Educational Measurement
ISSUES AND PRACTICE

NCME national council on measurement in education

Educational Measurement: Issues and Practice
Fall 2013, Vol. 32, No. 3, pp. 36–50

Instructional Topics in Educational Measurement (ITEMS) Module: Using Automated Processes to Generate Test Items

Mark J. Gierl and Hollis Lai, *University of Alberta*

Changes to the design and development of our educational assessments are resulting in the unprecedented demand for a large and continuous supply of content-specific test items. One way to address this growing demand is with automatic item generation (AIG). AIG is the process of using item models to generate test items with the aid of computer technology. The purpose of this module is to describe and illustrate a template-based method for generating test items. We outline a three-step approach where test development specialists first create an item model. An item model is like a mould or rendering that highlights the features in an assessment task that must be manipulated to produce new items. Next, the content used for item generation is identified and structured. Finally, features in the item model are systematically manipulated with computer-based algorithms to generate new items. Using this template-based approach, hundreds or even thousands of new items can be generated with a single item model.

Keywords: automatic item generation, item model, item development, test development, technology and testing

Automatic item generation (AIG; Embretson & Yang, 2007; Gierl & Haladyna, 2013; Irvine & Kyllonen, 2002) is a rapidly 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 (1969) more than 40 years ago, is gaining renewed interest because it serves as an approach to test development that addresses one of the most pressing and challenging issues facing educators today—the rapid and efficient production of high-quality, content-specific test items. AIG can be characterized as the process of using models to generate items with the aid of computer technology. The role of the test development specialist is critical for the creative task of designing and developing meaningful item models as well as identifying the content required for these models. The role of computer technology is critical for the algorithmic task of systematically combining large amounts of content in each model to produce new items. By combining content expertise and computer technology, testing specialists can create models that yield large numbers of high-quality items in a short period of time.

Significant developments in AIG research and practice have occurred in the last decade (for a historical overview of AIG, see Haladyna, 2013). These developments can be summarized as improvements that occur prior to, during, or after item generation. Research focused on improvements that occur prior to item generation address design-related issues such as cognitive model development (e.g., Embretson & Yang, 2007; Gierl & Lai, 2013; Gierl, Lai, & Turner, 2012), item model development (e.g., Gierl, Zhou, & Alves, 2008; Gierl & Lai, 2012a), and test space designs for AIG (e.g., Bejar et al., 2003; Embretson & Yang, 2007; Huff, Alves, Pellegrino, & Kaliski, 2013; Lai & Gierl, 2013; Luecht, 2013). Research focused on improvements that occur during item generation include many of the technological solutions for AIG (e.g., Gierl et al., 2008; Gütl, Lankmayr, Weinhofer, & Höfler, 2011; Higgins, 2007; Higgins, Futagi, & Deane, 2005; Mortimer, Stroulia, & Yazdchi, 2013), including the use of language-based approaches for item generation that draw on natural language processing and rule-based artificial intelligence (e.g., Gütl et al., 2011), frame-semantic representations (e.g., Deane & Sheehan, 2003, unpublished data; Higgins et al., 2005), and schema theory (e.g., Singley & Bennett, 2002). Research focused on improvements that occur after generation focus on item precalibration methods and the development of statistical models for generated items (e.g., Embretson, 1999; Geerlings, Glas, & van der Linden, 2011; Glas & van der Linden, 2003; Sinharay & Johnson, 2008, 2013; Sinharay, Johnson, & Williamson, 2003). Because of these important research developments, AIG has been used to create millions of new items in diverse content areas, including but not limited to K-12 levels in subjects such as Language Arts, Social Studies, Science, Mathematics (Gierl et al., 2008; Gierl & Lai, 2012a, 2013), and advanced placement (AP) Biology (Alves, Gierl, & Lai, 2010); in psychological domains such as spatial (Bejar, 1990), abstract (Embretson, 2002), figural inductive (Arendasy, 2005), and quantitative reasoning (Arendasy & Sommer, 2007; Embretson & Daniels, 2008; Sinharay & Johnson, 2008, 2013), as well as word fluency (Arendasy, Sommer, & Mayr, 2012), visual short-term memory (Hornke, 2002), and mental rotation (Arendasy & Sommer, 2010); and in licensure and certification testing areas such as nursing, architecture, and medicine (Wendt, Kao, Gorham, & Woo, 2009; Gierl et al., 2008; Gierl, Lai, & Turner, 2012).

Mark J. Gierl, Centre for Research in Applied Measurement and Evaluation, 6-110 Education North, Faculty of Education, University of Alberta, Edmonton, AB, Canada T6G 2R3. E-mail: mark.gierl@ualberta.ca. Hollis Lai, Centre for Research in Applied Measurement and Evaluation, 6-110 Education North, Faculty of Education, University of Alberta, Edmonton, AB, Canada T6G 2R3. E-mail: hollis.lai@ualberta.ca

The purpose of this instructional module is to describe and illustrate a method for generating test items. Our goal is to articulate the logic required for item generation. This logic can then be applied to different AIG approaches to understand the basis of item generation. The approach we describe in this module is *template-based*, meaning that an item model is used 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 must be manipulated to produce new items. To ensure our description is concrete, we illustrate template-based item generation using an example from a medical licensure exam. We use a medical example to highlight the applicability and the generalizability of the AIG approach to a complex problem-solving domain and to demonstrate that AIG is not strictly limited to logical and algorithmic domains like mathematics. A three-step process will be used to guide our description of template-based AIG. In step 1, an item model is developed to specify where content is placed in each generated item. In step 2, the content used for item generation is identified by test development specialists. In step 3, computer-based algorithms are used to place the content specified in step 2 into the item model developed in step 1. *Using this three-step process, hundreds or even thousands of items can be generated using a single item model.*

While AIG provides a solution for generating large numbers of new test items, the psychometric properties (e.g., item difficulty) of these items must still be evaluated. Item quality is typically determined through a field or pilot testing process where each item is administered to a sample of examinees so the psychometric characteristics of the item can be evaluated. This typical solution is often not feasible when thousands of new items have been generated. An alternative method for estimating the psychometric properties of the generated items is with statistical models that permit item *precalibration*. With precalibration, the psychometric properties of the items can be estimated during the item generation process. Unfortunately, a description of precalibration methods is beyond the scope of this module, as our focus is only on the production of new test items. Hence, precalibration methods will not be presented in any detail. For a recent review of these statistical methods, the reader is referred to Sinharay and Johnson (2013).

The Growing Need for Large Banks of Test Items

The principles and practices that guide the design and development of test items are changing because tests are changing. Interdisciplinary forces rooted in areas such as mathematical statistics, the learning sciences, medical education, educational psychology, computing science, and educational technology are now exerting a strong influence on educational measurement theory and practice resulting in different kinds of tests and testing tasks. The first and, probably, most obvious example is in the area of computer-based testing. Educational visionary Randy Bennett (2001) anticipated, more than decade ago, that computers and the Internet would become two of the most powerful forces of change in educational measurement. It is fair to say that his prediction was accurate. Computer-based testing is dramatically changing 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 new types of tests and testing tasks. 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 Graduate Management Achievement Test (GMAT), the Graduate Record Exam (GRE), the Test of English as a Foreign Language (TOEFL iBT), the American Institute of Certified Public Accountants Uniform CPA examination (CBT-e), the Medical Council of Canada Qualifying Exam Part I (MCCQE I), the National Council Licensure Examination for Registered Nurses (NCLEX-RN), and the National Council Licensure Examination for Practical Nurses (NCLEX-PN). Internet-based computerized testing offers many advantages to examinees and examiners compared to more traditional paper-based tests. For instance, computers support the development of innovative item types that allows examiners to use more diverse item formats and measure a broader range of knowledge and skills; items on computer-based tests can be scored immediately thereby providing examinees with instant feedback; computers permit continuous testing and testing on-demand which affords examinees with more flexible administration schedules.

But the advent of computer-based Internet testing has also raised new challenges, particularly in the area of test and item development. Large numbers of items are needed to support the banks necessary for computerized testing because items are continuously administered and exposed. As a result, these 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 2,000-item bank. The costs associated with developing these large banks are severe. 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 can range from $1,500 to $2,500. 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 alone just to develop the item bank for a computer-based test. Clearly, alternative item development methods are needed.

A second example of the need for large numbers of items can be found in the growing body of research on cognitive diagnostic assessment (CDA). A CDA contains items designed to measure the knowledge and skills required to solve specific problems in a particular content area for the purpose of providing examinees with detailed feedback on their problem-solving strengths and weaknesses. Often, a cognitive model is developed so that examinees' problem-solving skills can be included in the interpretations of their test performance. A cognitive model also guides item development so testing tasks that measure these specific skills can be created. A cognitive model in educational measurement refers to a simplified description of human problem solving on standardized tasks at some convenient grain size or level of detail in order to facilitate explanation and prediction of students' performance, including their strengths and weaknesses (Leighton & Gierl, 2007). Typically, the cognitive models used to develop CDAs contain large numbers of skills specified at a fine grain size because these skills must magnify the knowledge, skills, and processes that underlie test performance. Because large

numbers of skills are specified in the cognitive models, large numbers of items must be developed to measure these skills.

Recently, Gierl, Alves, Roberts, and Gotzmann (2009) created cognitive models for diagnosing skills in mathematics for the Preliminary SAT®/National Merit Scholarship Qualifying Test (PSAT). The PSAT is a standardized test created by the College Board that permits students to practice for the SAT Reasoning Test as well as enter the National Merit Scholarship Corporation programs. Three test development specialists created cognitive models in four mathematics content area. In total, 39 cognitive models containing 134 diagnostic skills were required. In other words, to measure PSAT mathematics skills for cognitive diagnosis, 134 items, at minimum, would be needed if the diagnostic test contained one item per skill. Gierl, Alves et al. (2009) recommended that at least three items per skill be created, meaning the PSAT diagnostic assessment would require at least 402 items, to increase the score reliability for each measured skill (see Gierl, Cui, & Zhou, 2009). Large numbers of diagnostic items are required for CDAs because these items are used to identify the specific knowledge, skills, and competencies required to solve problems so examinees can receive detailed feedback on their cognitive problem-solving strengths and weaknesses. Hence, as with the computer-based testing example described earlier, we are faced with an enormous and daunting challenge of creating thousands of new test items.

One way to address the challenge of creating more items is to hire a 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 we know this option is costly. An alternative method for item development that may help address the growing need to produce large numbers of new test items is through the use of AIG. In the next section we describe and illustrate a three-step process for AIG that could prove useful in producing new items to support the evolution and development of our new testing procedures and practices.

Step #1: Item Model Development

Overview

Item models provide the foundation for AIG. Hence, item model development serves as the first step in the process. Item models (Bejar, 1996, 2002; Bejar et al., 2003; LaDuca, Staples, Templeton, & Holzman, 1986) have been described using different terms, including 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 components in an assessment task that can be used for item generation. These components include the stem, the options, and the auxiliary information. The stem contains context, content, item, and/or the question the examinee is required to answer. The option includes a set of alternative answers with one correct option and one or more incorrect options or distracters. Both stem and options are required for multiple-choice item models. Only the stem is created for constructed-response item models. Auxiliary information includes any additional content, in either the stem or option, required to generate an item, including text, images, tables, graphs, diagrams, audio, and/or video. The stem and options can be further divided into elements. An element is the specific variable in an item model that is manipulated to produce new test items. An element is denoted as either a string, which is a non-numeric value or an integer, which is a numeric value. By systematically manipulating elements, new items can be created.

Types of Item Models

Test development specialists have the critical role of designing the item models. The principles, standards, guidelines, and practices used for traditional item development (e.g., Case & Swanson, 2002; Downing & Haladyna, 2006; Schmeiser & Welch, 2006) provide the foundational concepts necessary for creating item models. Some item model examples are also available in the literature (e.g., Bejar et al., 2003; Case & Swanson, 2002; Gierl & Lai, 2013; Gierl et al., 2008). Currently, two types of item models can be created for AIG: 1-layer and n-layer item models.

1-Layer item model. The goal of item generation using the 1-layer item model is to produce new assessment tasks by manipulating a relatively small number of elements in the model. This type of item model currently dominates the practical applications in AIG. Often, the starting point is to use a parent item. The parent can be found by reviewing items from previous test administrations, by drawing on a bank 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 through experience, practice, and intuition, an item model is created by identifying elements in the parent item that can be manipulated to produce new items. If the purpose of AIG is to generate statistically precalibrated items, then the test development specialist's task is to manipulate those elements in the parent that yield generated items with predictable and, often, similar psychometric characteristics. Generated items with comparable psychometric characteristics (e.g., similar difficulty levels) are isomorphic. Alternatively, if the purpose is to generate items that will not be statistically precalibrated (in this case, the generated items will need to be field tested), then the test development specialist is free to manipulate those elements expected to yield large numbers of instances of the parent item through the generative process. Generated items with different psychometric characteristics are also known as variants.

One disadvantage of using a 1-layer item model for AIG is that relatively few elements can be manipulated. The manipulations are limited because the number of potential elements in a 1-layer item model is relatively small (i.e., the number of elements are fixed to the total number of elements in the stem). Unfortunately, by restricting the element manipulations to a small number, the generated items may have the undesirable quality of appearing too similar to one another. In our experience, isomorphic items generated from 1-layer item models are referred to pejoratively by many test development specialists as "clones," "ghost" items or "Franken-items." Clones are often perceived to be generated items that are simplistic, easy to produce, and easy to detect.

One attempt to address the problem of generating isomorphic items was described by Gierl et al. (2008). They developed a taxonomy of 1-layer *item model types*. The purpose of this taxonomy was to provide test development specialists with design guidelines for creating item models that yield diverse types of generated items. Gierl et al.'s (2008) strategy

for promoting diversity was to systematically combine and manipulate those elements in the stem and options typically used for item model development. According to Gierl et al. (2008), the elements in the stem can function in four different ways. *Independent* indicates that the elements in the stem are unrelated to one another. Hence, a change in one stem element will not affect the other stem elements. *Dependent* indicates all element in the stem are related to one other. A change in one stem element will affect the other stem elements *Mixed* includes independent and dependent elements in the stem, where at least one pair of stem elements is related. *Fixed* represents a constant stem format with no variation. The elements in the options can function in three different ways. *Randomly selected* options refer to the manner in which the distracters are selected, presumably, from a list of possible alternatives. The distracters in this case are selected randomly. *Constrained* options mean that the keyed option and the distracters are generated according to specific constraints, such as algorithms, rules, formulas, or calculations. *Fixed* options occur when both the keyed option and distracters are fixed and therefore do not change across the generated items. A matrix of 1-layer item model types can then be produced by crossing the four different elements in the stem and the three different elements in the options. Gierl et al. claimed that the taxonomy is useful because it provides the guidelines necessary for designing diverse 1-layer item models by outlining their structure, function, similarities, and differences. It can also be used to ensure that test development specialists do not design item models where the same elements are constantly manipulated or where the same item model structure is frequently used.

Figure 1 contains an example of a medical parent item used to evaluate an examinee's ability to diagnose and treat hernias. This example will be used throughout the module to demonstrate the concepts in the three-step AIG process. This item was initially developed by a team of surgical content specialists for a medical licensure exam. Figure 2 contains the item model for the parent item presented in Figure 1. For this 1-layer surgical item model, the stem contains one integer (AGE) and six strings (GENDER; PAIN; LOCATION; ACUITYOFONSET; PHYSICALFINDINGS; WBC). Using the Gierl et al. (2008) taxonomy, this item model would be described as a dependent stem with constrained options. The integer and string elements in the stem are dependent because the values they assume will depend on the combination of content in the item model (this point will be explained in more detail in the section on step #2, " Identifying Content for Item Models"). The options are constrained by the combination of integer and string values specified in the stem.

n-Layer item models. The second type of item model can be described as *n*-layer (Gierl & Lai, 2012b). The goal of AIG using the *n*-layer item model is to produce items by manipulating a relatively large number of elements at two or more levels in the model. Much like 1-layer item modeling, 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 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 high. 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 syntactic 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 (hence, generation is described as nonlinear). 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-layer and *n*-layer item model is presented in Figure 3. For this example, the 1-layer model can provide a maximum of four different values for element A (see left-hand side of the figure). Conversely, the *n*-layer model can provide up to 64 different values by embedding the same four values for elements C and D within element B (see right-hand side of figure). 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 always increase the number of items that can be generated relative to a 1-layer structure.

One important advantage of using an *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 development specialists, because large numbers of systematic manipulations are occurring in each model thereby promoting heterogeneity in the generated items. The disadvantage of using an *n*-layer structure is that the models are complex and therefore challenging to create. Also, the effect of embedding elements, while useful for generating large numbers of diverse items, will make it challenging to predict the psychometric characteristics of the generated items. Hence, *n*-layer item modeling may yield items that are not possible to precalibrate and, therefore, these items will need to be field-tested using more conventional administration procedures.

An *n*-layer surgical item model is presented in Figure 4. This example helps illustrate how the structure of the item can be manipulated to produce more diverse generated items. In addition to manipulating the integer and string values, as with the 1-layer example, we now embed the integers and strings within one another to facilitate the generative process. That is, by embedding elements within elements, different question prompts, test findings, and situations can be used, thereby producing more heterogeneous items. For the *n*-layer example in Figure 4, two types of layers are used. The first type is sentence presentation. The last sentence of the item stem in Figure 2, for instance, can serve as an element by re-wording the phrase, "Which of the following is the next best step?" to "Which one of the following is the best prognosis?" or "Given this information, what is the best course of action?" The second type is sentence structure. Four alternative sentence structures can be used to present the

A 24-year-old man presented with a mass in the left groin. It appeared suddenly 2 hours ago while lifting a piano. On examination he has a tender firm mass in the left groin. Which one of the following is the next best step?

a. Immediate hernia repair
b. Needle aspiration
c. Ice packs to groin
d. Reduction of mass

FIGURE 1. Item used to measure examinees' ability to diagnose a hernia.

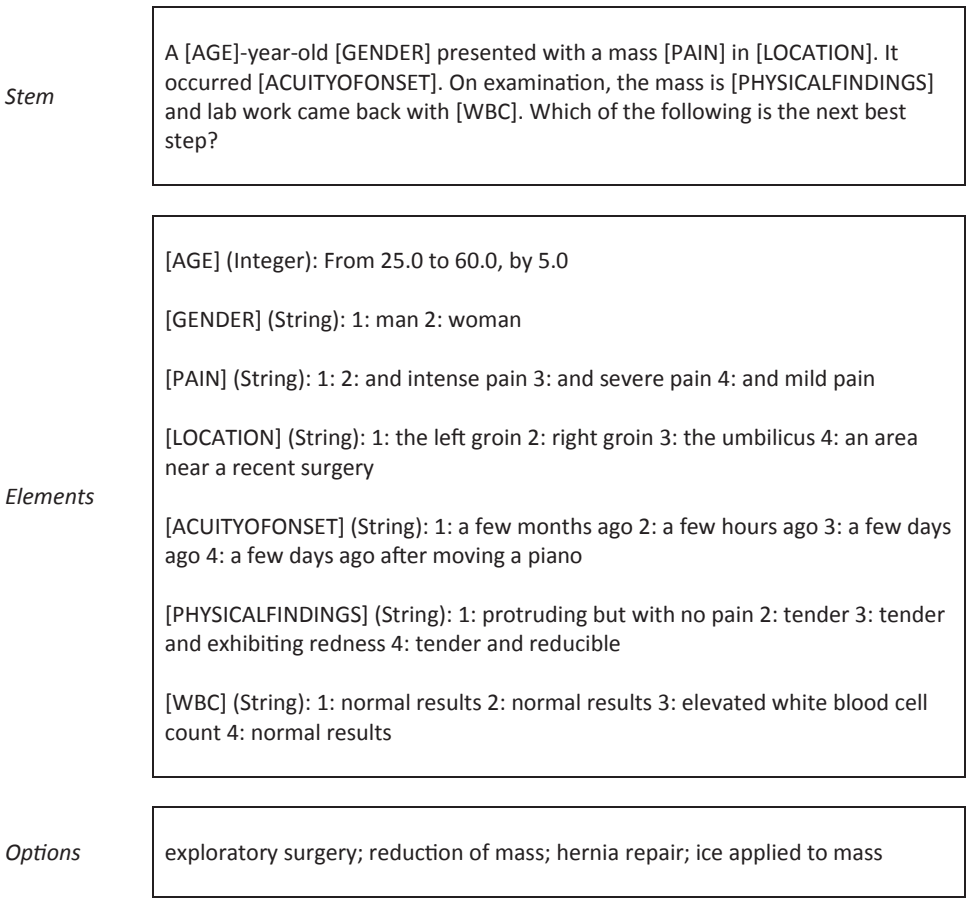

FIGURE 2. 1-layer item model for the surgery example.

information in the original sentence from the 1-layer item model "On examination, the mass is [PHYSICALFINDINGS] and lab work came back with [WBC]." For instance, three alternative structures are "Upon further examination, the patient had [WBC] and the mass is [PhysicalFindings]," "With [WBC] and [PhysicalFindings] in the area, the patient is otherwise nominal," and "There is [PhysicalFindings] in the [Location] and the patient had [WBC]." In short, by introducing new layers of elements such as sentence presentation and structure, more diverse surgical items can be generated with n-layer item models compared to the 1-layer approach.

Step #2: Identifying Content for Item Models

Overview

Once the item model is created, test development specialists must then identify the content that will be used in the model to produce new items. Drasgow, Luecht, and Bennett (2006) advised test development specialists to engage in the challenging task of content specification using a combination of design guidelines and principles discerned from experience, theory, and research. They proposed two general approaches: weak and strong theory.

Weak Theory AIG

The first approach described by Drasgow et al. (2006) for identifying item model content is with a weak theory approach. This approach is synonymous with our description for the 1-layer item model presented earlier in this module. Using weak theory, design guidelines are used to create item models that generate isomorphic instances using, as a starting point, parent items that highlight the underlying structure of the model. The test developer's task is to manipulate those elements in the parent that will yield new items. But, as we noted earlier, the drawback of using weak theory for 1-layer

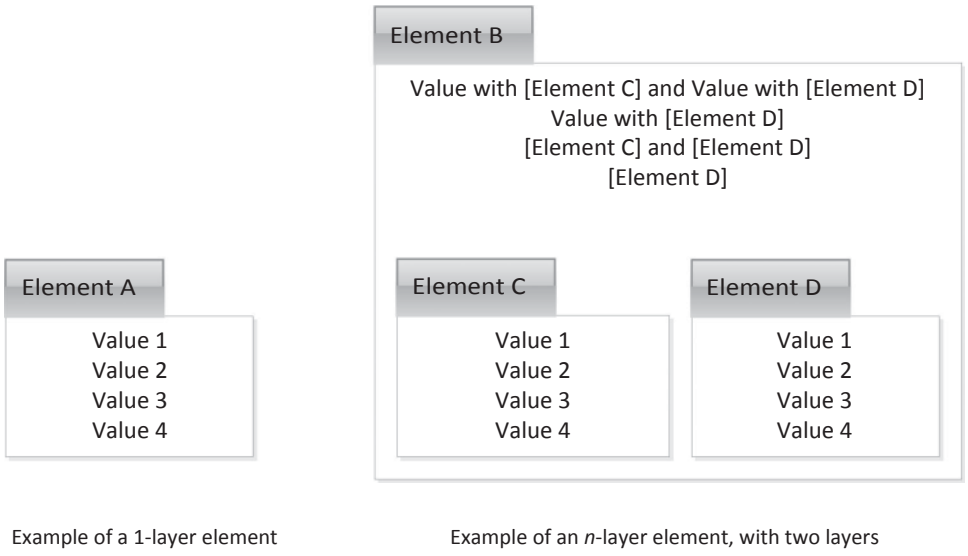

FIGURE 3. An comparison of the elements in a 1-layer and *n*-layer item model.

item model development is that relatively few elements can be manipulated. Weak theory can, in principle, be used with 1- and *n*-layer item models, but the majority of applications to-date have focused on 1-layer applications.

Strong Theory AIG

Another way to identify the content for an item model is with strong theory. Drasgow et al. (2006) describe strong theory as the process where a cognitive model is used to specify and manipulate those elements that affect the difficulty level of the generated items using a theoretical account of test performance. Cognitive theory helps highlight both the examinees' knowledge and skills required to solve the item as well as the content features in the item that affect difficulty. Drasgow et al. claimed that by modeling the interaction between the examinee and the content, it is possible to predict and therefore control the psychometric characteristics of the generated items. This approach, when successful, has the added benefit of yielding a strong inferential link between the examinees' test item performance in a specific content area with the interpretation of the examinees' test score because the model features that elicit the item characteristics are predictive of test performance (Bejar, 2013).

To date, the use of strong theory for AIG has focused on the psychology of specific response processes using articulate, 1-layer items model in areas such as spatial reasoning (Bejar, 1990), abstract reasoning (Embretson, 2002), figural inductive reasoning (Arendasy, 2005), and mental rotation (Arendasy & Sommer, 2010). Unfortunately, few comparable cognitive theories exist to guide our item development practices (Leighton & Gierl, 2011) or to account for test performance in broad content areas, typical of those found on most educational achievement tests (Ferrara & DeMauro, 2006; Schmeiser & Welch, 2006) or licensure and certification examinations (Clauser, Margolis, & Case, 2006). To illustrate the use of a strong theory approach with an *n*-layer item model in a broad content area, Gierl, Lai, and Turner (2012) introduced the concept of a cognitive model for AIG in the area of medical testing. We apply the logic described by Gierl, Lai, and Turner (2012) to our current surgical example.

Figure 5 contains a cognitive model for AIG required to diagnose and treat complications with hernias. This cognitive structure outlines the knowledge and skills required to make medical diagnostic inferences in order to treat the problem outlined in this case. It is presented in three panels. The top panel identifies the problem and its associated scenarios. The middle panel specifies the relevant sources of information. The bottom panel highlights the salient features, which includes the elements and constraints, within the relevant sources of information specified in the middle panel.

To generate items using a strong theory approach, we use the content and structure specified in the Figure 5 cognitive model to produce new test items. Four different hernia scenarios are used in this surgical example: asymptomatic incarcerated (AI), painful incarcerated (RI), strangulation (S), and reducible symptomatic (RS). Next, four sources of information are specified for this problem: Patient presentation, location, physical examination, and laboratory results. Finally, the salient features within the sources of information are highlighted. In our example, Patient Presentation had three features: acuity of onset, pain, and nausea and vomiting. Each feature, in turn, contains two nested components. The first nested component for a feature is the element. Elements contain content specific to each feature that can be manipulated for item generation. For the acuity of onset feature of Patient Presentation, range of time is the element. This element has four values: months or years, $\pm$ 6 hours over days, more than 6 hours, or any time. The second nested component for a feature is the constraint. Each element is constrained by the scenarios specific to this problem. For example, asymptomatic incarcerated hernias (AI) are associated with months or years in the acuity of onset feature for the patient presentation source of information (i.e., AI: Months-Years at the top left side of the Features panel in Figure 5).

The content presented in the cognitive model for AIG in Figure 5 serves two different purposes. The first purpose of the cognitive model is concrete. The cognitive model guides the computer-based algorithms described in step #3 so that new items can be assembled. Therefore, one important purpose of the cognitive model is to link the problem (diagnosing and treating hernias) and the associated scenarios

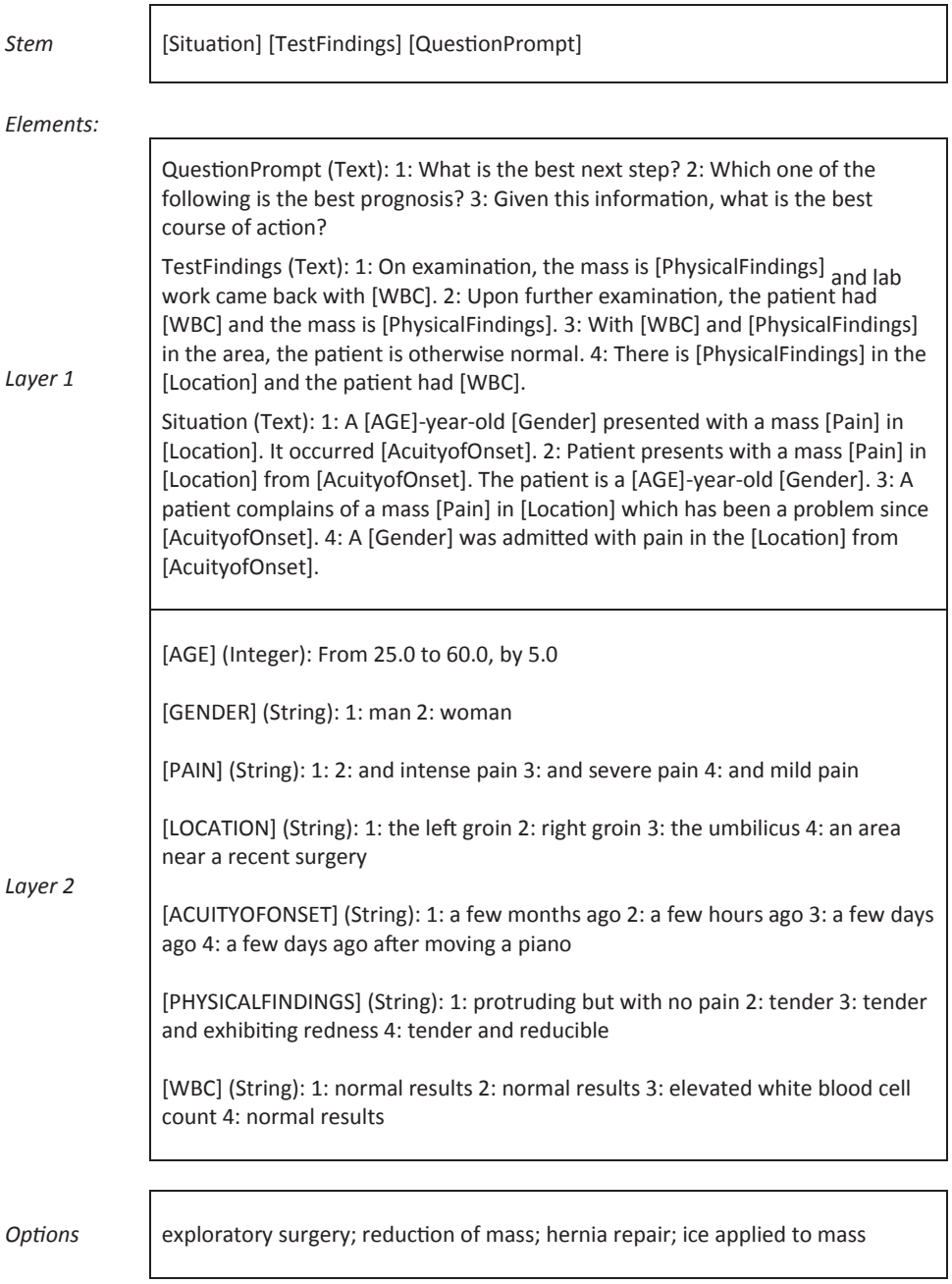

FIGURE 4. *n*-Layer item model for the surgery example.

(asymptomatic incarcerated; painful incarcerated; strangulation; RS) to the features (acuity of onset; pain; nausea and vomiting; groin pain; umbilicus pain; scars; tenderness; reducible; redness; white blood cell count) through the sources of information (patient presentation; location; physical examination; laboratory test results). These prescriptive links are used for item generation, as the features can be inserted in their appropriate information sources, as outlined in Figure 5, subject to the elements and their constraints.

The second purpose of the cognitive model is more abstract. It serves as an explicit representation of the problem-solving knowledge and skills required to diagnose and treat hernias. Norman, Eva, Brooks, and Hamstra (2006) claimed that *problem representation* was an important way to organize and study the content and processes required for expert medical reasoning and problem solving. Unlike the weak theory approach where the manipulation of elements in the model must be discerned through the guidelines, judgments, and experiences of the test development specialists, a strong theory approach provides a cognitive model and some associated design principles for generating items (Bejar, 2013). The cognitive model in Figure 5 was created by medical content specialists thereby serving as a representation of how they think about and solve problems related to hernias (see Gierl, Lai, & Turner, 2012). Two content specialists, who were both experienced medical examination item writers and practicing physicians, were asked to describe the knowledge and clinical-reasoning skills required to solve items on a medical licensure exam for the surgery content area. Their knowledge and skills were identified in an inductive manner using a

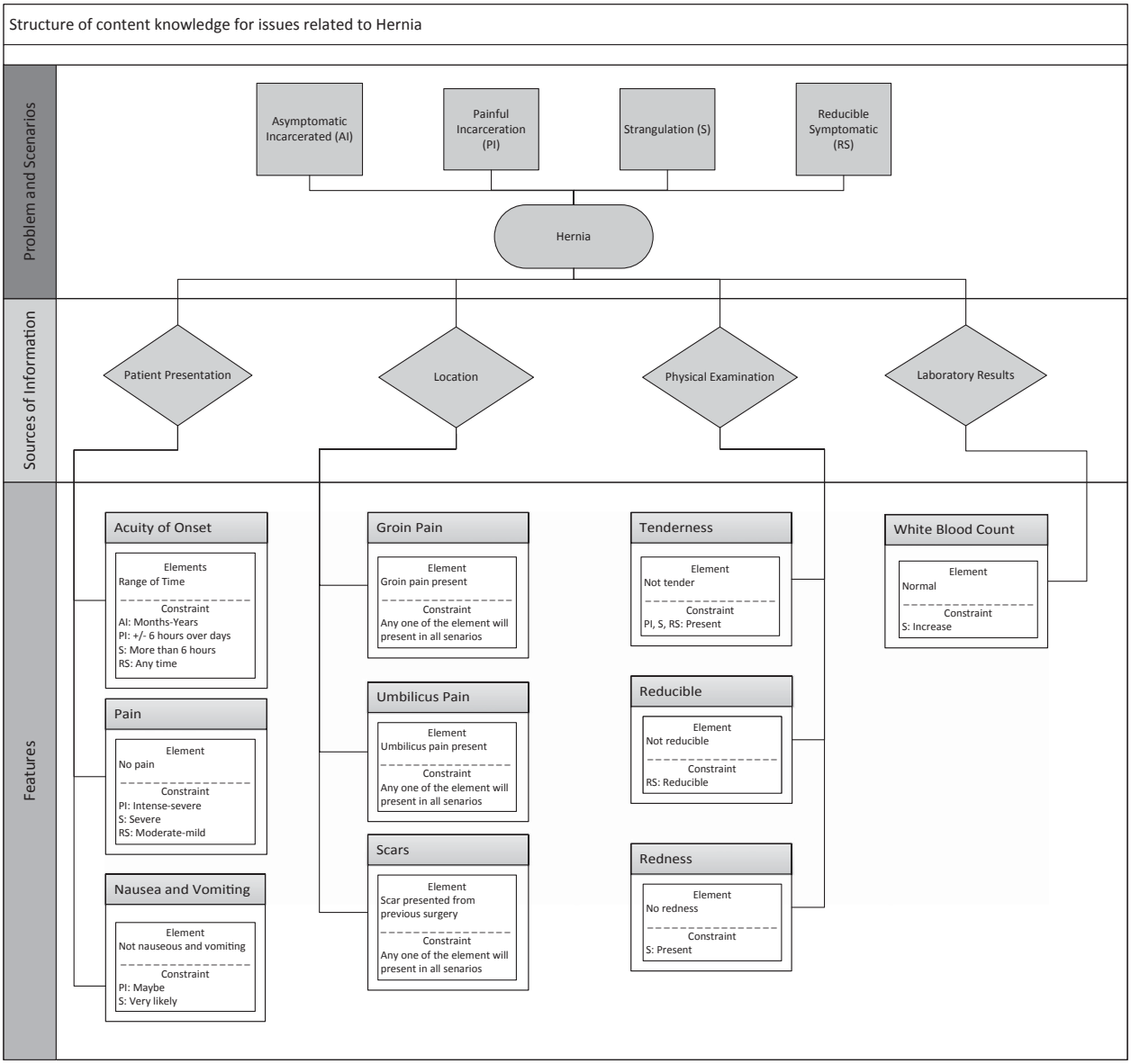

FIGURE 5. Cognitive model for AIG using surgery example.

verbal reporting method. That is, the content specialists were given an existing multiple-choice item and asked to identify and describe the key information that would be used to solve the item. This representation was documented as a cognitive model and then used to guide the detailed rendering process needed for item generation. The content and structure in Figure 5 was used to create the 1- and *n*-layer item models in Figures 2 and 4, respectively. Strong theory can be used with 1-layer and n-layer item models. However, strong theory has rarely been used, to-date, for AIG and, when it is used, the application tends to focus on a 1-layer model in psychological domains where articulate cognitive models exist.

Step #3: Generate Items Using Computer Technology and Evaluting Item Similarity

Overview

Once the item models are created and the content for these models has been identified by the 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 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 *ModelCreator* 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. Gütl et al. (2011) outlined the use of the *Enhanced Automatic Question Creator (EAQC)* to extract key concepts from text in order to generate multiple-choice and constructed-response test items. For this module, we illustrate the use of technology for generating test items using the IGOR software described by Gierl et al. (2008). The purpose of this illustration is simply to highlight the logic of how computer technology supplements content expertise to facilitate item generation. But it is also important to note that any linear programming method can be used to solve the type

of combinatorial problem found within AIG—IGOR is just one of many possible solutions available to researchers and practitioners. IGOR, which stands for **I**tem **G**enerat**OR**, is a JAVA-based program designed to assemble the content specified in an item model, subject to elements and constraints articulated in the cognitive model. The logic behind IGOR is straightforward: Iterations are conducted to assemble all possible combinations of elements and options, subject to the constraints. Without the use of constraints, all of the variable content (i.e., values for the integers and strings) would be systematically combined to create new items. Unfortunately, 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. For instance, asymptomatic incarcerated hernias (AI) are constrained by time because this type of hernia is associated with months or years in the acuity of onset feature (i.e., AI: Months-Years at the top left-hand side of the Features panel in Figure 5, example).

Item Generation with IGOR

To begin, IGOR reads an item model in the form of an XML (Extensible Markup Language) file. The content for the item model is formatted according to the same structure shown in Figures 2 and 4 (i.e., stem, elements, options). The Item Model Editor window permits the programmer to enter and structure each item model. The editor has three panels. The stem panel is where the stem for the item model is specified. The elements panel is used to manipulate the integer and string variables as well as to apply the constraints highlighted in the cognitive model. The options panel is used to specify the correct and incorrect alternatives. The options are classified as either a key or a distracter. The Elements and Options panels contain three editing buttons. The first edit button allows the user to add a new element or option. The second edit button is used to modify the current element or option. The third edit button removes the selected element or option from the model.

To generate items from a model, the Test Item Generator dialogue box is presented where the user specifies the item model file, the test bank output file, the answer key file, a portfolio output, and the Generator options (see Figure 6). For the current example, the item model file is loaded from the current item model which is specified as an XML file. For the test bank output file, the user selects the desired location for the generated items. The user can also save a separate key under the answer key option. The Portfolio is used to generate a file containing all IGOR input as well as a sample of the generated item output. Portfolio Size refers to the number of generated items that will be included in the portfolio and the location of the portfolio output is specified in the "Save to" location. Finally, the user can specify Generator options. These options include size of the generated item bank, the order of the options, and the number of options for each generated item. Once the files have been specified in the Test Item Generator dialogue box, the program can be executed by selecting the "Generate" button (see bottom right-hand side).

Using IGOR with the 1-layer surgery item model presented in Figure 2, 256 items were generated. A random sample of four items is presented in Table 2. When IGOR was used with the *n*-layer surgery item model presented in Figure 4, 16,384 items were generated. A random sample of four items is presented in Table 1.

Table 1. Random Sample of Four Generated Items Using *n*-Layer Surgery Item Model

4610. A patient complains of a mass in the left groin which has been a problem for a few months. On examination, the mass is protruding but with no pain and lab work came back with normal results. Which one of the following is the best treatment?

a. reduction of mass
b. exploratory surgery
c. hernia repair
d. ice applied to mass[a]

5326. A 50-year-old man presented with a mass and mild pain in the left groin. It occurred a few days ago after moving a piano. Upon further examination, the patient has normal results and the mass is tender and reducible. Which one of the following is the best treatment?

a. exploratory surgery
b. reduction of mass
c. hernia repair[a]
d. ice applied to mass

7325. A 45-year-old man presented with a mass and severe pain in right groin. It occurred a few days ago. There is tenderness and redness in the right groin and the patient has an elevated white blood cell count. Which one of the following is the best treatment?

a. reduction of mass
b. hernia repair[a]
c. ice applied to mass
d. exploratory surgery

12010. A patient complains of a mass and mild pain in the umbilicus which has been a problem since moving a piano a few days ago. The umbilicus is tender and reducible and the patient has normal results. Given this information, what is the best course of action?

a. exploratory surgery
b. ice applied to mass
c. reduction of mass[a]
d. hernia repair

[a]Correct option.

Evaluating the Comparability of Generated Items

To measure the similarity of the generated items, the intra-model differences, meaning items generated within the same item model, must be evaluated. Similarity can be quantified using the cosine similarity index (CSI). The CSI is a measure of similarity between two vectors of co-occurring texts. It is computed using the cosine of the angle between the two vectors in a multidimensional space of unique words. In other words, the CSI provides a global summary of the text similarity in a sample of the generated items. The CSI is given as

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

where A and B are two items expressed in a binary vector of word 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 place A and B in a vector format so the words can be compared, the occurrence of each word in the vector list is quantified with a value of 1. The resulting vectors for 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

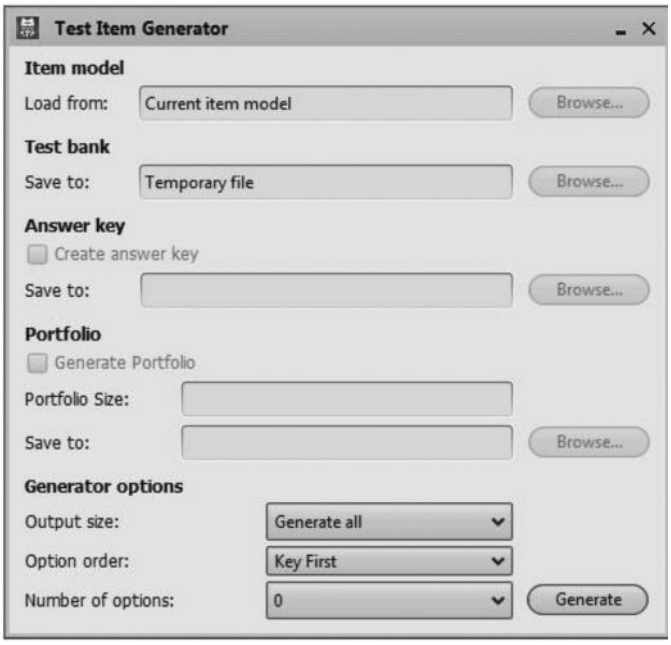

FIGURE 6. IGOR interface illustrating the generating functions.

overlapped between the two vectors, and a maximum of 1, meaning that the words represented by the two vectors are identical.

To compute the CSI, the items are compiled into a matrix of word occurrences for each item model, where the row represents a vector of a generated item, the column represents a unique word in the pool of generated items, and the row-by-column cell is numerated dichotomously to determine whether a given item contains a given word. The CSI is calculated for each unique item pair within the same item model resulting in a CSI mean and standard deviation for each model. The CSI is one type of word comparison methods among many that could be used to evaluate the similarity of the generated items (e.g., a word net or n-gram comparison could also be used). Becker and Kao (2009) demonstrated how the CSI could be used to detect stolen items from a test bank thereby providing a precedent for using this measure to evaluate item similarity.

To illustrate the use of the CSI in our example, a random sample of 100 items from each model was selected and analyzed. Because fewer elements are manipulated with the 1-layer compared to the *n*-layer approach, similarity should be higher for the 1-layer generated items. The 1-layer surgery item produced CSI values that ranged from 0.53 to 0.98 across the 4,950 pairwise comparisons conducted for this example, with an overall mean of 0.74 and a standard deviation of 0.11. Because the CSI ranges from 0 (no similarity) to 1 (perfect similarity),a high mean for the 1-layer model indicates that the generated items are quite similar while the low standard deviation reveals the items are relatively homogeneous. The *n*-layer model produced CSI values across the pairwise comparisons ranging from 0.17 to 1.00, with a comparatively lower mean of 0.53 and a higher standard deviation of 0.16. These results allow us to conclude that the *n*-layer item models do, in fact, produce more heterogeneous and diverse items compared to the items generated from 1-layer item models using a measure of text similarity.

Directions for Future Research in Automatic Item Generation

We outlined and illustrated one template-based method for generating test items. Three areas of future AIG research are also described before we present our conclusions.

Research on Item Model Development

Additional research on item models is needed, given that item models provide the foundation for AIG. Hence, the theory and practices that underlie item model development must be studied more systematically. Currently, there is little research, to our knowledge, on this topic. Research on item model development can be guided, in part, by the principles and practices that direct research on traditional test items. But item models are not test items and, as a result, the existing principles and practices that guide traditional item development will have limited utility. Therefore, research is needed on methods and procedures for designing and developing 1- and *n*-layer items models within both a weak and strong theoretical framework. Research must also be conducted to evaluate the properties of these models by focusing on their generative capacity (i.e., the number of items that can be generated from a single item model) as well as their generative veracity (i.e., the accuracy and usefulness of the generated items) so the strengths and weaknesses of different modeling procedures can be discerned.

Using n-*Layer Item Models for Multilingual Assessment*

The *n*-layer model is a flexible structure for item generation thereby permitting many different but feasible combinations of embedded elements. It can be used with any type of template-based item generation method. It can be used to generate different item types. And, as we illustrated, the n-layer models can accommodate a wide range of elements producing more heterogeneous and diverse items compared with the 1-layer approach. In addition to generating more

Table 2. Random Sample of Four Generated Items Using *n*-Layer Surgery Item Model

11. A 35-year-old woman presented with a mass in the left groin. It occurred a few months ago. On examination, the mass is protruding but with no pain and lab work came back with normal results. Which of the following is the next best step?
    a. ice applied to mass[a]
    b. exploratory surgery
    c. reduction of mass
    d. hernia repair
50. A 30-year-old man presented with a mass in an area near a recent surgery. It occurred a few months ago. On examination, the mass is protruding but with no pain and lab work came back with normal results. Which of the following is the next best step?
    a. ice applied to mass[a]
    b. exploratory surgery
    c. reduction of mass
    d. hernia repair
137. A 25-year-old woman presented with a mass and severe pain in the left groin. It occurred a few days ago. On examination, the mass is tender and exhibiting redness and lab work came back with an elevated white blood cell count. Which of the following is the next best step?
    a. ice applied to mass
    b. exploratory surgery
    c. reduction of mass
    d. hernia repair[a]
175. A 55-year-old woman presented with a mass and severe pain in the umbilicus. It occurred a few days ago. On examination, the mass is tender and exhibiting redness and lab work came back with an elevated white blood cell count. Which of the following is the next best step?
    a. ice applied to mass
    b. exploratory surgery
    c. reduction of mass
    d. hernia repair[a]

[a]Correct option.

variable items, one possible application of *n*-layer modeling may be in generating multilingual test items. Different languages require a different grammatical structure and word order (Higgins, Futagi, & Deane, 2005). With a 1-layer model, the grammatical structure and word order cannot be easily or readily manipulated because the generative operations are constrained to a small number elements at a single level. However, with the use of an *n*-layer model, the generative operations are expanded dramatically to include a large number of elements at multiple levels. Language, therefore, can serve as an additional layer that is manipulated during item generation. Figure 7 shows an embedded element structure that could be used to generate items in English and Spanish for our surgery item model. One important direction for future research, then, is to use *n*-layer item modeling to generate tasks in multiple languages by adding language as an additional layer in the model.

Using Measures of Item Similarity to Guide Statistical Precalibration

By incorporating methods from computational linguistics, we demonstrated how the CSI can serve as a measure for summarizing text similarity among the generated items. Item models, like items, require descriptive measures for their proper use. The CSI has many potential applications, but one of the most promising areas of application can be found in statistical precalibration. Sinharay et al. (2003) and Sinharay and Johnson (2008), for example, used the concept of item families to develop a statistical model for calibrating generated items. The siblings (i.e., generated items) in their statistical model must share some common features to be considered part of the family. Unfortunately, there are few empirical methods available for quantifying commonality with generated test items and, hence, sibling membership must be established more subjectively using judgements and ratings from test development specialists. As an empirical measure of sibling commonality, the mean CSI among the generated item within an item model family could be used to describe similarity. The CSI outcomes could then be used as evidence to decide whether an item model has generated clones or isomorphs (i.e., high mean CSI and low standard deviations) suitable for family membership or whether it has generated variants (i.e., low mean CSI and high standard deviation) which are unsuitable for a specific family membership. Hence, future studies should be conducted to evaluate the effectiveness of using the CSI for establishing membership for precalibration methods that require item families.

Conclusions

Testing agencies require large numbers of high-quality items that are produced in a timely and cost-effective manner. Computer-based testing and CDA serve as two examples where an abundant supply of test items is required to implement new and innovative testing methods. Test development specialists may find that AIG helps address some of these item development challenge. AIG is the process of using item models to generate test items with the aid of computer technology. The template-based AIG approach we described in this module requires three steps. First, test development specialists create item models that specify the elements in the assessment task that must be manipulated. Second, the content used for item generation is identified and structured by test development specialists. Third, elements in the item model are manipulated with computer-based algorithms to produce new items. Using this three-step process, hundreds or even thousands of new items can be generated using a single item model. Not surprising, AIG is seen by some researchers and practitioners as a "dream come true," given the laborious processes and high costs required for traditional item development.

AIG has three important benefits compared to the more traditional approach to test development. First, AIG can be used to develop item banks. The purpose of item modeling is to create a single model that yields many test items. Multiple models can then be created to supply item banks with thousands of new test items. With this approach, item exposure through test administration is minimized, even with continuous testing, because a large bank of operational items is available. Second, AIG may lead to more efficient and cost-effective item production because item models can be re-used. Because each model produces many items, these models can be used continuously as long as the exposure rates for the generated items are monitored. With a more traditional approach, each item is unique and, thus, each item is produced

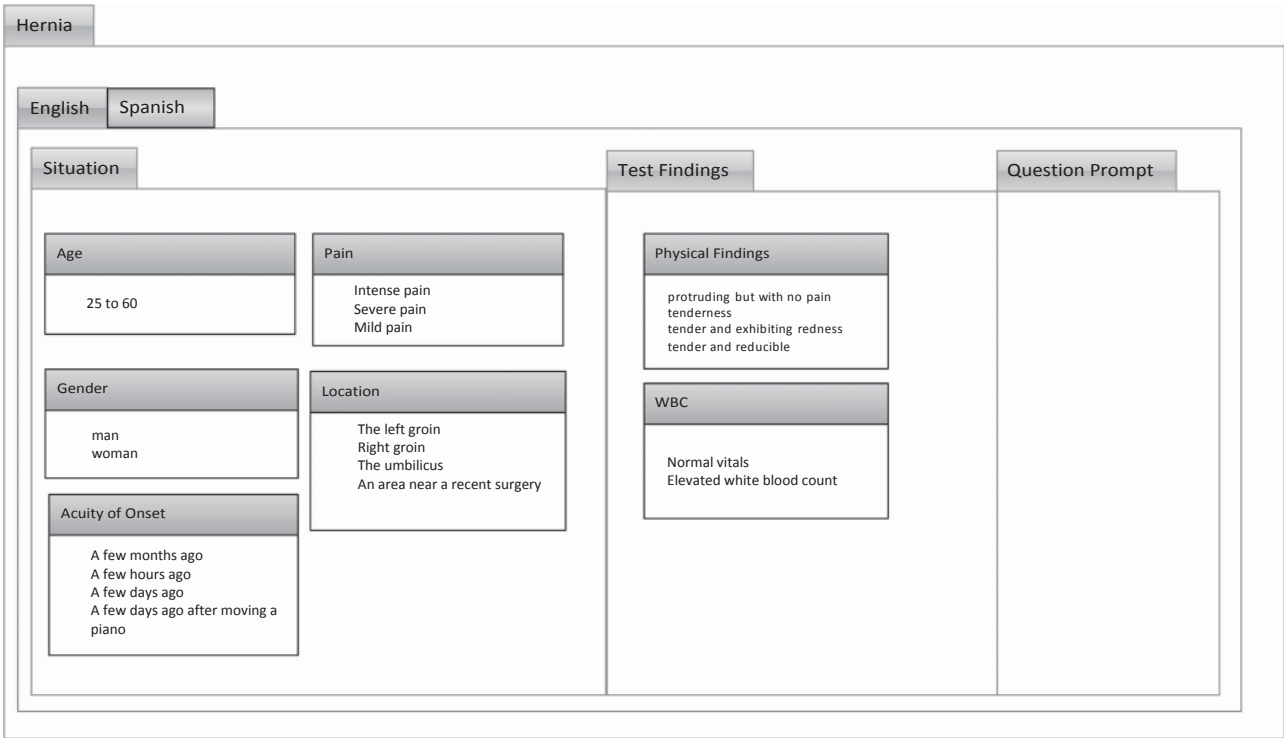

FIGURE 7. *n*-Layer surgery item model with language as a layer.

in a one-at-a-time manner. As a result, item use is typically limited to either a single or a relatively small number of administrations. Item model re-use may also help minimize errors that can occur in item development, such as including or excluding words, phrases, or expressions along with spelling, grammatical, punctuation, capitalization, typeface, and formatting problems (Schmeiser & Welch, 2006) because only a small number of specific elements in the stem and options are manipulated when generating new items. Third, AIG is a scalable process because it treats the item model as the unit of analysis. As a result, one item model can generate many test items. With a more traditional approach, the test item is the unit of analysis with each item being created individually. Because of this unit of analysis difference, the cost per item should decrease because test development specialists are producing models that yield multiple items rather than producing single unique items. AIG is also a scalable production process because one model yields many items.

When these three benefits are considered as a whole, AIG could be characterized as a shift away from the "art" of item development where assessment tasks are created solely from test developer expertise, experience, and judgment toward a new "science" of item development where these tasks are created by combining the knowledge and skills of the test development specialists with the algorithmic power of a computer. But it is important to add that, in our view, this new science of AIG does not diminish the role of the test development specialist. Rather, it helps focus the developer's role on the creative task of identifying, organizing, and evaluating the content needed to develop test items. The test development specialist is essential in AIG for identifying the knowledge and skills required to think about and solve problems, organizing this information into a cognitive model, and designing meaningful item models. We often associate these activities with the art of test development because they require judgement, expertise, and experience. The role of computer technology in AIG is required for the generative task of systematically combining large amounts of information in each item model. We often associate these activities with the science of modern computing. By merging the outcomes from the content-based creative task with the technology-based generative task, automated processes can be used to facilitate and promote a new approach to item development.

Acknowledgements

We would like to thank the Medical Council of Canada, the College Board, and CTB/McGraw-Hill for supporting 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 Medical Council of Canada, the College Board, or CTB/McGraw-Hill.

We would also like to thank ITEMS editor, Dr. W. Holmes Finch, for his invaluable comments and suggestions.

References

Alves, C., Gierl M. J., & Lai, H. (2010, April). *Using automated item generation to promote principled test design and development*. Paper presented at the meeting of the American Educational Research Association, Denver, CO.

Arendasy, M. E. (2005). Automatic generation of Rasch-calibrated items: Figural matrices test GEOM and endless loops test E^c. *International Journal of Testing*, *5*, 197–224.

Arendasy, M. E., & Sommer, M. (2007). Using psychometric technology in educational assessment: The case of a schema-based isomorphic approach to the automatic generation of quantitative reasoning items. *Learning and Individual Differences*, *17*, 366–383.

Arendasy, M. E., & Sommer, M. (2010). Evaluating the contribution of different item features to the effect sixe of the gender differences in

three-dimensional mental rotation using automatic item generation. *Intelligence*, *38*, 574–581.
Arendasy, M. E., Sommer, M., & Mayr, F. (2012). Using automatic item generation to simultaneously construct German and English versions of a word fluency test. *Journal of Cross-Cultural Psychology*, *43*, 464–479.
Becker, K. & Kao, S.(2009, April). *Finding stolen items and improving item banks*. Paper presented at the meeting of the American Educational Research Council, San Diego, CA.
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: Lawrence Erlbaum.
Bejar, I. I. (2013). Item generation: Implications for a validity argument. In M. J. Gierl & T. Haladyna (Eds.), *Automatic item generation: Theory and practice* (pp. 40–55). New York, NY: Routledge.
Bejar, I. I., Lawless, R., Morley, M. E., Wagner, M. E., Bennett, R., & Revuelta, J. (2003). A feasibility study of on-the-fly item generation in adaptive testing. *Journal of Technology, Learning, and Assessment*, *2*(3), 1–55. 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.
Bormuth, J. (1969). *On a theory of achievement test items*. Chicago, IL: University of Chicago Press.
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* (pp. 247–266), New York, NY: Springer.
Case, S. M., & Swanson, D. B (2002). *Constructing written test questions for the basic and clinical sciences* (3rd ed.). Philadelphia, PA: National Board of Medical Examiners.
Clauser, B., Margolis, M. J., & Case, S. M. (2006). Testing for licensure and certification in the professions. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 701–731). Washington, DC: American Council on Education.
Downing, S. M., & Haladyna, T. M. (2006). *Handbook of test development*. Mahwah, NJ: Lawrence Erlbaum.
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. (1999). Generating items during testing: Psychometric issues and models. *Psychometrika*, *64*, 407–433.
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: Lawrence Erlbaum.
Embretson, S. E., & Daniels, R. C. (2008). Understanding and quantifying cognitive complexity level in mathematical problem solving items. *Psychological Science Quarterly*, *50*, 328–344.
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). Oxford, UK: Elsevier North Holland.
Ferrara, S., & DeMauro, G. E. (2006). Standardized assessment of individual achievement in K-12. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 579–621). Westport, CT: National Council on Measurement in Education and American Council on Education.
Geerlings, H., Glas, C. A. W., & van der Linden, W. J. (2011). Modeling rule-based item generation. *Psychometrika*, *76*, 337–359.
Gierl, M. J., Alves, C., Roberts, M., & Gotzmann, A. (2009, April). *Using judgments from content specialists to develop cognitive models for diagnostic assessments.* Paper presented in symposium *How to Build a Cognitive Model for Educational Assessments*, conducted at the meeting of the National Council on Measurement in Education, San Diego, CA.
Gierl, M. J., Cui, Y., & Zhou, J. (2009). Reliability of attribute-based scoring in cognitive diagnostic assessment. *Journal of Educational Measurement*, *46*, 293–313.
Gierl, M. J., & Haladyna, T. (2013). *Automatic item generation: Theory and practice*. New York, NY: Routledge.
Gierl, M. J., & Lai, H. (2012a). Using item models for automatic item generation. *International Journal of Testing*, *12*, 273–298.
Gierl, M. J., & Lai, H. (2012b, April). *Using automatic item generation to create items for medical licensure exams*. Paper presented at the meeting of the National Council on Measurement in Education, Vancouver, BC.
Gierl, M. J., & Lai, H. (2013). Using weak and strong theory to create item models for automatic item generation: Some practical guidelines with examples. In M. J. Gierl & T. Haladyna (Eds.), *Automatic item generation: Theory and practice* (pp. 26–39). New York, NY: Routledge.
Gierl, M. J., Lai, H., & Turner, S. (2012). Using automatic item generation to create multiple-choice items for assessments in medical education. *Medical Education*, *46*, 757–765.
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), 1–51. Retrieved from http://www.jtla.org
Glas, C. A. W., & van der Linden, W. J. (2003). Computerized adaptive testing with item cloning. *Applied Psychological Measurement*, *27*, 247–261.
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. *The Electronic Journal of e-Learning*, *9*, 23–38.
Haladyna, T. (2013). Automatic item generation: A historical perspective. In M. J. Gierl & T. Haladyna (Eds.), *Automatic item generation: Theory and practice* (pp. 13–25). New York, NY: Routledge.
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.
Hornke, L. F. (2002). Item generation models for higher order cognitive functions. In S. H. Irvine & P. C. Kyllonen (Eds.), *Item generation for test development* (pp. 159–178). Mahwah, NJ: Lawrence Erlbaum.
Huff, K., Alves, C., Pellegrino, J. & Kaliski, P. (2013). Using evidence-centered design task models in automatic item generation. In M. J. Gierl & T. Haladyna (Eds.), *Automatic item generation: Theory and practice* (pp. 102–118). New York, NY: Routledge.
Irvine, S. H., & Kyllonen, P. C. (2002). *Item generation for test development*. Hillsdale, NJ: Lawrence 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, H., & Gierl, M. J. (2013). Generating items under the assessment engineering framework. In M. J. Gierl & T. Haladyna (Eds.), *Automatic item generation: Theory and practice* (pp. 77–101). New York, NY: Routledge.
Lai, H., Gierl. M. J., & Alves, C. (2010, April). *Using item templates and automated item generation principles for assessment engineering*. Paper presented at the meeting of the National Council on Measurement in Education, Denver, CO.
Leighton, J. P., & Gierl, M. J. (2007). *Cognitive diagnostic assessment for education: Theory and applications*. Cambridge, UK: Cambridge University Press.
Leighton, J. P., & Gierl, M. J. (2011). *The learning sciences in educational assessment: The role of cognitive models*. Cambridge, UK: Cambridge University Press.
Luecht, R. (2013). An introduction to assessment engineering for automatic item generation. In M. J. Gierl & T. Haladyna (Eds.),

Automatic item generation: Theory and practice (pp. 59–76). New York, NY: Routledge.
Minsky, M. (1974). *A framework for representing knowledge*. MIT-AI Laboratory Memo 306. Cambridge, MA: Massachusetts Institute of Technology.
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: Lawrence Erlbaum.
Mortimer, T., Stroulia, E., & Yazdchi, Y. (2013). IGOR: A web-based item generation tool. In M. J. Gierl & T. Haladyna (Eds.), *Automatic item generation: Theory and practice* (pp. 217–230). New York, NY: Routledge.
Norman, G., Eva, K., Brooks, L., & Hamstra, S. (2006). Expertise in medicine and surgery. In K. A. Ericsson, N. Charness, P. J. Feltovich, & R. R. Hoffman (Eds.), *The Cambridge handbook of expertise and expert performance* (pp. 339–353). Cambridge, UK: Cambridge University Press.
Reiter, E. (1995). NLG vs. templates. *Proceedings of the Fifth European Workshop on Natural Language Generation* (pp. 95–105). Leiden, The Netherlands. Available at http://arxiv.org/pdf/cmp-lg/9504013.pdf
Rudner, L. (2010). Implementing the Graduate Management Admission Test Computerized Adaptive Test. In W. van der Linden & C. Glas (Eds.), *Elements of adaptive testing* (pp. 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: Lawrence Erlbaum.
Sinharay, S., & Johnson, M. S. (2008). Use of item models in a large-scale admissions test: A case study. *International Journal of Testing*, *8*, 209–236.
Sinharay, S., & Johnson, M. S. (2013). Statistical modeling of automatically generated items. In M. J. Gierl & T. Haladyna (Eds.), *Automatic item generation: Theory and practice* (pp. 183–195). New York, NY: Routledge.
Sinharay, S., Johnson, M. S., & Williamson, D. M. (2003). Calibrating item families and summarizing the results using family expected response functions. *Journal of Educational and Behavioral Statistics*, *28*, 295–313.
Wendt, A., Kao, S., Gorham, J., & Woo, A. (2009). Developing item variants: An empirical study. In D. J. Weiss (Ed.), *Proceedings of the 2009 GMAC Conference on Computerized Adaptive Testing*, www.psych.umn.edu/psylabs/CATCentral/ (accessed April 1, 2011).

Instructional Topics in Educational Measurement (ITEMS)

Using Automated Processes to Generate Test Items

Self-Test (with Key)

Mark J. Gierl
Hollis Lai
Centre for Research in Applied Measurement and Evaluation
University of Alberta

1. What are the three general steps required for template-based automatic item generation?

KEY: Automatic item generation is the process of using item models to generate test items with the aid of computer technology. Template-based AIG typically requires three general steps. First, test development specialists create item models that specify the elements in the assessment task that must be manipulated. Second, the content used for item generation is identified and structured by test development specialists. This content can be structured using either a weak or strong theory approach. Third, elements in the item model are manipulated with computer-based algorithms to produce new items.

2. What is an item model? What are the three components in this model?

KEY: An item model is like a template, mould, or prototype that highlights the features in an item that must be manipulated to produce new items. Item models contain the components in an assessment task that can be used for item generation. The first component is the stem. The stem contains context, content, item, and/or the question the examinee is required to answer. The second component is the options. The options includes a set of alternative answers with one correct option and one or more incorrect options or distracters. Both stem and options are required for multiple-choice item models. Only the stem is created for constructed-response item models. The third component is the auxiliary information. The auxiliary information includes any additional content, in either the stem or option, required to generate an item, including text, images, tables, graphs, diagrams, audio, and/or video. The stem and options can be further divided into elements. An element is the specific variable in an item model that is manipulated to produce new test items.

Items 3 and 4 require Figure 2 and 4 from the module.

3. For the 1-layer surgery item model, how many unique combinations can be assembled?

KEY: 8 AGE * 2 GENDER* 4 PAIN * 4 LOCATION * 4 ACUITY-OFONSET *4 PHYSICALFINDINGS * 4WBC = 16384

For the n-layer surgery item model, how many unique combinations can be assembled?

KEY: 16384 from 1st Layer * 3 QuestionPrompt * 4 TestFindings *4 Situation = 786432

4. What would be an expression of a constraint if items should only be presented with males over 40?

KEY: ([AGE]>40 && [GENDER] == 1)

What would be an expression of a constraint if items should only be presented with males over 40? And women under 45?
KEY: ([AGE]>40 && [GENDER] == 1) || ([AGE]<45 && [GENDER] == 2)

5. What is the CSI for the following pair of item stems (without removing common words)?

A: a 24-year-old man presented with a mass in the left groin. It appeared suddenly 2 hours ago while lifting a piano. on examination he has a tender firm mass in the left groin. Which one of the following is the next best step?
B: A 50-year-old man presented with a mass and mild pain in the left groin. It occurred a few days ago after moving a piano. Upon further examination, the patient has normal results and

the mass is tender and reducible. Which one of the following is the best treatment?
KEY:

$$CSI = \frac{\sum A * B}{\sqrt{\sum A^2} * \sqrt{\sum B^2}}$$

$$CSI = \frac{62}{\sqrt{75} * \sqrt{89}}$$

$$CSI = 0.75$$

Word co-occurrence matrix:

word	A	B
2	1	0
24-year-old	1	0
50-year-old	0	1
a	4	4
after	0	1
ago	1	1
and	0	3
appeared	1	0
best	1	1
days	0	1
examination	1	1
few	0	1
firm	1	0
following	1	1
further	0	1
groin	2	1
has	1	1
he	1	0
hours	1	0
in	2	1
is	1	2
it	1	1
left	2	1
lifting	1	0
man	1	1
mass	2	2
mild	0	1
moving	0	1
next	1	0
normal	0	1
occurred	0	1
of	1	1
on	1	0
one	1	1
pain	0	1
patient	0	1
piano	1	1
presented	1	1
reducible	0	1
results	0	1
step	1	0
suddenly	1	0
tender	1	1
the	4	5
treatment	0	0
upon	0	1
which	1	1
while	1	0
with	1	1

6. One possible criticism of automatic item generation is that it undermines the role of the content expert by placing "the machine" in charge of creating test items. Would you agree with this criticism?

KEY: We have been involved in many discussions and we have been asked to address this point in numerous content specialists meetings. We adamantly do not agree that "the machine" is assuming control over item development, hence our answer is no, we do not agree with this criticism. Automatic item generation merely represents a shift in the responsibilities of content experts. The role of the content expert is critical for the creative task of designing and developing meaningful item models as well as identifying the content required for these models. The role of computer technology is critical for the algorithmic task of systematically combining large amounts of content in each model to produce new items. By combining content expertise and computer technology, testing specialists can create models that yield large numbers of high-quality items in a short period of time. We can also represent the roles of content experts, computer technology, and automatic item generation with this diagram: