Journal of Applied Testing Technology, *Vol 16(1), 1-18, 2015*

A Method for Generating Educational Test Items that are Aligned to the Common Core State Standards

Mark J. Gierl[1*], Hollis Lai[2], James B. Hogan[3] and Donna Matovinovic[4]

[1]Faculty of Education, University of Alberta, 6-110 Education North, Edmonton - T6G 2G5, AB, Canada, mark.gierl@ualberta.ca.
[2]School of Dentistry, Faculty of Medicine and Dentistry, University of Alberta
[3]Assistant Vice President, Assessment Strategy, ACT Inc.
[4]Vice President, Test Development, ACT Inc.

Abstract

The demand for test items far outstrips the current supply. This increased demand can be attributed, in part, to the transition to computerized testing, but, it is also linked to dramatic changes in how 21st century educational assessments are designed and administered. One way to address this growing demand is with automatic item generation. Automatic item generation involves the process of using models to generate items with the aid of computer technology. The purpose of this study is to describe and illustrate a methodology that permits the generation of huge number of diverse and heterogeneous test items that are closely aligned to the Common Core State Standards in Mathematics.

Keywords: Automatic Item Generation, Test Development, Testing and Technology

1. Introduction

The principles and practices that guide the design and development of test items are changing because our assessment practices are changing. Educational visionary Randy Bennett (2001) anticipated that computers and the internet would become two of the most powerful forces of change in educational measurement. Bennett's premonition was accurate. Internet-based computerized testing has dramatically changed educational measurement because test administration procedures combined with the growing popularity of digital media and the explosion in internet use have created the foundation for different types of tests and testing practices. As a result, many educational tests that were once given in a paper format are now administered by computer using the internet. Many common and well-known exams can be cited as examples including the ACT (College Readiness Exam), ACT Aspire, the Graduate Management Admission Test, the Graduate Record Exam, the Test of English as a Foreign Language, the American Institute of Certified Public Accountants Uniform CPA examination, the Medical Council of Canada Qualifying Exam Part I, the National Council Licensure Examination for Registered Nurses, ACT Compass, and the National Council Licensure Examination for Practical Nurses. This rapid transition to computerized testing is also occurring in K-12 education. As early as 2009, Education Week's "Technology Counts" reported that educators in more than half of the US states—where 49 of the 50 states at that time had educational achievement testing—administer some form of computerized testing.The move toward Common Core State Standards will only accelerate this transition given that the two largest consortiums, PARCC and SMARTER Balance, are using technology to develop and deliver computerized tests.

Computerized testing offers many advantages to examinees and examiners compared to more traditional

*Author for correspondence

paper-based tests. For instance, computers support the development of technology-enhanced item types that allows examiners to use more diverse item formats and measure a broader range of knowledge and skills. Computer algorithms can also be developed so these new item types are scored automatically and with limited human intervention thereby eliminating the need for costly and time-consuming marking sessions. Because items are scored immediately, examinees receive instant feedback. Computerized tests also permit continuous and on-demand administration thereby allowing examinees to have more choice about where and when they take their exams.

2. The Need for an Endless Supply of New Test Items

But the advent of computerized testing has also raised new challenges, particularly in the area of item development. Large numbers of items are needed to support the banks necessary for computerized testing when items are continuously administered and, therefore, exposed. As a result, banks must be frequently replenished to minimize item exposure and maintain test security. Breithaupt, Ariel, and Hare (2010) claimed that a high-stakes 40-item computer adaptive test with two administrations per year would require, at minimum, a bank with 2,000 item. The costs associated with creating banks this size are substantial. For instance, Rudner (2010) estimated that the cost of developing one operational item using the traditional approach where content experts use test specifications to individually author each item ranged from $1,500 to $2,500[1]. If we combine the Breithaupt et al. (2010) bank size estimate with Rudner's cost per item estimate, then we can project that it would cost between $3,000,000 to $5,000,000 to develop the item bank for a single computer adaptive test in an assessment program.

3. Automatic Item Generation: One Feasible Solution

One way to address the challenge of creating more items is to hire large numbers of developers who can scale up the traditional, one-item-at-a-time content specialists approach to ensure more items are available. But this option is expensive. An alternative method that may help address the growing need to produce large numbers of new testing tasks is through the use of automatic item generation (AIG). AIG (Gierl & Haladyna, 2013; Embretson & Yang, 2007; Irvine & Kyllonen, 2002) is an evolving research area where cognitive and psychometric theories are used to produce tests that contain items created using computer technology. AIG, an idea described by Bormuth in 1969, is gaining renewed interest because it addresses one of the most pressing and challenging issues facing administrators in assessment programs today—the rapid, efficient, and continuous production of high-quality, content-specific, test items.

4. Benefits of Automatic Item Generation

AIG has at least four important benefits. First, AIG permits test developersto create a single item model that, in turn, yields many test items. An item model is a template which highlights how the features in an assessment task can be manipulated to produce new items. Multiple item models can be developed which will yield hundreds or possibly thousands of new test items. These itemsare then used to populate item banks. A computerized testdraws a sample of the items from the bank to create a new test.

Second, AIG can lead to more cost-effective development because the item model is continually re-used to yield many test items compared with developing each item individually and, often, from scratch. In the process, costly yet common errors in item development (e.g., including or excluding words, phrases, or expressions along with spelling, grammatical, punctuation, capitalization, typeface, and formatting problems) can be avoided because only specific elements in the stem and options are manipulated to produce large numbers of items. In other words, the item model serves as a template for which the test developer manipulates only specific, well-defined, elements. The remaining elements are not altered during development. The view of an item model as a template with both fixed and variable elements contrasts with the more traditional view of a single item where every ele-

[1]We have verified the accuracy of this per item cost estimate with many content specialists, both in the licensure and certification was well as the K-12 achievement testing areas.

ment is unique, both within and across items. Drasgow, Luecht, and Bennett (2006, p. 473) provide this description of the traditional content specialists approach to item development:

The demand for large numbers of items is challenging to satisfy because the traditional approach to test development uses the item as the fundamental unit of currency. That is, each item is individually hand-crafted—written, reviewed, revised, edited, entered into a computer, and calibrated—as if no other like it had ever been created before.

Third, AIG treats the item model as the unit of currency where a single model is used to generate many items compared with a more traditional approach where the item is treated as the unit of analysis, as noted by Drasgow et al. (2006). Hence, AIG is a scalable process because one item model can generate many test items. With a more traditional approach, the test item is the unit of analysis where each item is created individually. Because of this unit of analysis shift, the cost per item should decrease because test developers are producing models that yield multiple items rather than producing single unique items. The item models can also be re-used, particularly when only a small number of the generated items are used on a specific test form.

Fourth, AIG may enhance test security. Security benefits could be realized when large numbers of items are available, simply by decreasing the item exposure rate. In other words, when item volume increases, item exposure decreases, even with continuous testing, because a large bank of operational items is available during test assembly and the use of each individual item is minimized. Security benefits can also be found within the generative logic of item development because the elements in an item model are constantly manipulated and, hence, varied thereby making it challenging for the examinees to memorize the items.

5. Purpose of Study

The purpose of this studyis to describe and illustrate a method for generating test items that are aligned to the Common Core State Standards in Mathematics (CCSSM). In fact, the method we describe can be applied to any standards-based approach where a detailed description of the teaching and learning objectives is available. We used the CCSSM as a point of reference because of its current popularity and importance in the North American testing context. We present the basic logic required for generating items using a template-based method. By template-based AIG, we mean methods that draw on item models to guide the generative process. An item model is comparable to a mould, rendering, or prototype that highlights the features in an assessment task that may be manipulated to produce new items. To ensure our description is both concrete and practical, we illustrate template-based item generation throughout this study using onesample mathematics item (it will be referred to as the "paper mural" item).[2]

The paper is divided into three major sections which correspond to each stage of our study. First, we introduce and demonstrate the logic for generating items using automated processes. This section describes the methods and results used in Stage 1 in our study. Second, we show how each item model stem from Stage 1 can be aligned to different grade and skill categories within the CCSSM to permit the scaling and pre-alignment of the generated items. These activities occurred as part of Stage 2 in our study. Third, we generated the appropriate keys and distractors for the item models from Stage 2. This task was conducted as part of Stage 3 in our study.

6. Stage 1: Item Model Development and Item Generation

Item models provide the foundation for AIG. Item models (Bejar, 1996, 2002; Bejar, Lawless, Morley, Wagner, Bennett, & Revuelta, 2003; LaDuca, Staples, Templeton, & Holzman, 1986) have been described using different terms such as schemas (Singley & Bennett, 2002), blueprints (Embretson, 2002), templates (Mislevy & Riconscente, 2006), forms (Hively, Patterson, & Page, 1968), frames (Minsky, 1974), and shells (Haladyna & Shindoll, 1989). Item models contain the variables in an assessment task that can be manipulated and used for generation. They include the stem, the options, and the auxiliary information. The stem is the part of an item model which contains the context, content, and/or the question the examinee is

[2]The paper mural item is adapted from an operation test item used by ACT Inc. However, the paper mural item is not an operational test item. It is presented in our study for illustrative purposes only.

required to answer. The options include the answers with one correct option and one or more incorrect options or distracters. For multiple-choice item models, both stem and options are required. For constructed-response (also called open ended) 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 can be further divided into elements. Each element contains content that is manipulated to generate new test items. Elements are denoted as strings, which are non-numeric content, and integers, which are numeric content. We will be referring often to the elements of an item model in this study.

Drasgow, Luecht, and Bennett (2006) claimed that items models can be created using either a weak or a strong theory approach. With weak theory, a combination of outcomes from research, theory, and experience provide the guidelines necessary for identifying and manipulating the elements in a model that yield generated items. The weak theory approach is most suitable for broad content domains where few theoretical descriptions exist on the knowledge and skills required to solve test items. With strong theory, a cognitive model provides the principled basis for identifying and manipulating those elements that yield generated items. To date, the use of strong theory AIG has focused on the psychology of specific response processes, such as spatial reasoning (Bejar, 1990) and abstract reasoning (Embretson, 2002), where articulated cognitive models of task performance exist. For most educational achievement tests, few comparable cognitive theories exist to guide item development practices (Leighton & Gierl, 2011) or to account for test performance in broad content areas (Schmeiser & Welch, 2006). Hence, weak theory approaches to item modeling currently prevail.

The goal of automatic generation using an item model cast within a weak theory framework is to produce new assessmenttasks by manipulating a relatively small number of elements in the model. Often, the starting point is to use a parent item whose psychometric characteristics are known. The parent items can be found by reviewing items from previously administered tests, by drawing on an inventory of existing test items, or by creating the parent item directly. The parent item highlights the underlying structure of the model, thereby providing a point-of-reference for creating alternative items. Then by drawing on their experiences, content specialists create the modelby identifying characteristics of the parent that can be manipulated to produce new items. This approach to AIG is called 1-layer item modeling (Gierl & Lai, 2013).

One drawback of using a weak theory 1-layer item modeling is that relatively few elements can be manipulated. The manipulations are limited because the number of potential elements in any one item model is, typically, small. One important consequence of manipulating only a small number of elements is that the generated items may be overtly similar to one another. In our experience, this type of item modeling poses a problem in the application of AIG because most content specialists view this process negatively and often refer to it as "item cloning".

A generalization of the 1-layer item model is the n-layer item model (Gierl & Lai, 2013). The goal of automatic generation using the n-layer model is to generate items by manipulating a relatively large number of elements at two or more layers in a parent item 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 manipulationsare constrained to a linear set of generative operations using a small number of elements at a single level, the n-layer model permits manipulations of a nonlinear set of generative operations using elements at multiple levels. As a result, the generative capacity of the n-layer model is substantially increased.

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 (e.g., Higgins, Futagi, & Deane, 2005). This hierarchical organization, where elements are embedded within one another, can also be used as a guiding principle to generate large numbers of meaningful test items. The use of an n-layer item model is therefore a flexible template for expressing different structures thereby permitting the development of 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 simultaneously at different levels to produce different items. In the computational linguistic literature, our n-layer structure could be characterized as a generalized form of template-based natural language generation, as described by Reiter (1995).

A comparison of the 1- and n-layer item model is presented in Figure 1. For this example, the 1-layer model

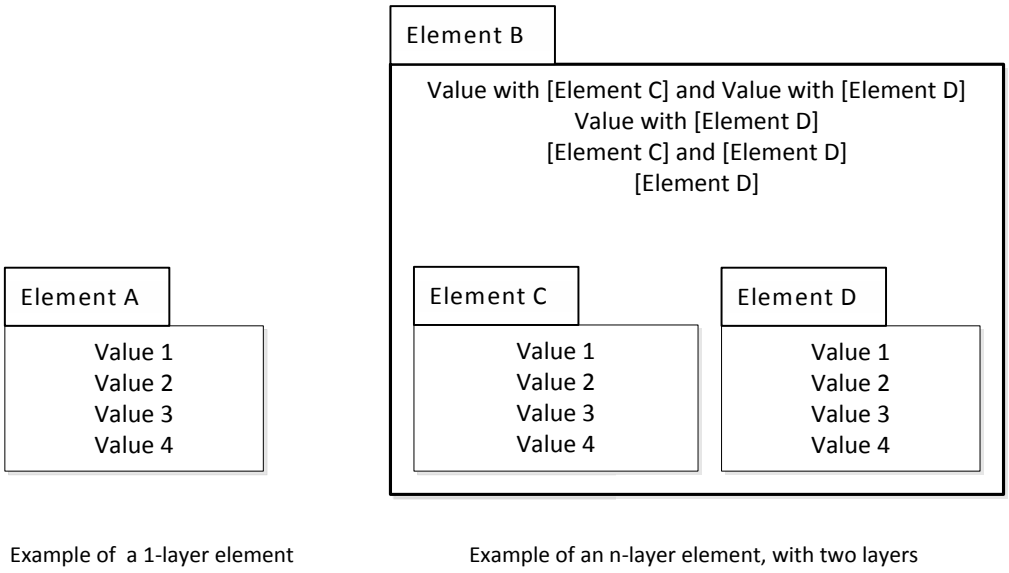

Figure 1. A comparison of the elements in a 1-layer and n-layer item model.

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 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 (Lai, Gierl, & Alves, 2010), the use of an n-layer item model will increase the number of items that can be generated relative to the 1-layer structure.

One important advantage of using a n-layer item model is that more elements can be manipulated simultaneously thereby expanding the generative capacity of the model. Another important advantage is that the generated items will likely appear to be quite different from one another because more content in the model is manipulated. Hence, n-layer item modeling can help address the problem of cloning that concerns some test developers because large numbers of systematic manipulations are occurring in each model thereby promoting heterogeneity in the generated items (this point will be demonstrated quantitatively and qualitatively in Stage 3 of our study). To measure and compare the similarity of items created using different 1- and n-layer models, the cosine similarity index (CSI) can be used. The CSI is a measure of similarity between two vectors that represent co-occurring texts. It is computed using the cosine of the angle between the two vectors in a multidimensional space of unique words or numbers.The CSI will be used to evaluate the similarity of text within the generated items.

Once the n-layer item models are created and the content for these models identified by test development specialists, this information is then assembled to produce new items. This assembly task must be conducted with some type of computer-based assembly system because it is a complex combinatorial problem. Different types of software have been written to generate test items. For instance, Higgins (2007) introduced Item Distiller as a tool that could be used to generate sentence-based test items. Higgins, Futagi, and Deane (2005) described how the software Model Creator can produce math word problems in multiple languages. Singley and Bennett (2002) used the Math Test Creation Assistant to generate items involving linear systems of equations. More recently, Gütl et al. (2011) outlined the use of the Enhanced Automatic Question Creator (EAQC) to extract key concepts from text to generate multiple-choice and constructed-response test items. For this study, we used the IGOR software system described by Gierl et al. (2008) for item generation. IGOR, which stands for Item GeneratOR, is a JAVA-based program designed to assemble the content specified in an item model, subject to elements and constraints articulated in the item model. Iterations are conducted in IGOR to assemble all possible combinations of elements and options, subject to the constraints. Without the use of constraints, all of the variable content would be system-

Table 1. Summary of Generative Outcomes from 18 Parent Items

Item Model	Format/Element Range[1]	Number of Generated Items
Model 1	Five Option/Restricted	> 10,000[2]
Model 2	Five Option/Restricted	660
Model 3	Five Option/Restricted	> 10,000
Model 4	Five Option/Restricted	3,198
Model 5	Open Ended/Restricted	>10,000
Model 6	Open Ended/Restricted	3,468
Model 7	Five Option/Restricted	852
Model 8	Five Option/Restricted	3,060
Model 9	Five Option/Restricted	>10,000
Model 10	Open Ended/Restricted	>10,000
Model 11	Five Option/Restricted	>10,000
Model 12	Five Option/Restricted	1,788
Model 13	Five Option/Restricted	1,026
Model 14	Five Option/Restricted	4,168
Model 15	Five Option/Restricted	>10,000
Model 16	Five Option/Restricted	>10,000
Model 17	Five Option/Restricted	1,080
Model 18	Five Option/Restricted	>10,000
Total		>109,300

[1]With a restricted element range, only a sample of items are generated. By increasing the range, the number of generated items would increase.

[2]Item generation was truncated at 10,000 as an upper bound. Hence, 10,000 is considered a large number of items in our study.

atically combined to create new items. However, some of these items would not be sensible or useful. Constraints therefore serve as restrictions that must be applied during the assembly task so that meaningful items are generated.

To illustrate the logic and the application of the AIG method implemented in Stage 1, we will use the paper mural item example. The paper mural item is based on one of the 18 parent items selected and developed by the mathematics content specialists for this study (the 18 parent items are referred to as models 1 to 18, respectively). The parent paper mural item is shown in Figure 2. The item model is shown in Figure 3. For this item model, string elements included four names, two genders, three product names, and three product materials.The string elements help illustrate the n-layering approach. The stem is presented as:

Alex is coloring a paper mural using 80 crayons shared with 9 of his friends. Each of his friends has the same number of crayons. There were 8 crayons left over after Alex handed them out to his friends. Which of the following equations represents this situation?

This stem can be transformed into the following item model:

<Name> is coloring a <Product.Name> using <Product.Material> shared with <Gender> friends. Each of <Gender> friends has the same number of <Product.Material>, <Product.number>. There were <Product.Material> left over after <Name> handed them out to <Gender> friends. Which of the following equations represents this situation?

The process of n-layer then involves the substitution of different values for the name, gender, produce names, and produce materials across all combination of values to produce different sets of items. The key and the distractors for the multiple-choice answer options used rangesofinteger elements.In total, the paper mural item model generated 660 items for one CCSSM (4.OA.A.3) at one grade (4).

Our n-layer item modeling AIG method was also applied to 17 other parent items that measured important CCSSM. These parent items were selected by the Mathematics content specialists. A summary of the generated outcomes across the 18 parent items is presented in Table 1.

7. Stage 2: Align Item Models to CCSSM

The CCSSM define what students should understand and be able to do in their study of mathematics. The Standards are intended to describe these outcomes, with clarity and specificity, from Kindergarten to high school. The paper mural item assesses understandings and skills associated with 4.OA.A.3 in CCSSM. This set of understandings and skills is a part of the fourth grade expectations (4), the Operations and Algebraic Thinking domain (4.OA), the first cluster (4.OA.A)—use the four operations with whole numbers to solve problems. The cluster level is the unit of coherence for CCSSM. The content included in the paper mural item is strongly connected to the third clarifying part of the cluster description (4.OA.A.3), "Solve multistep word problems posed with whole numbers and havingwhole-number answers using the four operations, including problemsin which remainders must be interpreted. Represent these problemsusing equations with a letter standing for the unknown quantity.Assess the reasonableness of answers using mental computation

Alex is coloring a paper mural using 80 crayons shared with 9 of his friends. Each of his friends has the same number of crayons, x . There were 8 crayons left over after Alex handed them out to his friends.
Which of the following equations represents this situation?

$80 = 9x + 8$ *
$80 = 8x + 9$
$80 = 9(8) + x$
$8 = 9x - 80$
$80 = 8x - 9$

Common Core Standard: 4.OA.A.3

Figure 2. Paper mural parent item, with CCSSM identified.

Table 2. Common Core Standard 4.OA.A.3 for the Paper Mural Parent Item

4.OA Operations and Algebraic Thinking
A. Use the four operations with whole numbers to solve problems.
3. Solve multistep word problems posed with whole numbers and having whole-number answers using the four operations, including problems in which remainders must be interpreted. Represent these problems using equations with a letter standing for the unknown quantity. Assess the reasonableness of answers using mental computation an destimation strategies including rounding.

andestimation strategies including rounding". A summary of 4. OA.A.3 is presented in Table 2. Identifying the appropriate domain and standard and then applying this classification to items and item models was the focal task in Stage 2.

Stage 2 required two related steps. In the first step, the CCSSM measured by each item model developed in Stage 1 were identified. The most closely related CCSSM at other grade levels were also identified. In the second step, the stem for each item model from Stage 1 was modified so that it could be used to create new item models that measured CCSSM for the related content areas and skill levelsacross grades, as identified in step 1 of Stage 2. At this point, we must introduce some new concepts that were developed for this study.

This study introduces the reader to new methods for scaling item development using AIG. Hence, the concept of "scalability" is used as a guiding principle within our n-layering AIG methodology. Scaling in AIG occurs when items are used to create item models. The item models, in turn, are used to generate items. The generated items can then be used to create new item models. This process of systematically embedding items and item models within one another can occur continuously—this continuous process is what we mean by scaling n-layer item modeling in AIG. The starting point for developing an item model is a parent item. Figure 2 contains the parent item for the paper mural example. Earlier in this study, we described a parent item as an existing item that highlights the underlying structure of the item model and provides a point-of-reference for generating alternative items. All parent items in this study were aligned to CCSSM by the Mathematics content specialists. Then, the content specialist create an item model by identifying characteristics in the parent item that can be manipulated to produce new items. When this parent item is the first item to initiate the AIG process, it can be referred to as a generationzero or G0 parent item. That is, a G0 item starts n-layer AIG. An item model, by way of contrast, specifies thevariables within an item that can be manipulated and used to produce more items. When the item model is the first model in the AIG process, it can be referred to as the generation zero or G0 item model.In other words, a G0 item model is the first item model in the AIG process. Finally, parent item models can be used to create sibling items and sibling item models. We will introduce a method in this study for using parent item models to produce sibling item models. A parent item model is the original model that, in turn, is modified to create a new sibling item model. Because scaling involves layering items and models, the concept of a parent and sibling can become confusing because siblings can also become parents that, in turn, produce siblings. Therefore, we will maintain the term "generation" in the lineage sense of the wordand describe the first sibling item model as generation one or the G1 item model. Step 2 in Stage 2 required the creation of G1 item models using the G0 item models. The G0 item models were created from G0 items in Stage 1.

<**Name**>is coloring a <**Product.Name**>using <**Product.Material**>shared with <**Gender**> friends. Each of <**Gender**>friendshasthe same number of <**Product.Material**>, <**Product.number**>.There were <**Product.Material**>left over after <**Name**>handed them out to <**Gender**> friends.Which of the following equations represents this situation?

Figure 3. The item model for the paper mural parent item.

Table 3. Summary of Common Core Standards in Mathematics Related to 4.OA.A.3

3.OA Operations and Algebraic Thinking
A. Represent and solve problems involving multiplication and division.
1. Interpret products of whole numbers, e.g., interpret 5 × 7 as the total number of objects in 5 groups of 7 objects each. For example, describe a context in which a total number of objects can be expressed as 5 × 7.
2. Interpret whole-number quotients of whole numbers, e.g., interpret 56 ÷ 8 as the number of objects in each share when 56 objects are partitioned equally into 8 shares, or as a number of shares when 56 objects are partitioned into equal shares of 8 objects each. For example, describe a context in which a number of shares or a number of groups can be expressed as
56 ÷ 8.
3. Use multiplication and division within 100 to solve word problems in situations involving equal groups, arrays, and measurement quantities, e.g., by using drawings and equations with a symbol for the unknown number to represent the problem.
4. Determine the unknown whole number in a multiplication or division equation relating three whole numbers. For example, determine the unknown number that makes the equation true in each of the equations 8 × ? = 48, 5 = ?÷ 3, 6 × 6 = ?.
5.OA Operations and Algebraic Thinking
Write and interpret numerical expressions.
1. Use parentheses, brackets, or braces in numerical expressions, andevaluate expressions with these symbols.
2. Write simple expressions that record calculations with numbers, and interpret numerical expressions without evaluating them. For example, express the calculation "add 8 and 7, then multiply by 2" as 2 × (8 + 7). Recognize that 3 × (18932 + 921) is three times as large as 18932 + 921, without having to calculate the indicated sum or product.
6. EE Expressions and Equations
A. Apply and extend previous understandings of arithmetic to algebraic expressions.
2. Write, read, and evaluate expressions in which letters stand for numbers.
a. Write expressions that record operations with numbers and with letters standing for numbers. For example, express the calculation "Subtract y from 5" as 5 – y.

8. Step 1: Identify CCSSM Related to G0 Item Models

The paper mural item measures CCSSM 4.OA.A.3.The paper mural item model was created so that it also measured 4.OA.A.3 (Table 3). Based on this standard, the Mathematics content specialists linked the CCSSM outcomes for the paper mural item model to CCSSM at different grade levels—3.OA.A, 5.OA.A, and 6.EE.A.2. In other words, the paper mural item model could be explicitly linked to standards at three other grade levels (a summary of the linked standards is provided in Table 3).

In total, the CCSSM for 10 of the 18 item models from Stage 1 were readily linked to CCSSM at different grade levels by the Mathematics content specialists. A summary of the standard specified for each G0 item model along with a list of the related standards for each model is presented in Table 4. The number of grade levels measured by each model is also presented. For the 10 G0 item models developed in Stage 2 of our study, the number of

grade levels ranged from 3 to 5. The average number of linked grade levels per G0 item model was 3.8. That is, the content and skills specified for a standard from a single G0 item model created in Stage 1 could be directly linked to standards at approximately four different grade levels in the CCSSM. This result implies thata single G0 item model could be adapted to measure outcomes at four different grade levels in CCSSM.

9. Step 2: Create G1 Item Model Stems To Measure Different CCSSM

With each G0 item model positioned within the CCSSM framework and with the related CCSSM for each G0 item model clearly identified by the Mathematics content specialists, the next task is to modify the stem in the G0 item model to produce a G1 item model that could be used to measure the content and skills specified in the related CCSSM but at different grade levels. This item model adaptation task served as step 2 in Stage 2 of our study. The Mathematics content specialists modified each G0 item model using the requirements of the related CCSSM. The adaptation was conducted by placing the G0 item model beside its standard, and then listing the grade-related standards below and above the grade-specific standard for the G0 model. The content in the stem for each G0 item model was then adjusted so that it met the requirements for the related standard. The adjustment process was guided by the judgements, experiences, and discussions among the Mathematics content specialists. A sample of the coding sheet used for the paper mural G0 item model is presented in Figure 4. The Mathematics content specialists identified three item model "versions" for 3.OA.A, six "versions" for 5.OA.A, and three "versions" for 6.EE.A.2 (Figure 4 contains a sample of two version per Standard). This outcome can be described, using Grade 3 as an example, as G1 3.OA.A V1 (i.e., Generation1 item model, 3.OA Domain, A Standard, Version1), G1 3.OA.A V2 item model, and G1 3.OA.A V3 item model. Hence, for this example, the paper mural G0 item model was directly related to standards at three different grade levels, with multiple item model versions per grade. Or, said differently, a single G0 item model for paper mural was scaled to 12 G1 items models, all of which were aligned to the CCSSM. In short, when the n-layer AIG method is combined with the CCSSM to guide item production, a dramatic increase occurs in our ability to scale test development.

In total, 32 G1 item models with 81 different versions were created by the content specialists using the 10 G0 item models. The number of versions for the G1 item models ranged from 4 to 13. The average number of G1 items models created for each G0 item model was 8.1. This outcome reveals that approximately eight G1 item models can be created for each G0 item model with the added benefit that the G1 models are aligned to the CCSSM. The benefit of this alignment is fully realized when we then place these 81 G1 item models back into the Stage 1 n-layer AIG methodology to generate new test items which measure specific CCSSM at multiple grade levels. In other words, the results from Stage 2 can easily be scaled because the Stage 2 outcomes are layered or embedded within the Stage 1 AIG method.

10. Stage 3: Generating key and Distractors for Adapted Item Models

With a large number of G1 item model stemsfrom Stage 2 now available, the last task is to specify the formula required to generate the keys and distractors for each item model. Recall, the task for step 2 in Stage 2 was to design the stem for each G1 item model. The G1 item models, while closely related to the parent (i.e., G0 item model), will still contain new elements. As a result, the keys and distractors must be created so that they conform to the element changes thereby producing the correct answer as well as plausible but incorrect distractors for each generated item. With the paper mural G0 item model, changes to the string elements (e.g., <Name>,<Product.Material>, <Product.Name>) must be carefully coordinated with changes to the integer values for each G1 item model to produce generated items with appropriate keys and distractors. The starting point for creating the keys and distractors comes from the formula used with the G0 item models (Figure 3). But the keys and distractors must also be coordinated with changes made to the string and integer elements in each G1 item model so that the generated options are plausible. Moreover, the range for the integer elements must be updated to ensure the numeric values are appropriate for students across different grade levels.

Standard	Item Model Adaptation
3.OA.A	Alex has a total of 69 crayons to make paper murals. Each paper mural will require 23 different crayon colors. What is the maximum number of paper murals Alex can make?
	Alex will make paper murals. Each paper mural will have 22 crayon colors. What is the total number of crayon colors 11 paper murals will have?
4.OA.A.3	PARENT ITEM
5.OA.A	Alex has 4 crayons. Leena has 13 times as many crayons as Alex. Then, Leena got 27 extra colors from a friend. Which of the following expressions represents the total number of crayons Alex and Leena have?
	Alex has 23 crayons. Leena has 13 times as many crayons as Alex. Then, Leena got 41 extra crayons from a friend. What is the total number of crayons Alex and Leena have?
6.EE.A.2	Alex has 25 crayons. Leena has 13 times as many crayons as Alex. Then, Leena got b extra crayons from Jeff. Which of the following expressions represents the total number of crayons Alex and Leena have?
	Alex has 14 crayons, and Leena has b crayons. For a class project, Alex and Leena need a total number of crayons equal to 7 times the crayons they currently have altogether. Which of the following expressions represents the total number of crayons Alex and Leena need for the school project?

Figure 4. The specific and the adapted parent item model stem by CCSSM.

The programmed solutions we present in this study are based on the recommendations and suggestions provided to us by the Mathematics content specialists. Using the paper mural example, a sample of the key and distractor formulas for one version at each grade level for a G1 item model is presented in Figure 5.

The key and distractors were created for five of the 10 Stage 2 G1 item models. We found that a large number of specific outcomes must be produced and then reviewed for each G1 item model to ensure that the key generates the correct response and that the distractors yield plausible but incorrect responses for each item model. Therefore, in Stage 3, we sampled half of the G1 item models from Stage 2 to demonstrate the process required to create the keys and distractors. But there is a clear benefit of this addition work—the increased generation capacity for each of the five G1 item models relative to their initial G0 specifications. At Stage 1, the generative capacity ranged from 660 to > 10,000 for the five item models (Table 1). At Stage 3, the generative capacity was > 10,000 for all five item models (Table 5).

Also, to demonstrate that the n-layer modeling produces more diversity among the generated items, word similarity was compared for each of the five G1 item models relative to their previous G0 incarnation. To measure and compare the similarity of items created using different n-layer model, the intra-model differences, meaning items generated within the same model, must be evaluated. Similarity can be quantified usingthe cosine similarity index (CSI). The CSI is a measure of similarity between two vectors that represent co-occurring texts. It is computed using the cosine of the angle between the two vectors in a multidimensional space of unique words or numbers.The CSI can be expressed as

$$\cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|},$$

where A and B are two items expressed in a binary vector of word or number occurrences. For example, if A is a list of three words (e.g., dog, walk, talk) and B is a list of three words (e.g., cat, walk, mock), then the length of both binary vectors is the number of unique words used across both lists (i.e., dog, walk, talk, cat, mock). To vectorize A and B so the words and numbers can be compared, the occurrence of each word or number in the corresponding list is quantified with a value of 1. The resulting vectors for

Standard	Stem	Key	Distractor 1	Distractor 2	Distractor 3	Distractor 4
3.OA.A	Alex has a total of 69 crayons to make paper murals. Each paper mural will have 23 crayon colors. What is the maximum number of paper murals Alex can make?	[[Total. No. of crayons]] / [[Crayon colors per paper mural]]	[[Total. No. of crayons]] + [[Crayon colors per paper mural]]	[[Total. No. of crayons]] - [[Crayon colors per paper mural]]	[[Total. No. of crayons]]	[[Crayon colors per paper mural]]
4.OA.A.3	PARENT ITEM					
5.OA.A	Alex has 4 crayons. Leena has 13 times as many crayons as Alex. Then, Leena got 27 extra crayons from a friend. Which of the following expressions represents the total number of crayons Alex and Leena have?	Expression only : [[Alex's crayons]] + (x times) ([[Alex's crayons]]) + [[extra number]]	[[Alex's crayons]] + (x times)([[Alex's crayons]]) - [[extra number]]	(x times)([[Alex's crayons]]) +[[extra number]]	(x times) [[Alex's crayons]] +[[extra number]])	[[Alex's crayons]] + (x times) [[Alex's crayons]] + [[extra number]])
6.E.E.A.2	Alex has 25 crayons. Leena has 13 times as many crayons as Alex. Then, Leena got b extra crayons from Jeff. Which of the following expressions represents the total number of crayons Alex and Leena have?	Expression only : [[Alex's crayons]] + (x times) ([[Alex's crayons]]) + [[extra number]]	[[Alex's crayons]] + [[x times]]*[[Leena crayons]] x b	[[x times]] ([[Alex's crayons]] + [[Leena crayons]]) + b	[[x times]] ([[Alex's crayons]] + [[Leena crayons]]) - b	[[Alex's crayons]] + [[x times]] ([[Leena crayons]] + b)

Figure 5. Sample of the key and distractor formulas for the adapted paper mural item models using one grade-specific CCSSM.

A and B in our example are [1,1,1,0,0] and [0,1,0,1,1]. The CSI has a minimum value of 0, meaning that no word or number overlapped between the two vectors, and a maximum of 1, meaning that the text represented by the two vectors are identical.

The CSI results are presented in Table 6. For the items generated using the G0 models, the CSI ranged from 0.36 to 0.66. The average CSI across the five G0 models was 0.53. For the items generated using the G1 models, the CSI ranged from 0.20 to 0.55. The average CSI across

Table 4. Summary of Parent Item Models, Related CCSSM, and Number of Grade Levels

Item Model	Specific Standard	Related Standards	Number of Grade Levels
Model 1	4.MD.A	3.MD.C.7.A; 3.MD.C.7.B; 3.MD.D.8; 5.MD.A	3
Model 2	4.OA.A.3	3.OA.A; 5.OA.A; 6.EE.A.2	4
Model 3	5.NBT.A.3.B	4.NBT.A.2; 4.NBT.B.5; 5NBT.B.6; 6.NS.B.2	3
Model 4	3.OA.A	1.OA.A; 2.OA.A; 4.OA.A	4
Model 5	3NBT.A.1.B	4NBT.A.3; 5NBT.A.4	3
Model 6	3.OA.D.9	4.OA.C.5; 5.OA.B.3	3
Model 7	5.NF.A.2	4.NF.B.3D; 6.NS.A.1; 7.RP.A.3.	4
Model 8	6.RP.A.2	4.NF.C; 5.NF.B; 7.RP.A	4
Model 12	6G.A.2	3MD.D.8; 4MD.A.3; 5MD.C.5.B; 7G.B.6	5
Model 14	6RP.A.3.C	4.NF.B.4; 5.NF.B.7; 7RP.A.3; 8.F.B.4	5

the five G1 models was 0.33. Recall that the CSI ranges from 0 (no word or number overlap) to 1 (complete word and number overlap). These results reveal that G1 item modeling did, in fact, add substantial variability to the generation process. For every model in Stage 3, the CSI mean decreased—meaning that the generated items were more diverse—as we moved from G0 to G1 item models. In cases where the stem and options contain a lot of written content, the CSI change was dramatic (e.g., Model 1 G0 CSI=0.66; G1 CSI=0.32) and in other cases when the stem and options were largely numeric, the change was moderate (e.g., Model 7 G0 CSI=0.64; G1 CSI=0.55). But, regardless of the content of the item model, n-layering clearly adds diversity to the generation process thereby producing more heterogeneous itemsas the number of layers increases in the modeling process.

11. Summary and Implications for Test Development

The main objective of this study was to create a methodology to scale the generation process while, at the same time, to ensure that the generated items were aligned to specific CCSSM. In this study we describe the logic required for generating items using a template-based method. By template-based AIG, we mean methods that draw on item models to guide the generative process. To ensure our description is both concrete and practical, we

Table 5. Summary of Generative Outcomes from 5 Stage 3 G1 Item Models

Item Model	Format/Element Range	Number of Generated Items
Model 1	Five Option/Restricted	> 10,000
Model 2	Five Option/Restricted	> 10,000
Model 3	Five Option/Restricted	> 10,000
Model 7	Five Option/Restricted	> 10,000
Model 12	Five Option/Restricted	> 10,000
Total		**>50,000**

Table 6. CSI Measures for Items Generated Using the G0 and G1 Models

Item Model	G0 CSI Mean (SD)	G1 CSI Mean (SD)
Model 1	0.66 (0.09)	0.32 (0.18)
Model 2	0.38 (0.27)	0.23 (0.20)
Model 3	0.60 (0.01)	0.36 (0.29)
Model 7	0.64 (0.09)	0.55 (0.14)
Model 12	0.36 (0.25)	0.20 (0.21)
Overall	**0.53 (0.14)**	**0.33 (0.20)**

illustrate template-based item generation using the paper mural item (Figure 2). Our study was divided into three major sections. The first section, called Stage 1, described the AIG methods and results used in our study. The second section, called Stage 2, demonstrated how each item model stem from Stage 1 could be aligned to different grade and skill categories within the CCSSM to permit scaling of the generated items. The third section, called Stage 3, outlines a method for generating the appropriate keys and distractors for the item models from Stage 2. Next, we describe some of the possible implication of item generation for test development. We begin with a discussion of how AIG can be scaled using the methods and results developed for this study. Then, we describe the importance and the benefits of aligning content outcomes to test specifications before generating items.

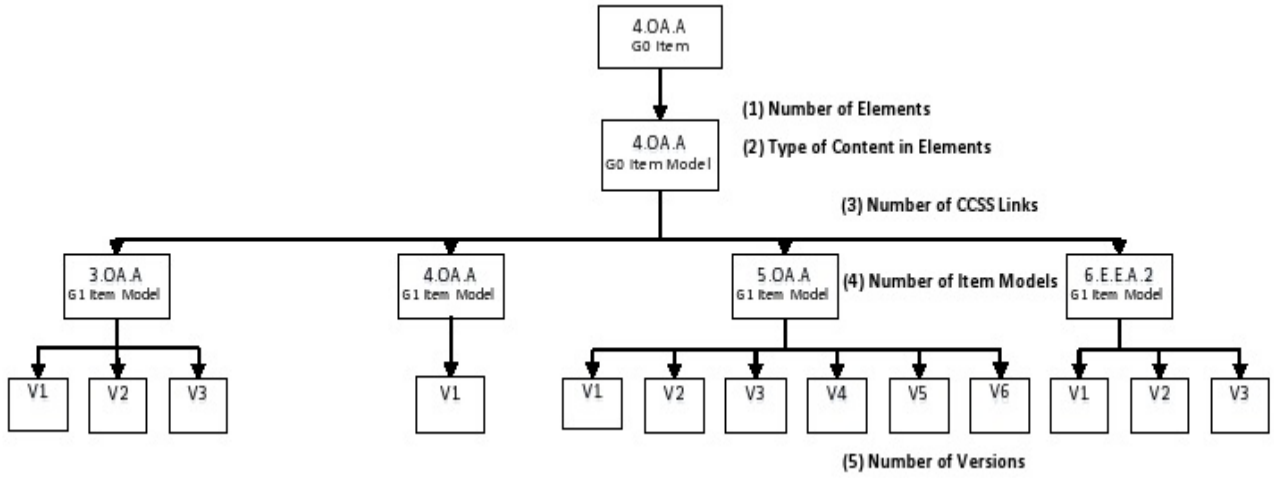

Figure 6. The content specialist's five decision-making points that affect the number of items that will be generated.

12. Scaling AIG to Promote Item Generation

We implemented a three-stage process in the current study. In Stage 1, wedescribed and demonstrated the logic for generating items. In Stage 2, we showed how each item model stem from Stage 1 can be aligned to different grade and skill categories within the CCSSM. In Stage 3, we generated the appropriate keys and distractors for the pre-aligned item models from Stage 2. We also noted that Stage 2 required two related steps. In the first step, the CCSSM measured by each item model developed in Stage 1 were identified. The most closely related CCSSM at other grade levels were also identified. In the second step, the stem for each item model from Stage 1 was modified so that it could be used to create new item models that measured CCSSM for the related content areas and skill levels across grades, as identified in step 1 of Stage 2.

Stage 2 introduced a new method for scaling AIG. Scaling occurs when items and item models are embedded within one another using our n-layering approach. We also demonstrated how this process of systematically embedding items and item models within one another can produce an increasingly large number of diverse and heterogeneous test items. The starting point is to identify a Generation 0 parent item (G0 item). This item becomes the basis for creating a Generation 0 item model (G0 item model). G0 item models are then modified—in our study the modification was based on content specifications in the CCSSM—to produce siblings, which we called Generation 1 item models (G1 item models). As we move from G0 items to G0 item models and then from G0 item models to G1 item models, generative capacity and item diversity increases (Table 1, 5, and 6).

It is also important to note, however, that item generation capacity is guided and controlled, in large part, by the substantive decisions made by content specialists. Content specialists are responsible for specifying the number of elements in the item model, for producing the content in each element, for linking the G0 item models to G1 item models, and for creating either a single version or multiple versions of each item model. These five items are shown in Figure 6 (right side) using the paper mural example. Item generation capacity is governed by (1) the number of elements in the G0 item model (the paper mural item model stem contained four string elements, <names>, <gender>, <product name>, <product material>), (2) the amount and type of content for each element (e.g., string element <product names> in the paper mural item model contained three variations), (3) the number of CCSSM links for each item model (the CCSSM of 4.OA.A.3 for paper mural was linked to 3.OA.A; 5.OA.A; 6.EE.A.2),

(4) the number of item models (paper mural was used to create four item models), and (5) the number of versions for each item model (12 different versions of the four paper mural item models were created). An increase or decrease in the outcome for one or more of these five decision-making points will affect generation capacity thereby demonstrating how n-layer modeling can serve as a powerful catalyst for item development, particularly when the goal is to produce large numbers of diverse and heterogeneous test items.

13. Guiding the Production of Item Content using AIG

While it is beneficial to use our n-layering AIG approach for high-output item production, it is also important to have some control over the outcomes of the generation process. The generated items must be properly coded and banked if they are to be useful in the test development process. That is, the generated items must be closely associated with the content codes in the developer's test specification or test blueprint. These content codes are needed to provide the developer with the required information necessary to efficiently access the generated items from the bank during test assembly or test administration. Our research on AIG, to-date, has relied on an "exploratory" approach. By exploratory, we mean that parent items were first identified, then item models were created and generation conducted and, finally, the generated items were coded. This approach to item generation is inefficient when large numbers of items are created because content classification is conducted as the last step after a large number of items are produced. Exploratory AIG in analogous to exploratory factor analysis where, in factor analysis, the technical analysis is conducted first (i.e., the items are statistically associated with factors) and the content analysis is conducted second (i.e., the substantive meaning of the factors is determined by the content specialists).

In our study, we introduced a new approach to item generation that can be described as "confirmatory" AIG. By confirmatory, we mean that content or, in the current study, the mathematics standards, for the parent items are specified first, item models that measure these standards are created second and, finally, items that measure these standards are generated at the end. The outcome of this process is that the generated items are "pre-aligned" to the CCSSM because of the careful attention devoted to content alignment during the creation of the item models. The benefit of this confirmatory approach is that the generated items contain CCSSM codes which are based on the Mathematics content specialists initial CCSSM classification of the elements and the content in the item models. Hence, the benefits of scaling AIG (i.e., using n-layer modeling to produce large numbers of diverse and heterogeneous test items) can be combined with the benefits of confirmatory AIG (i.e., generating items pre-aligned to specific content and skill specifications like CCSSM) to implement a new methodology for developing test items that helps addressone of the most pressing and challenging issues facing testing companies—the rapid, efficient, and continuous production of high-quality, content-specific, test items. Large numbers of diverse mathematics items that are aligned to the CCSSM can now be generated using the methods described and illustrated in this study.

14. Directions for Future Research

In the current study, our applications were limited to the multiple-choice item format. Multiple-choice items are used extensively in large-scale testing programs. Hence, it is desirable to generate this item format. With the implementation of computerized testing, alternative item formats will become popular. Hence, one direction for future research is to apply the three-step AIG method presented in this study to other item formats. Also, studies designed to validate the meaning of the generated items and to use the items to make inferences about students' knowledge and problem-solving skills have not been conducted. These activities can include substantive and statistical studies designed to evaluate item quality. To-date, the quality of the generated items have received limited empirical evaluation. Hence, future studies are required to assess the quality of the generated items.

15. Acknowledgements

We would like to thank the members of the ACT Mathematics Test Development team for their help during this study: Ken Mullen, Kirsty Gardner, Dennis Kwaka,

Dalia Allencherry, and Ian MacMillan. The authors would like to thank ACT Inc. for their support with this research. However, the authors are solely responsible for the methods, procedures, and interpretations expressed in this study. Our views do not necessarily reflect those of the ACT Inc.

16. References

Bejar, I. I. (1990). A generative analysis of a three-dimensional spatial task. Applied Psychological Measurement, 14, 237–245.

Bejar, I. I. (1996). Generative response modeling: Leveraging the computer as a test delivery medium (ETS Research Report 96-13). Princeton, NJ: Educational Testing Service.

Bejar, I. I. (2002). Generative testing: From conception to implementation. In S. H. Irvine & P. C. Kyllonen (Eds.), Item generation for test development (pp.199–217). Hillsdale, NJ: Erlbaum.

Bejar, I. I., Lawless, R., Morley, M. E., Wagner, M. E., Bennett, R. E., & Revuelta, J. (2003). A feasibility study of on-the-fly item generation in adaptive testing. Journal of Technology, Learning, and Assessment, 2(3). Available from http://www.jtla.org.

Bennett, R. (2001). How the internet will help large-scale assessment reinvent itself. Educational Policy Analysis Archives, 9, 1–23.

Breithaupt, K., Ariel, A., & Hare, D. (2010). Assembling an inventory of multistage adaptive testing systems. In W. van der Linden & C. Glas (Eds.), Elements of adaptive testing (p. 247–266), New York, NY: Springer.

Drasgow, F., Luecht, R. M., & Bennett, R. (2006). Technology and testing. In R. L. Brennan (Ed.), Educational measurement (4th ed., pp. 471–516). Washington, DC: American Council on Education.

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

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). North Holland, UK: Elsevier.

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

Gierl, M. J., & Lai, H. (2013). Using automated processes to generate test items. Educational Measurement: Issues and Practice, 32, 36–50.

Gierl, M. J., Zhou, J., & Alves, C. (2008). Developing a taxonomy of item model types to promote assessment engineering. Journal of Technology, Learning, and Assessment, 7(2). Retrieved from http://www.jtla.org.

Gütl, C., Lankmayr, K., Weinhofer, J., & Höfler, M. (2011). Enhanced Automatic Question Creator – EAQC: Concept, development and evaluation of an automatic test item creation tool to foster modern e-education.Electronic Journal of e-Learning, 9, 23–38.

Haladyna, T., &Shindoll, R. (1989). Items shells: A method for writing effective multiple-choice test items. Evaluation and the Health Professions, 12, 97–106.

Higgins, D. (2007). Item Distiller: Text retrieval for computer-assisted test item creation. Educational Testing Service Research Memorandum (RM-07-05). Princeton, NJ: Educational Testing Service.

Higgins, D., Futagi, Y, & Deane, P. (2005). Multilingual generalization of the Model Creator software for math item generation. Educational Testing Service Research Report (RR-05-02). Princeton, NJ: Educational Testing Service.

Hively, W., Patterson, H. L., & Page, S. H. (1968). A "universe-defined" system of arithmetic achievement tests. Journal of Educational Measurement, 5, 275–290.

Irvine, S. H., &Kyllonen, P. C. (2002). Item generation for test development. Hillsdale, NJ: Erlbaum.

LaDuca, A., Staples, W. I., Templeton, B., &Holzman, G. B. (1986). Item modeling procedures for constructing content-equivalent multiple-choice questions. Medical Education, 20, 53–56.

Lai, J., Gierl. M. J., & Alves, C. (2010, April).Using item templates and automated item generation principles for assessment engineering. In R. M. Luecht (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.

Leighton, J. P., & Gierl, M. J. (2011). The learning sciences in educational assessment: The role of cognitive models. Cambridge, UK: Cambridge University Press.

Minsky, M. (1974). A framework for representing knowledge. MIT-AI Laboratory Memo 306.

Mislevy, R. J., &Riconscente, M. M. (2006). Evidence-centered assessment design. In S. M. Downing & T. Haladyna (Eds.), Handbook of test development (pp. 61-90). Mahwah, NJ: Erlbaum.

Reiter, E. (1995). NLG vs. templates. Proceedings of the Fifth European Workshop on Natural Language Generation (pp. 95-105). Leiden, The Netherlands.

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.
Schmeiser, C.B., & Welch, C.J. (2006). Test development. In R. L. Brennan (Ed.), Educational measurement (4th ed., pp. 307-353). Westport, CT: National Council on Measurement in Education and American Council on Education.
Singley, M. K., & Bennett, R. E. (2002). Item generation and beyond: Applications of schema theory to mathematics assessment. In S. H. Irvine & P. C. Kyllonen (Eds.), Item generation for test development (pp. 361-384). Mahwah, NJ: Erlbaum.