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 spot-on. 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 test items. 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 in the domain of certification and licensure testing 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). 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 U.S. 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 consortia, PARCC and SMARTER Balance, are using technology to develop and deliver computerized tests and to design constructed-response items and performance-based tasks that will be scored using computer algorithms.

Computerized testing offers many advantages to examinees and examiners compared to more traditional paper-based tests. For instance, computers support the development of technology-enhanced item types that allow 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 and scoring sessions. Because items are scored immediately, examinees receive instant feedback on their strengths and weaknesses. Computerized tests also permit continuous and on-demand administration, thereby allowing examinees to have more choice about where and when they write their exams.

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 computerized adaptive test, which is a commonly used administrative format for certification and licensure testing, with two administrations per year would require, at minimum, a bank with 2,000 items. The costs associated with developing 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. 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 computerized adaptive test.

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 specialist approach to ensure more items are available. But we know this option is costly. 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 (Embretson & Yang, 2007; Gierl & Haladyna, 2013; Irvine & Kylloenen, 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 (1969) more than four decades ago, is gaining renewed interest because it addresses one of the most pressing and challenging issues facing educators today—the rapid and efficient production of high-quality, content-specific test items. This production is needed, in part, to support the current transition to computerized testing.

AIG has at least four important benefits for test developers. First, AIG permits the test developer to create a single item model that, in turn, yields many test items. An item model is a template that highlights the features in an assessment task that can be manipulated to produce new items. Multiple models can be developed that will yield hundreds or possibly thousands of new test items. These items are then used to populate item banks. Computerized tests draw on a sample of the items from the bank to create new tests.

Second, AIG can lead to more cost-effective development because the item model is continually reused 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 across large numbers of items (Schmeiser & Welch, 2006). 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 conventional view of a single item where every element is unique, both within and across items. Drasgow, Luecht and Bennett (2006, p. 473) provide this description of the traditional content specialist 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 fundamental 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 reused, particularly when only a small number of the generated items are used on a specific test form, which, again, could yield economic benefits.

Fourth, AIG may enhance test security. Security benefits could be realized when large numbers of items are available, simply by decreasing the per-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. 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 difficult for the examinees to memorize and reproduce items.

**Purpose of Chapter**

Haladyna (2013) presented a comprehensive overview of AIG history, beginning in the 1950s with the development of Louis Guttman’s facet theory. But the last decade was characterized by a flurry of AIG research. This research has focused, in part, on design-related issues, such as cognitive model development (e.g., Embretson & Yang, 2007; Gierl & Lai, 2013b; Gierl, Lai & Turner, 2012), item model development (e.g., Bejar & Cooper, 2013; Gierl & Lai, 2012b; Gierl, Zhou & Alves, 2008) and test designs for AIG (e.g., Bejar et al., 2003; Embretson & Yang, 2007; Huff, Alves, Pellegrino & Kaliski, 2013; Lai & Gierl, 2013; Luecht, 2013). Research has focused on technological advances for AIG (e.g., Gierl et al., 2008; Gütl, Lankmayr, Weinhofer & Höfler, 2011; Higgins, 2007; Higgins, Futagi & Deane, 2005; Mortimer, Strouila & 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., Aldabe & Maritxalar, 2010; Gütl et al., 2011; Karamanis, Ha & Mitkov, 2006; Mitkov, Ha & Karamanis, 2006; Moser, Gütl & Lui, 2012), frame-semantic representations (e.g., Cubric & Tosic, 2010; Deane & Sheehan, 2003; Higgins et al., 2005), schema theory (e.g., Singley & Bennett, 2002) and semantic web-rule language (Zoumpatianos, Papalouros & Kotis, 2011). AIG research has also focused on estimating the psychometric characteristic of the generated items (e.g., Cho, DeBoeck, Embretson & Rabe-Hesketh, in press; Embretson, 1999; Geerling, Glas & van der Linden, 2011; Glas & van der Linden, 2003; Sinharay & Johnson, 2008, 2013; Sinharay, Johnson & Williams, 2003). Because of these important 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, 2012b, 2013b) 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; Cho, DeBoeck, Embretson & Rabe-Hesketh, in press; Embretson & Daniels, 2008; Sinharay & Johnson, 2008, 2013) as well as situational judgment (Bejar & Cooper, 2013), word fluency (Arendasy, Sommer & Mayr, 2012), visual short-term memory (Hornke, 2002), vocabulary recall (Brown, Frishhoff & Eskenazi, 2005), cloze tasks (Goto, Kojiri, Watanabe, Iwata & Tamada, 2010), analogies (Alsubait, Parsia & Sattler, 2012) and mental rotation (Arendasy & Sommer, 2010); and in licensure and certification content areas, such as nursing, architecture and medicine (Karamanis et al., 2006; Gierl et al., 2008; Gierl, Lai & Turner, 2012; Wendt, Kao, Gorham & Woo, 2009).

The purpose of this chapter is to describe and illustrate a practical method for generating test items. We will present the basic logic required for generating items using a template-based method that provides the basis for understanding other AIG approaches. By template-based AIG, we mean methods that draw on item models to guide the generative process. An item model is comparable to a mold, rendering or prototype that highlights the features in an assessment task that must be manipulated to produce new items. To ensure our description is both concrete and practical, we illustrate template-based item generation using an example from a medical licensure exam. This example was selected to highlight the applicability and the generalizability of template-based AIG using a complex
Automatic Item Generation

A three-step process is presented. In step 1, the content required for item generation is identified by domain specialists. In step 2, an item model is developed to specify where this content is placed in each generated item. In step 3, computer-based algorithms are used to place the content specified in step 1 into the item model developed in step 2. Using this three-step method, large numbers of items can be generated using a single item model.

While AIG provides a method for producing new items, the psychometric properties (e.g., item difficulty) of these newly generated items must also be evaluated. Item quality is often determined through a field-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 may not be feasible or desirable 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. A description of precalibration statistical methods is beyond the scope of this chapter. However, a recent review of these methods is presented in Sinharay and Johnson (2013).

AIG Three-Step Method

Step 1: Cognitive Model Development

Overview

To begin, test developers must identify the content that will be used to produce new items. This content is identified using design principles and guidelines that highlight the knowledge, skills and abilities required to solve problems in a specific domain. This content must also be organized and structured in a manner that can promote item generation. A strong body of literature exists on how medical knowledge is conceptualized, organized and structured. Norman, Eva, Brooks and Hamstra (2006), for instance, characterized the organization of medical knowledge as causal (normal human functioning and disease processes), analytic (relationship of specific symptoms and features with specific conditions) and experiential (prior case experiences). Leighton and Gierl (2011) provided a detailed account for how knowledge is organized and structured to account for mathematical reasoning, reading comprehension and scientific reasoning.

Just as frameworks are needed to study the structure and application of knowledge in medicine, mathematics, reading and science, frameworks are also needed to generate test items. Figure 21.1 contains a framework that specifies the knowledge required to make a therapeutic (i.e., drug intervention) decision to address infection during pregnancy. Gierl, Lai and Turner (2012) called the framework in Figure 21.1 a cognitive model for AIG. A cognitive model for AIG is intended to highlight the knowledge, skills and abilities required to solve a problem in a specific domain. This model also organizes the cognitive- and content-specific information into a coherent whole, thereby presenting a succinct yet structured representation of how examinees think about and solve problems.

To create the cognitive model in Figure 21.1, two content specialists, who were experienced medical item writers and practicing physicians, described the knowledge, content and clinical-reasoning skills required to solve different problems using therapeutic interventions. The knowledge and skills for the Figure 21.1 cognitive model were identified in an inductive manner by asking the content specialists to review a parent multiple-choice item (see Figure 21.2) and then to identify and describe key information that would be used by an examinee to solve the item. Three types of key information required to solve the parent item in this example can be described. They include the problem and associated scenarios, sources of information, and features (see Figure 21.3).

These three types of key information are specified as separate panels in Figure 21.3. The top panel identifies the problem and its associated scenarios. The content specialists first began by identifying...
A 24-year-old pregnant female at 24 weeks gestation, presents with clinical and radiological signs and symptoms consistent with a left lower lobe pneumonia. Which one of the following antibiotics is the most appropriate?

1. Levofloxacin.
2. Tetracycline.
3. Clarithromycin.
4. Doxycycline.
5. Azithromycin. *

*correct option.
the problem (i.e., infection and pregnancy) specific to the existing test item. Then they identified different drug types that could be prescribed (i.e., penicillin [P], cephalosporin [C], macrolides [M], sulfa [S], furantoin [F]) to treat infection during pregnancy, along with the associated noncommercial drug names (e.g., penicillin G, amoxicillin, ampicillin).

The middle panel specifies the relevant sources of information required to create variables that can be manipulated in the item model. Sources of information can be case-specific (e.g., type of infection) or generic (e.g., patient characteristics). We selected a relatively simple example for illustrative purposes in this chapter, where only two sources of information were identified from a universe of all possible sources of information. But many different sources of information related to the problem and its associated scenarios (e.g., symptomatic presentation, laboratory results, patient history) could be included in the cognitive model, thereby increasing its generative capacity. That is, the cognitive

Figure 21.3  A general cognitive model structure for AIG.
model is developed, in part, to reflect the knowledge, skills and abilities required to solve the problem. But the model can also be developed with the more pragmatic goal of reaching a generative target given the developer’s item banking requirements.

The bottom panel highlights the salient features, which include the elements and constraints, within each source of information. For Figure 21.1, six features (i.e., urinary tract infection, pneumonia, cellulitis, gestation period, allergy, age) were identified across two sources of information. Each feature also specifies 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. As one example, the cellulitis feature in the bottom left corner of the cognitive model contains the element “present” (i.e., the pregnant patient with infection has cellulitis). The second nested component for a feature is the constraint. Each element is constrained by the scenarios specific to this problem. For instance, cephalosporin (C) and macrolides (M) are the drugs “very likely” to be used to treat infection and pregnancy when the type of infection is cellulitis. A generalized cognitive model structure that can be used to produce items in different content areas and in diverse knowledge domains is presented in Figure 21.3.

The content presented in the cognitive model for AIG serves two purposes. The first purpose is practical. 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 (infection and pregnancy) and the associated drug types and noncommercial drug names (e.g., penicillin G, amoxicillin and ampicillin are drug names for the drug type penicillin) to the features (urinary tract infection, pneumonia, cellulitis, gestation period, allergy, age) through the sources of information (type of infection, patient characteristics). These prescriptive links are used for item generation, as the features can be inserted in their appropriate information sources, as outlined in Figure 21.1, subject to the elements and their constraints to yield new test items.

The second purpose of the cognitive model is more abstract. A cognitive model for AIG highlights the knowledge, skills and abilities required to solve a problem in a specific domain. It also organizes the cognitive- and content-specific information to provide a structured representation of how examinees think about and solve problems. Hence, the cognitive model could be considered a construct representation that guides item development. More than 30 years ago, Embretson (1983) suggested that cognitive theory could enhance psychometric practice by illuminating the construct representation of a test. The construct that underlies test performance is represented by the cognitive processes, strategies, knowledge and content used by an examinee to respond to a set of test items. Once these cognitive requirements are sufficiently described, Embretson also claimed they could be assembled into cognitive models to develop items that elicit specific knowledge structures and cognitive processing skills. Test scores anchored to a cognitive model should be more interpretable and, perhaps, more meaningful to a diverse group of users because performance is described not only using a specific set of cognitive skills in a well-defined content area but also using items developed to directly measure these skills. Norman, Eva, Brooks and Hamstra (2006) provided a similar line of reasoning by stating that problem representation was an important way to organize and study the content and processes required for expert medical reasoning and problem solving. The method described in this chapter provides an operational example of how Embretson’s construct representation and Norman et al.’s problem representation can be used to generate test items. The cognitive model for AIG was created by medical content specialists, thereby serving as a representation of how these experts think about and solve problems related to infection and pregnancy. This representation was documented in the form of an explicit cognitive model and then used to guide the detailed assembly process needed for item generation. The item model rendering and computer-based assembly are described next.
Step 2: Item Model Development

Overview

With the content identified and structured using the cognitive model in step 1, this content must now be positioned within a template that, in turn, will create the assessment tasks. This template is called an item model. Item models (Bejar, 1996, 2002; Bejar et al., 2003; LaDuca, Staples, Templeton & Holzman, 1986) have been described using different terms, including schemas (Singly & Bennett, 2002), blueprints (Embreton, 2002), templates (Mislevy & Riconcente, 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 manipulated 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 options include a set of alternative answers with one correct option and one or more incorrect options or distractors. 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.

Types of Item Models

The principles, standards, guidelines and practices used for traditional item development (e.g., Case & Swanson, 2002; Downing & Haladyna, 2006; Haladyna & Rodriguez, 2013; Rodriguez, this volume; Schmeiser & Welch, 2006) currently provide the foundational concepts necessary for creating item models. A literature on item model development is also beginning to emerge (e.g., Gierl et al., 2008; Gierl & Lai, 2013b), and some illustrative examples are available (e.g., Bejar et al., 2003; Case & Swanson, 2002; Gierl et al., 2008; Gierl & Lai, 2013b). Two types of item models can be created for AIG: 1-layer and n-layer item models (Gierl & Lai, 2012a).

1-LAYER ITEM MODEL

The goal of item generation using the 1-layer item model is to produce new test items by manipulating a relatively small number of elements in the model. We use the item model element as the unit of analysis in our description because it is the most specific variable in the cognitive model that is manipulated to produce new items. The 1-layer item modeling currently dominates 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 for the infection and pregnancy example was presented in Figure 21.2. The parent item highlights the underlying structure of the model, thereby providing a point of reference for creating alternative items. Then, an item model is created from the parent by identifying elements that can be manipulated to produce new items.

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 is 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. These items are often described as isomorphic. In our experience, generated isomorphic items from 1-layer models are referred to pejoratively by many test developers as “clones,” “ghost” items or “Franken-items.” Isomorphic items are often perceived to be simplistic and easy to produce.
One early 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 developers with design guidelines for creating item models that yield diverse types of generated items. Gierl et al.'s 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., 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 elements 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 distractors are selected, presumably, from a list of possible alternatives. The distractors in this case are selected randomly. Constrained options mean that the keyed option and the distractors are generated according to specific constraints, such as algorithms, rules, formulas or calculations. Fixed options occur when both the keyed option and distractors 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 developers do not design item models where the same elements are constantly manipulated or where the same item model structure is frequently used.

Figure 21.4 contains an example of the 1-layer item model based on the Figure 21.3 parent item. For this 1-layer item model, the stem contains two integers (GESTATION PERIOD; AGE) and two strings (TYPE OF INFECTION; ALLERGY). Using the Gierl et al. (2008) taxonomy described earlier, this item model would be described as a mixed stem with constrained options. The GESTATION PERIOD integer and TYPE OF INFECTION and ALLERGY string elements in the stem are dependent because the values they assume will depend on the combination of content in the item model. The AGE integer, however, is free to vary with all combinations of items; hence it is independent of the other elements (hence both independent and dependent elements are included in this example, making it mixed). The options are constrained by the combination of integer and string values specified in the stem, regardless of the AGE element.
**II-LAYER ITEM MODELS**

The second type of item model can be described as multiple- or n-layer (Gierl & Lai, 2012a). 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 (e.g., Higgins, Futagi & Deane, 2005). Language is often structured hierarchically, meaning that content or elements are often embedded within one another. This hierarchical organization 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.

A comparison of the 1-layer and n-layer item model is presented in Figure 21.5.

For this example, the 1-layer model can provide a maximum of four different values for element A. Conversely, the n-layer model can provide up to 64 different values by embedding the same four values for elements C and D 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 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 developers because large numbers of systematic manipulations are occurring in each model, thereby promoting heterogeneity in the generated items. The disadvantage

![Figure 21.5](image-url)
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 using precalibration statistical methods.

An n-layer infection and pregnancy item model is presented in Figure 21.6.

This example illustrates 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

![Figure 21.6](image-url)
example, we now embed the integers and strings within one another to facilitate the generative process. For the n-layer example, two layers are used. The first layer is sentence structure. The first sentence states, “A \([\text{AGE}]\)-year-old pregnant female at \([\text{GESTATION PERIOD}]\) weeks gestation[\text{ALLERGY}] presents with clinical and radiological signs and symptoms consistent with \([\text{TYPE OF INFECTION}]\).” The second sentence states, “Suppose a pregnant woman[\text{ALLERGY}] was admitted with signs consistent with \([\text{TYPE OF INFECTION}]\). She was \([\text{GESTATION PERIOD}]\) weeks into her term.” The second layer includes the same elements specified in the 1-layer model (see Figure 21.4)—namely, Type of Infection, Allergy and Gestation Period. In sum, by introducing layered elements, more diverse items can be generated because the 1-layer model is a subset of the n-layer model.

**Step 3: Generating Items Using Computer Technology**

**Overview**

Once the item models are created and the content for these models has been identified by the test developers, 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 *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 to generate multiple-choice and constructed-response test items. For this chapter, 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 remember 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.

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 cognitive 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 systematically 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. For instance, the use of the drug types cephalosporin (C) and macrolides (M) are constrained by cellulitis because this type of infection is “very likely” to be treated with C and M (i.e., C, M very likely at the bottom left side of the Features panel in Figure 21.5).

**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 21.4 and 21.6 (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 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 distractor. To generate items from a model, the Test Item Generator dialogue box is presented, where the user specifies the item model
Table 21.1 Random-Sample of Four Generated Items Using 1-Layer Infection and Pregnancy Item Model

1. A 24-year-old pregnant female at 8 weeks gestation presents with clinical and radiological signs and symptoms consistent with a urinary tract infection. Which one of the following antibiotics is the most appropriate for this patient?
   1. Sulfa
   2. Penicillin
   3. Furantoin
   4. Isotretinoin
   5. Cephalosporin

799. A 32-year-old pregnant female at 8 weeks gestation presents with clinical and radiological signs and symptoms consistent with cellulitis. Which one of the following antibiotics is the most appropriate for this patient?
   1. Cephalexin
   2. Amoxicillin
   3. Isotretinoin
   4. Nitrofurantoin
   5. Sulfamethoxazole

942. A 24-year-old pregnant female at 12 weeks gestation presents with clinical and radiological signs and symptoms consistent with cellulitis. Which one of the following antibiotics is the most appropriate for this patient?
   1. Amoxicillin
   2. Isotretinoin
   3. Erythromycin
   4. Nitrofurantoin
   5. Sulfamethoxazole

1364. A 32-year-old pregnant female at 12 weeks gestation presents with clinical and radiological signs and symptoms consistent with cellulitis. Which one of the following antibiotics is the most appropriate for this patient?
   1. Amoxicillin
   2. Cephalexin
   3. Isotretinoin
   4. Nitrofurantoin
   5. Sulfamethoxazole
IGOR was used with the n-layer item model presented in Figure 21.6, 1,453 items were generated. A random sample of four items is presented in Table 21.2.

### Table 21.2 Random Sample of Four Generated Items Using N-Layer Infection and Pregnancy Item Model

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2. A 24-year-old pregnant woman at 8 weeks gestation presents with shortness of breath, cough, purulent sputum and a mild fever. Chest X ray shows infiltrates in the left, lower lobes. Which one of the following antibiotics is the most appropriate?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>466. A 27-year-old woman primigravida was admitted with shortness of breath, cough, purulent sputum and a mild fever. Chest X ray shows infiltrates in the left, lower lobes. She was 14 weeks into her term. Which one of the following antibiotics is the most appropriate?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>568. A 32-year-old pregnant woman at 14 weeks gestation presents with clinical and radiological signs and symptoms consistent with a left lower lobe pneumonia. Which one of the following antibiotics is the most appropriate?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1234. A 35-year-old woman primigravida who is allergic to penicillin was admitted with shortness of breath, cough, purulent sputum and a mild fever. Chest X ray shows infiltrates in the left, lower lobes. She was 12 weeks into her term. Which one of the following antibiotics is the most appropriate?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Evaluating Word Similarity of Generated Items

To measure and compare the word similarity of the items created using 1- and n-layer models, the intramodel differences, meaning items generated within the same model, must be evaluated. Because fewer variables are manipulated with the 1-layer approach, word similarity should be higher for items generated with this model compared with the n-layer model. Word similarity can be quantified using...
a natural language processing measure called the cosine similarity index (CSI). The CSI is a measure of word similarity between two vectors of co-occurring texts. It is one kind of word similarity measure. It is calculated using an algorithm based on a text-vector indexing technique (Bayardo, Ma & Srikant, 2007; Becker & Kao, 2009; Spertus, Sahami & Buyukkokten, 2005) where the similarity between two vectors of co-occurring texts is computed using the cosine of the angle between the two vectors in a multidimensional space of unique words. The CSI is given by

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

where $\vec{A}$ and $\vec{B}$ are two binary vectors that represent the word occurrence in strings A and B from the universe of unique words. The CSI ranges from 0 and 1. The minimum CSI value of 0 means that no word overlapped between the two vectors. The maximum CSI value of 1 means that words represented by the two vectors are identical.

To illustrate the use of the CSI in our infection and pregnancy example, a random sample of 100 items from the 1- and n-layer item models was selected and analyzed. Because fewer elements are manipulated with the 1-layer model, word similarity should be higher for these items. The 1-layer item produced a CSI mean of 0.87 and a standard deviation of 0.09, indicating that the generated items are quite similar to one another and relatively homogeneous (see Table 21.1). The n-layer model produced a comparatively lower CSI mean of 0.65 and a higher standard deviation of 0.17. These results reveal that the n-layer item model produces a more heterogeneous and diverse item set compared to the items generated from a 1-layer item model (see Table 21.2).

**Multilingual Item Generation**

The n-layer model is a flexible structure for item generation, thereby permitting many different but feasible combinations of embedded elements. In addition to generating more diverse 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 of 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.

Earlier in this chapter we described a method for using the plausible values specified in a cognitive model to generate new items by systematically replacing the item model content using computer algorithms. These replacement values are specified in the cognitive model as elements. As item models become more complex due to the requirements specified in cognitive models and in the linguistic complexity required for adapting items into different languages, the number of elements used for item generation dramatically increases. The increase in the number elements is problematic because it complicates the programming task and it affects the computation time required to run IGOR. To address this problem, Gierl, Lai, Fung and Zheng (in press) introduced the concept of a linked element as a way to facilitate the IGOR programming task and to increase IGOR’s computational speed.

Recall that the use of layered elements permits content to be embedded within content in an item model (see Figure 21.6). Layered elements, therefore, have a “vertical” function for item content (i.e., content within content). Linked elements also expand the capabilities of item modeling by permitting content to be transformed within an item model. For multilingual AIG, the transformation is from one language to another. Linked elements, therefore, have a “horizontal” function for item content (i.e., content in language 1 is transformed to content in language 2). The linked elements used for
language transformations can function in four different forms: words, key phrases, single sentences and multiple sentences. These four forms are then used to adapt words, phrases and sentences from one language to another to permit multilingual AIG.

In our current example, we generated infection and pregnancy items in English. However, Canada is officially bilingual. Therefore, the Medical Council of Canada, the agency that licenses physicians, must administer items in both English and French. To accommodate item development in this scenario, we demonstrate how items can be generated simultaneously in English and French. The multilingual AIG example was created with the help of a bilingual medical content specialist. Four types of linked elements were identified and used for multilingual AIG in our example. First, linked elements are specified in the form of a word. These elements require the direct translation or adaptation of a single word between languages in the n-layer item model. Second, linked elements are specified in the form of a key phrase. These elements require the direct translation or adaptation of key phrases between languages. Third, linked elements are specified in the form of a single sentence. These elements require the direct translation or adaptation of words and key phrases as well as the coordination of these elements to produce a coherent sentence. Because the literal or direct combination of words and key phrases can produce awkward expressions, some linguistic refinement may be required to produce a more precise sentence. Fourth, linked elements are specified in the form of multiple sentences. A multiple-sentence linked element could be the entire test item. Because words, key phrases and single sentences have been carefully adapted prior to assembling multiple sentences, only small adjustments should be required for this linked element transformation. However, as with the linked elements at the single sentence level, care must be taken to coordinate these elements so a coherent whole is produced.

Taken together, linked elements specify content in four different forms that provide the translation or adaptation necessary to program IGOR so item generation can occur in multiple languages. Our example is constrained to two languages, but three or more languages can be developed using the same linked element logic to permit simultaneous multilingual item generation. Moreover, IGOR is character set–neutral, meaning that characters from any language can be used to generate test items. Once the four-level linked elements are completed, a multilingual AIG linking map is produced. The map summarizes the necessary links for words, key phrases, single sentences and multiple sentences (for more details, see Gierl et al., in press). Then, IGOR is programmed using the item model content in Figure 21.6 as well as the linking map to produce new items. Using this approach, a total of 2,906 items were generated—1,453 English and 1,453 French items.

Summary
Testing agencies need large numbers of high-quality items that are produced in a timely and cost-effective manner. The rapid transition to computerized testing has only served to accentuate this need. AIG helps address some of these development challenges. The template-based AIG approach we described in this chapter requires three steps. First, the content used for item generation is identified and structured using a cognitive model. Second, item models are created to specify where the content from the cognitive model should be positioned within the template-based assessment structure. Third, elements in the item model are manipulated with computer-based algorithms to produce new items. Using this three-step method, hundreds or thousands of new items can be generated using a single item model, as the demonstration for the infection and pregnancy example used throughout this chapter helps to illustrate.

The New Art and Science of Item Development
In their seminal chapter on test development in the fourth edition of the handbook *Educational Measurement*, Cynthia Schmeiser and Catherine Welch begin with this provocative question: Test
development: art or science? Our chapter on AIG could be interpreted as a shift away from the “art” of test development, where assessment tasks are created solely from content expertise, experience and judgment, toward a new “science” of test development, where these tasks are created by combining the knowledge and skills of the content specialists with the algorithmic power of modern computing to produce new items. But it is important to add that, in our view, this new science of test development does not diminish, in any way, the role of content specialists. Rather, it helps focus their responsibilities on the creative task of identifying, organizing and evaluating the content needed to develop test items. That is, the test developer 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. These responsibilities will not be replaced any time soon by computer technology because they require refined judgment, expertise and experience. The role of computer technology in AIG is required for the generative and, frankly, monotonous 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. Hence, in our view, AIG represents a merger between the art and science of item development, where well-defined responsibilities that adhere to specialized skills according to the appropriate division of labor contribute to the production of large numbers of items. It is also worth mentioning that the phrase automatic item generation inherited from researchers and practitioners in our past could also be characterized using a less threatening and, possibly, more descriptive phrase, such as technology-enhanced item development, because items are generated automatically only in step 3 (i.e., using computer algorithms), after the test developer has created the cognitive and item models in steps 1 and 2.

Limitations and Next Steps

We presented a general method for creating test items. But the psychometric properties (e.g., item difficulty) and the quality of the generated items must still be evaluated. Psychometric properties are often determined through a field-testing process, where each item is administered to a sample of examinees so the item statistics can be calculated. The psychometric properties of the generated items presented in this chapter are unknown. Methods designed to precalibrate generated items are in the early stage of development (cf. Sinharay & Johnson, 2013). Item quality is evaluated with judgments from content specialists who scrutinize the items. One recent study conducted by Gierl and Lai (2013a) compared medical items developed with an AIG to items developed with traditional test development procedures, using eight indicators of item quality. Gierl and Lai reported that the quality of the multiple-choice items produced using AIG and traditional processes was comparable in seven of the eight indicators of item quality. Consistently, however, the AIG items were flagged as inferior on the eighth indicator, which was the quality of the incorrect options or distractors. Hence, a methodology for generating plausible distractors must also be developed.

Acknowledgments

We would like to thank Vasily Tanygin and Fahad Latifi for their constructive comments on an earlier version of this chapter. We would also like to thank the Medical Council of Canada for supporting this research. However, the authors are solely responsible for the methods, procedures and interpretations expressed in this chapter and these views do not necessarily reflect those of the Medical Council of Canada.
References


