any form of template-based item generation. It can be used to generate different item types. And, as was illustrated in our study, it can accommodate a wide range of elements at different layers within the model. We demonstrated the applicability of this method by generating 1,340 middle of the sequence nonverbal reasoning items that could be used by the Australian Council for Educational Research for the Undergraduate Medicine and Health Sciences Admission Test.

7.1 Directions for Future Research

In addition to generating more diverse and heterogeneous items, another application of *n*-layer modeling is generating multilingual test items. Different languages require different words, word orders, and grammatical structures. With a 1-layer model, these variables are not easily or readily manipulated because the generative operations are constrained to a small number elements at a single layer. However, with the use of an *n*-layer model, the generative operations are expanded dramatically to include more elements at multiple layers. Hence, language can serve as a layer that is manipulated during item generation. Therefore, one important direction for future research is to use *n*-layer item modeling to generate tasks in multiple languages by adding language as a

layer in the model. A multilingual n-layer item model would permit testing agencies to generate large numbers of diverse items in multiple languages using a structured item development approach that is efficient and economical.

Acknowledgments. We would like to thank Vasily Tanygin for programming the nonverbal reasoning n-layer item model. We would also like to thank the Australian Council for Educational Research for supporting this research. However, the authors are solely responsible for the methods, procedures, and interpretations expressed in this study and these views do not necessarily reflect those of the Australian Council for Educational Research.

References

1. Drasgow, F., Luecht, R.M., Bennett, R.: Technology and testing. In: Brennan, R.L. (ed.) *Educational measurement*, 4th edn, pp. 471–516. American Council on Education, Washington, DC (2006)
2. Embretson, S.E., Yang, X.: Automatic item generation and cognitive psychology. In: Rao, C.R., Sinharay, S. (eds.) *Handbook of Statistics: Psychometrics*, vol. 26, pp. 747–768. Elsevier, North Holland, UK (2007)
3. Gierl, M.J., Haladyna, T.: *Automatic Item Generation: Theory and Practice*. Routledge, New York (2013)
4. Irvine, S.H., Kyllonen, P.C.: *Item Generation for Test Development*. Erlbaum, Hillsdale, NJ (2002)
5. Bejar, I.I., Lawless, R., Morley, M.E., Wagner, M.E., Bennett, R.E., Revuelta, J.: A feasibility study of on-the-fly item generation in adaptive testing. *J. Technol. Learn. Assess.* **2**(3) (2003). <http://www.jtla.org>
6. LaDuca, A., Staples, W.I., Templeton, B., Holzman, G.B.: Item modeling procedures for constructing content-equivalent multiple-choice questions. *Med. Educ.* **20**, 53–56 (1986)
7. Gierl, M.J., Lai, H.: Using automatic item generation to create items for medical licensure exams. In: Becker, K. (Chair), *Beyond Essay Scoring: Test Development Through Natural Language Processing*. Paper Presented at the Annual Meeting of the National Council on Measurement in Education, Vancouver, BC (2012)
8. Higgins, D., Futagi, Y., Deane, P.: Multilingual generalization of the model creator software for math item generation. *Educational Testing Service Research Report (RR-05-02)*. Educational Testing Service, Princeton, NJ (2005)
9. Reiter, E.: NLG vs. templates. In: *Proceedings of the Fifth European Workshop on Natural Language Generation*, pp. 95–105. Leiden, The Netherlands (1995)
10. Lai, J., Gierl, M.J., Alves, C.: Using item templates and automated item generation principles for assessment engineering. In: Luecht, R.M. (Chair) *Application of Assessment Engineering to Multidimensional Diagnostic Testing in an Educational Setting*. Paper Presented at the Annual Meeting of the National Council on Measurement in Education, Denver, CO. (2010)
11. Geerlings, H., Glas, C.A.W., van der Linden, W.J.: Modeling rule-based item generation. *Psychometrika* **76**, 337–359 (2011)

12. Sinharay, S., Johnson, M.S.: Statistical modeling of automatically generated items. In: Gierl, M.J., Haladyna, T. (eds.) *Automatic Item Generation: Theory and Practice*, pp. 183–195. Routledge, New York (2013)
13. Gierl, M.J., Lai, H., Fung, K., Zheng, B.: Using technology-enhanced processes to generate items in multiple languages. In: Drasgow, F. (ed.) *Technology and Testing: Improving Educational and Psychological Measurement*. Routledge, New York (in press)

Computer Assisted Assessment. Research into
E-Assessment

18th International Conference, CAA 2015, Zeist, The
Netherlands, June 22–23, 2015. Proceedings

Ras, E.; Joosten-ten Brinke, D. (Eds.)

2015, X, 155 p. 29 illus. in color., Softcover

ISBN: 978-3-319-27703-5