Assessment Engineering Task Model Maps, Task Models and Templates as a New Way to Develop and Implement Test Specifications

Richard M. Luecht

University of North Carolina at Greensboro

Abstract

Assessment engineering is a new way to design and implement scalable, sustainable and ideally lower-cost solutions to the complexities of designing and developing tests. It represents a merger of sorts between cognitive task modeling and engineering design principles—a merger that requires some new thinking about the nature of score scales, item difficulty, and content. This article summarizes some recent developments in developing AE task model maps, task models, and templates as alternative to more traditional test specifications, and discusses some of the necessary quality control mechanisms that can ensure the consistent production of high quality items and test forms over time.

Key Words: Assessment engineering, Item templates, Evidence centered design

Assessment engineering (AE) is a somewhat novel approach to large-scale testing where engineering-based principles are used to direct the design and development as well as the analysis, scoring, and reporting of assessment results (Luecht, 2006, 2007, 2008a-c, 2009, 2010, 2011; Luecht, Burke & Devore, 2009; Luecht, Dallas & Steed, 2010; Luecht & Masters, 2010; Masters & Luecht, 2010; Shu, Burke & Luecht, 2010). In many ways, AE is a highly structured and formalized manufacturing-engineering process for designing and implementing cognitively based assessments envisioned under an evidence-centered design (ECD) framework for summative or formative purposes (Mislevy, 1994, 2006; Mislevy, Steinberg, & Almond, 2003).

Under the AE framework, an assessment begins with a process of laying out ordered proficiency claims along the intended scale and then developing cognitive task models that provide evidence about the proficiency claims. Next, task templates are created from the cognitive model to produce a framework for consistently replicating the assessment tasks (items). Finally, a hierarchical calibration system is used for quality control of the task models and templates, and to maintain the statistical and interpretive integrity of the score scale.

For AE and ECD to succeed, traditional and somewhat outmoded ways of thinking about content and test specifications may need to be abandoned. This paper highlights recent developments in developing AE task model maps, task models, and templates to present an alternative way to conceptualize and implement item and test design specifications. The paper includes a discussion of quality control mechanisms that help ensure the consistent production of items and test forms.

Rethinking Content Blueprints and Assessment Design

Content validity arguably remains one of the most popular and pervasive approaches to test validation in K-12 educational assessment, in professional certification and licensure testing, and in higher education admissions and placement testing. In its most basic form, content validity is implied/claimed if the test content matches the intended *content blueprint* representing the domain of interest (e.g., Anastasi, 1986).

A traditional content blueprint usually consists of a list of topics or standards that subject matter experts or SMEs (e.g., licensed professionals, preceptors, teachers/educators) decide adequately represents a particular domain to be assessed. Each topic or standard creates a specification for how many items to include on each test form. These specifications can be enumerated in terms of exact item counts (e.g., exactly 5 items requiring students to *use the quadratic formula or factoring techniques or both to determine whether the graph of a quadratic function will intersect the x-axis in zero, one, or two points*), acceptable ranges of item counts (e.g., 3 to 6 items), or percentages of the total test length item content requirements (e.g., 6% to 12%).

The depth and breadth of a content blueprint may can vary across domains—possibly listing only a few high-level content categories or covering three or more sublevels of a lengthy content and skills outline. Given the practical test length restrictions imposed by costs, logistics, and policies, most content blueprints represent a compromise of priorities within the domain of interest. SMEs and boards charged with assessment oversight often impose those priorities. However, the priorities may also be informed by practice analyses or surveys that quantify the relative amounts of time spent on and importance of individual topics or standards.

The relative ease of assembling together groups of SMEs to develop a blueprint as a means of validating a test is probably one of the key reasons why content validity and SME-based test blueprinting persist; however, not without serious criticism. For example, Messick (1989) argued that content validity only establishes the relevance and representativeness of the target distribution of item content relative to the larger domain. Content validity does not provide any direct evidence that aids in the interpretation of scores or inferences drawn from observable performances on a particular form of the test. Rather, the basic inference of content blueprinting is that the more complete behavioral or learning domain of interest can be represented by a smaller sample of *observed* performances, if those performances are fairly and reliably evaluated (i.e., responsibly scored) and the sample of items selected is sufficiently large to control for sampling errors appropriate to the intended level of inference about performance in the larger domain (Guion, 1977; Kane, 2006).

What is wrong with content blueprinting as a means of describing test specifications? It certainly has a long history of apparent use. However, there are several nontrivial problems associated with content blueprints. First, most content coding schemes are inherently fallible in that these schemes lack a firm system of rules or concrete indicators for consistently writing or coding items to the content categories. Two groups of SMEs—even SMEs with extensive experience in the test development process—will often disagree on the content representation or specific coding of items. Although compromises are almost always reached in practice, the degree of dissention is seldom documented.

Second, a content blueprint is usually developed as an independent set of test specifications. The statistical specifications (e.g., targeted mean difficulty, a prescribed level of reliability, or an IRT test information target) are not considered alongside the content blueprint;

rather, we often attempt to build test forms to be statistically parallel, subject to meeting the content blueprint. Automated test assembly (ATA) may be used to reconcile these possibly competing demands between content requirements and desired statistical properties of the scores (e.g., van der Linden, 2005). If ATA proves infeasible, we must choose between whether the content specifications or the statistical targets take precedence. In short, there is often a counter-productive (and unnecessary) tension between content and statistical test specifications. Neither set of specifications is independently sufficient to build high-quality test forms, but compromising between them may not produce the most useful test scores either.

Third—and somewhat related to the second limitation—most item-level content specifications ignore the item difficulty and task complexity attributes that contribute to that difficulty. For example, if we choose one easy item and one difficult item from the same content category, could we seriously argue that those items are measuring the *same* complexity of content in terms of knowledge, skills, resource utilization, and context? Although so-called *depth-of-knowledge* indicators (Webb, 2005) or Bloom's taxonomy (Bloom & Krathwohl, 1956) might be layered onto the outline as a tip-of-the-hat toward cognitive complexity, even those types of cognitive specifications are usually interpreted and used in only the vaguest sense.

Assessment engineering (AE) makes three fundamental assertions that may challenge test developers to move past content blueprinting and traditional modes of test development. The first assertion is that ***content is NOT and SHOULD not be considered to be the same across a score scale***. Content is and must be more complex as we progress along a scale. As depicted in Figure 1, incremental complexity can arise from requiring more complex skills, working with more complex knowledge objects, or both. Traditional item content blueprints and coding schemes tend to ignore task complexity and instead rely on the statistical item difficulty to act as a surrogate indicator of increasing or decreasing complexity. The problem with that traditional approach is that we never know how complex a particular item is until we administer it to a large number of examinees and estimate its difficulty. That is, we basically ignore any intentional complexity designed into each item and instead allow a relatively simply psychometric model to sort out the ordering of items. We have no way to compare the cognitive complexity across items or determine if that level of complexity is what was *intended* to challenge the examinees (Mislevy, 2006).

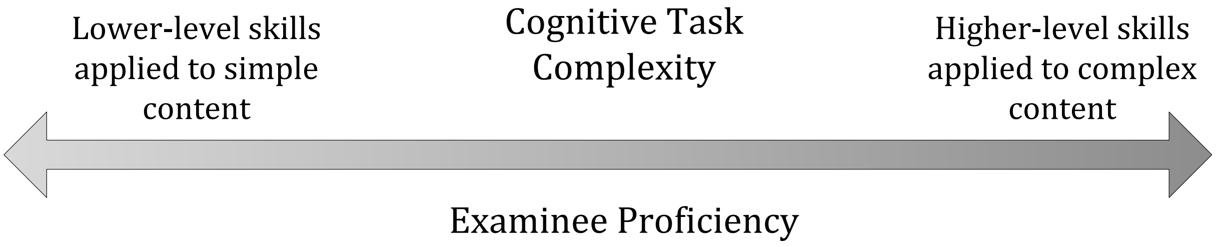

Lower-level skills applied to simple content
Cognitive Task Complexity
Higher-level skills applied to complex content
Examinee Proficiency

Figure 1. Incremental changes in task complexity along a scale

A second fundamental AE assertion is that a *family* of items can be designed from a well-developed, empirically verified model of cognitive task complexity that incorporates the essential elements of most content-based standards—but with a more precise specification of the constellation of skills (i.e., procedural knowledge) needed to perform the tasks in that family, as well as specifications of the declarative knowledge components, available auxiliary aids or tools, and overall contextual complexity of the task setting presented to the examinee as part of the task challenges stemming from that family. A third assertion is that, through the careful engineering of item or task templates and other empirically based quality control mechanisms, large numbers of items can be generated manually or in an automatic fashion—all of which share exactly the same cognitive task complexity specifications and perform as isomorphs from a statistical or psychometric perspective. This is perhaps the most important assertion in that it implies the possibility of replacing a dual or three-part set of item and test specifications (i.e., content, cognitive level, and statistics) with a single specification where the location of each task model along a scale defines the intended cognitive complexity of the task challenges and simultaneously accounts for the item location/difficulty in a statistical sense.

Designing Assessment Engineering Task Models and Task Model Maps

If a goal of AE is to replace traditional content blueprints with something else, a fair question is, what is that "something else"? How can we get there? Unfortunately, it is not a simple process. However, it can be done. The payoffs are: (a) the generation of one or more well-designed scales that do not require tens of thousands of examinee responses and data-hungry psychometric models to maintain, and (b) an extensive supply of low-cost items. The "something else" is accomplished by four core AE processes: (1) construct mapping and evidence modeling; (2) building task models and task model maps; (3) designing templates and item writing; and (4) calibrating the assessment tasks and quality control. Given the focus of this paper on task modeling and template development, only a brief overview is provided below about the first and fourth processes.

Construct Mapping and Evidence Modeling

Construct mapping is a fundamental design process that lays out the *ordered* proficiency-related claims that we wish to make about examinees' knowledge and skills at incrementally higher levels of one or more construct-based scales (Wilson, 2005). Analogous to building a house, a construct map is a well-articulated *vision* for what the eventual scale(s) can tell us about examinees' proficiencies as they progress from the lowest to the higher levels of the scale. Ideally, this process is carried out *before* we begin writing items and attempting to scale the responses using a psychometric model of choice. We can and should plan to iteratively adjust and reposition the proficiency claims along the scale in the same way that an architect might make design adjustments—in this case modifying, but continuing to clearly articulate and document the vision of what we want to say about the examinees at any given point of the scale. It is somewhat akin to developing a very detailed set of achievement- or performance-level descriptors (e.g., Hambleton & Pitoniak, 2006).

Linking this discussion to the previous section, it seems interesting to note that most content blueprints have little to no tangible association with a *scale*. Granted, the eventual statistical scale that emerges from psychometric processing of the response data is assumed to measure the domain sampled via the content blueprint—if the test assembly process matches that blueprint. However, the nature of that inference is highly speculative (Messick, 1989). This point also highlights the fundamental problem with content blueprints. Traditionally, what constitutes successful mastery of a content domain is often impossible to know *until*: (a) a scale has been created demonstrating adequate psychometric quality; (b) test forms have been generated matching the blueprint and intended psychometric properties of the target scale; and (c) standards setting or other methods are employed to add interpretive meaning to the devised scale. In contrast, a construct map is *designed* to be an ordered (i.e., continuous or discrete) description of the progression of knowledge and skill performance claims along an *intended* score scale. The notion of an ordered *scale* is included from Day 1 of the assessment design process, as it should be (Bejar, Braun, & Tannebaum, 2007).

A construct map should also be supplemented with evidence models that describe the expected performance constituting “success” at that point along the scale. Evidence models ultimately become the link between the proficiency claims along the scale and the eventual task models that will be developed to provide evidence about examinee performance—the nature of appropriate task challenges. Evidence modeling includes gathering and documenting exemplars of real work products or other tangible performance-based evidence that constituents and subject-matter experts would agree justifies that a particular examinee has successfully exhibited the cognitive skills and knowledge implicitly or explicitly required by each proficiency claim (Mislevy, 2006; Mislevy, Steinberg, & Almond, 2003; Mislevy & Risconscente, 2006).

It may be important to reiterate the need keep the goal up front—the intended scale(s) and valid interpretations of that/those scale(s). Creating task models and templates, and writing test items and performance tasks without a clear and constant eye on the intended construct(s), is potentially wasteful in terms of the cost and investment in resources to develop and validate the task models and templates that, ultimately, may not provide very useful information. It is addressing writing a love letter, "To whom it may concern." Maintaining the vision of the desired proficiency interpretations along a scale from the onset would seem to be a good way to keep the eventual test development efforts on track and forcing the psychometric modeling into more of confirmatory mode that essentially confirms the intended design, rather than capitalizing on the strongest covariance patterns in a data set.

Task Modeling

A *task model* is a cognitively oriented specification for an entire *family* of items. Each task model integrates declarative knowledge components, relationships among those components, and cognitive skills, as well as relevant content, contexts, and auxiliary features that affect the cognitive complexity of the task. Declarative knowledge components can range from simple identities (e.g., the name or definition of something—that is, simple static representations of discrete knowledge in memory) to a system of declarative knowledge that comprises many declarative knowledge components, relevant relationships among those components, extensive component properties, and possible relationships among the component properties. Cognitively speaking, dealing with a system of declarative knowledge should logically be more taxing than working mentally with an isolated fact or singular declarative concept. Cognitive skills (i.e., procedural knowledge) are applied by the examinee to the declarative knowledge components. Lower-level skills may be combined and enhanced to form higher-level skills, or new skills may emerge and improve as we move up the scale.

A task model therefore includes the *joint* specification of declarative knowledge components, relevant properties of those components, relationships among the components, and cognitive skills required to successfully complete a random task challenge drawn from a task model family. Each task model specification locates that family of tasks relative to other task models. More complex tasks are located higher on the scale and require more cognitive skills, possibly handling incrementally more complex declarative knowledge components, perhaps working without the benefit of auxiliary aids or tools, and/or working in an increasingly more complex or distractive environment. In short, a task model presents a well-articulated cognitive challenge to the examinees.

At a very basic level of comparison to current blueprinting practices, a task model is akin to integrating Bloom's taxonomy (Bloom & Karthwohl, 1956) directly into each content-based standard and simultaneously indicating the range of content-based challenges an examinee would be expected to handle at a certain point along the scale.

The idea of locating a content-based task model along a scale may seem a bit abstract to test developers and item designers who are more accustomed to working with content-based test specifications. It may likewise seem strange to psychometricians who are used to referring exclusively to "*p*-values" or IRT-based item difficulties as the primary scale location parameters of interest. Here, the implication is that each task model can be *located* on the intended scale by creating a detailed design specification that associates the task model's apparent cognitive complexity to the proficiency claims and associated evidence models. The notion of statistical item difficulty remains very real for AE in that each task model is further expected to maintain its difficulty (i.e., statistical location on the scale) as well as other psychometric characteristics (e.g., item discrimination and residual covariance patterns). If that can be accomplished, the entire family of items generated from a particular task model can be treated as statistical isomorphs. In fact, each task model forms the basis for creating multiple assessment task templates and eventually populating an item pool with large numbers of items that operate in a predictable manner—the item family. That is, each task model can be represented by multiple templates, and each template can generate multiple items. This provides enormous efficiencies to treat the task model as a family of templates and each template as a family of items. The association between task models, templates, and items makes it entirely feasible to implement a hierarchical IRT calibration system and related, powerful quality control mechanisms (Geerling et al., 2011; Glas & van der Linden, 2003; Shu et al., 2010).

Assessment engineering task modeling is a different and far more detailed way to design test specifications. The premise is that traditional dual or three-part content and statistical (and possibly cognitive levels) specifications can replaced with a system of elaborate cognitive specifications that detail every assessment task in a way that also takes into account depth-of-knowledge, required cognitive skills, and subject-specific content and contexts to describe a family of items that present similar challenges to the examinee and that behave in a statistically equivalent manner. The location of each task models on the scale, as well as item discrimination, must eventually be empirically validated—ideally using an appropriate, hierarchical IRT calibration model (Geerlings et al., 2011; Shu et al., 2010). If done properly, new items developed for well-behaved task model families may not need to be pilot tested, building the potential for tens, hundreds or even thousands of items to be produced from each task model. As test developers become more familiar with the AE design process, new task models and associated templates can be more readily developed at lower costs in the future to address changing domains or alterations in content over time. In educational settings, this could potentially help realize the goal of providing teachers with an enormous supply of items that measure instructionally sensitive constructs and that provide immediate, on-demand feedback about individual student progress for formative and instructional planning.

Task Model Grammars

A task model can be represented in many ways. One approach is to develop a task-model grammar (TMG). The TMG expressions provide an explicit description of: (a) the combination(s) skills needed to solve the task; (b) the types of declarative knowledge components that are typically used to challenge the examinee in that region of the scale; (c) the information density and complexity of the task components (e.g., one simple component vs. a system of components with complex relations); (d) auxiliary information, resources or tools that facilitate or complicate the task; and (e) other relevant properties or attributes associated with each component and/or set of relations that might affect item difficulty (i.e., specific values and properties of the knowledge objects, relations, or cognitive skills).

An example of a very simple task model with one cognitive skill and only one knowledge object might be represented by a generic TMG statement, $f(x_1)$. A somewhat more complex task model involving the same skill could be written as $f(x_1, x_2)$, where x_1 and x_2 denote two distinct knowledge objects to be manipulated. Finally, $f(g[x_1, x_2], x_3)$ could represent an even more complex (difficult) set of tasks where x_1, x_2, and x_3 are knowledge-objects, $f()$ is (again) the required skill, and $g[\cdot]$ represents a set of relations between x_1 and x_2). We would therefore hypothesize a difficulty ordering of these task models as: $f(x_1) < f(x_1, x_2) < f(g[x_1, x_2], x_3)$. In practice, the functional statements are replaced with action verbs or specific skills that the examinees need to apply to complete items associated with each task model.

It is important to remember that each task model has an intended location on the proficiency scale. Operationally, that location is maintained via the template and quality controls that apply to the family of items produced from each task model. Conceptually, the task model explains the "challenge" presented to the examinees. Its location provides an interpretation of the expected response to the intended level of challenge offered by the task model. The analogy of a diving competition may help illustrate this notion of intentionally altering a challenge. Asking a diver to jump off the end of a diving board placed just above the water level in a pool is certainly easier than asking the same person to jump off a board located three meters above the pool. The changing height of the dive is conceptually similar to the changing the complexity of declarative task components. Asking the diver to execute a simple forward dive would appear to be easier than requiring the diver to do a forward dive with a 2.5 somersault. This is analogous to changing the required level of procedural skill for the task challenge. Finally, if we allow the three-meter dives to take place from a spring board, the added bounce may provide more aid than requiring the divers to perform from a solid platform. In the same way, we increment task-model complexity by explicitly changing the challenge(s) via the TMG specifications.

A TMG expression might look something like the following:

$$action_2\left[action_1\left(is.related\left(object_1, object_2\right), object_3 \middle| context, aux.tools\right)\right]. \qquad (1)$$

Predicate calculus expressions like these have been successfully used in cognitive research to denote semantic complexity (Kintsch, 1988). The "actions" can vary in their *ordered* complexity (e.g., *identify* is a lower complexity action than *analyze*) or functionally nested such as $action_1($ $action_2((\ldots action_m)))$ to denote complexity. The “objects” can differ in number and in complexity by assigning more or less relevant properties to them. Adding more objects or objects with more relevant properties implies greater complexity by potentially increasing the cognitive load of the task challenges. Adding more relevant “relations” can also increase cognitive load, as can adding specific properties to the relations (e.g., directionality, hierarchical associations). The “context” and “auxiliary tools” expressions are also included as the final elements in the predicate clause; both have properties and controls that can alter the complexity of the task in predictable ways. This type of predicate calculus representation of task complexity can also be conveniently represented using modern, object-oriented database designs and structures. The importance of connecting the complexity representation of cognitive skills and knowledge to specific, manipulable, data-based features should be apparent from the perspective of our goal to locate each task model (and its associated family of templates and items) on a particular proficiency scale.

As suggested above, the cognitive skills statements can be specified using *action verbs*. However, action verbs also need to maintain their relative location compared to other action verbs. One way to do this is to construct a list of plausible action verbs, define each verb, and select those verbs that are as unambiguously meaningful as possible. For example, *comprehends* is not necessarily a useful action verb because it implies a rather abstract set of cognitive operations that are difficult to link to observable behaviors in any concrete way. We might instead prefer the use of more outcome-focused action verbs such as *analyze* and *compute*, provided that we explicitly define the types of skills associated with each verb. The definitions are absolutely essential for every verb used to ensure that each verb always represents the same primitive or constellation of skills across tasks associated with a particular construct. In addition, it is important to develop the action verbs in the context of a specific domain. Unlike Bloom's and other cognitive taxonomies, AE neither provides nor makes any assumptions that the same cognitive (procedural) skill descriptors developed for one domain such as high school algebra will generalize to another domain like reading comprehension. It is quite possible that action verbs for different domains will take on different meanings (and possibly, different relative sequencing) for different constructs.

There are at least two ways to define the procedural skills for a TMG by using: (a) primitive actions or skills, or (b) skill constructions or complex skills. *Primitives* are considered verbs (actions) that have no need for an explicit reduced form. That is, primitive verbs have a fixed interpretation of required skill location on the scale, consistent with the evidence models and construct maps claims. Once defined, a primitive verb always maintains its relative ordering with respect to other action verbs, at least as used in describing the task models associated with a particular construct[1]. Procedural skills or actions representing skill constructions can combine lower-level primitive clauses to form higher-order clauses. For example, *calculate*[*identify.method*[*data*]] defines a multistep computational procedure leading to a particular result. Skill constructions can also be denoted by qualifiers such as *calculate.simple* or *calculate.complex*, where the meaning insofar as level of skills is carefully documented and consistently used in that domain. A single verb or skill identifier can be applied to complex skill constructions, replacing combinations of primitives; for example, *analyze*[*data*]= *calculate*[*identify.method*[*data*]].

[1] It is possible and even likely that action verbs will take on different meanings (and possibly, different relative sequencing) for different traits. That is fully acceptable, provided that the construct-specific context is retained as part of the definition of the verb.

As shown in Equation 1, the TMG statements must also indicate the declarative knowledge components—that is, the static or generated knowledge to which the cognitive skills are applied when the examinee encounters the actual items constructed from each task model. For example, asking the examinees to identify the main idea in a paragraph composed of simple sentences and relatively common vocabulary should be easier than applying the same main idea identification skills to a passage containing somewhat unfamiliar vocabulary, complex grammatical constructions or syntax, and other passage features that increase the information density of the passage. As noted earlier, declarative knowledge components can range from simple concepts to an entire system of knowledge components linked together by complex relations among the components.

There are at least three classes of declarative knowledge components that define the "stuff" to which examinees are expected to apply the procedural skills: (a) *knowledge objects* (e.g., static or generated data, graphs, text, or other information); (b) acceptable *relations* among the knowledge objects; and (c) and *auxiliary tools/information* that define, elaborate, or otherwise qualify knowledge components of each task. In general, dealing with more knowledge objects, using more complex knowledge objects, introducing (more) complex relations, and offering fewer auxiliary tools or resources (or potentially punitive resources) will add to the overall information density and challenge the examinee more by increasing cognitive processing loads in working memory. The assumption is that forcing the examinee to contend with more complex knowledge components will tend to increase the difficulty of the family of items for that task model.

Developing a TMG requires a thoughtful and systematic evaluation of the *joint* declarative and procedural cognitive components that make assessment tasks more or less complex—more or less challenging. This is usually a team effort. The evaluation may stem from reverse-engineering existing test items, or, for new tests, cognitive scientists, psychometricians and SMEs may work together to design new task models from the ground up, following the construct map.

As an illustration, Figure 2 shows a collection of TMG elements for an arithmetic reasoning test. This TMG is based on a review of existing test items on a large-scale placement/entrance examination used by the U.S. military. There are five components in the TMG: (a) actions or skills (simple to complex); (b) variables or variable sets used in the problems (simple to complex); (c) properties of the values assigned to variables or constants; (d) design factors for the distractors (assumes a multiple-choice item format); and (e) verbal/information load of the context or problem (simple to complex). It is important to realize that this type of TMG requires continual review and refinement by SMEs and item designers to ensure understanding, consistent use, and the ongoing utility of the elements. New elements can be added as needed or existing elements can be modified/deleted.

Action or Skill Levels	Action or Skill Primitives
Complex	build_equation.complex(solve_for_variable,equation,operation=compound)
Complex	build_equation.implicit(variable_list, operators,target_expression)
Complex	convert.implicit_abstract(x,X,abstract_units)
Complex	convert.implicit_simple(x,X,common_units)
Complex	identify.implicit_complex(common_variables,expression,nature_relations)
Complex	simplify_equation.complex(expression,nature_simplification)
Moderate	build_equation.simple(solve_for_variable,equation,operation=add/subtract)
Moderate	build_equation.simple(solve_for_variable,equation,operation=add/subtract)
Moderate	identify.implicit(common_variable,expression)
Moderate	multiply.fraction_whole(1/n,x)
Moderate	set_inequality(equation,direction)
Moderate	simplify_equation.fraction(expression$_1$,expression$_2$)
Moderate	sum.variable_set($\underline{x}$,n)
Simple	accumulate(x,y,z)
Simple	add(x,y)
Simple	convert.explicit(x,X)
Simple	divide(x,y)
Simple	identify.explicit(x,X)
Simple	multiply(x,y)
Simple	substitute(value,x)
Simple	subtract(x,y)
Variable Class Levels	Variable Properties
Complex	(solution_set)=given[x.vector, length>=5]
Complex	(solution_set)=given{x, y, f(x), g(y) or g[f(x)]}
Moderate	(solution_set)=given[x,f(x)]
Moderate	($\underline{x}$)=given(x.vector,length<5)
Simple	(x,y)=given(x.explicit_value,y.explicit_value)
Simple	x=given(x.explicit_unit)
Simple	x=given(x.explicit_value)
Value Properties	Value Properties
decimal.simple	non-repeating or exact decimal representations
decimal_complex	repeating decimals or decimals formed by nonintuitive rounding
fraction.complex	fractions not belonging to the value.fraction.simple class
fraction.simple	fractions that resolve to common values (e.g., 1/2=.5) or 1/n for common n
integer_large	all integers not in the class of value.integer.small
integer_small	positive integers to 2 digits, years, common values
nice_number	numbers that are easy to deal with or relate to other numbers easily

Distractor Option Classes	Description
assoc.cues	number of exact replications of information
plausibility	number of intuitively plausible distractors
oper.one-off	number of incorrect actions/instantiations needed

Context Classes	Description
verbal.complex	extensive verbal load in terms of vocabulary and sentence structure
verbal.light	low verbal load in terms of vocabulary and sentence structure
verbal.moderate	moderate verbal load in terms of vocabulary and sentence structure
visual.complex	complex graphics or visuals
visual.simple	simple or no visuals, graphics or images

Figure 2. Sample TMG elements for an arithmetic reasoning test

A given TMG statement is actually a combination of all five elements. For example, a given task model for a family of arithmetic reasoning items might be made up of a procedural skill related to converting common units (e.g., inches to feet) using positive integer values, explicit numerical links to all distractors (e.g., using similar or the same numbers improperly), reasonable plausibility of the distractors, all distractors based on faulty “one-off ” operations (e.g., multiplying instead of dividing) and only light verbal load. The corresponding elements from Figure 2 would be: (a) convert.implicit_abstract(x,X,abstract_units); (b) with the variables constrained to be x=given(x.explicit_unit); (c) values of the variables/constants specified as x∈(positive integers to 2 digits (e.g., years, common values); (d) distractors constrained to be associative.cues=explicit, distractor.plausibility=moderate, and operations.one-off=all distractors (based on units); and (e) the verbal context set to verbal.light.

Task Model Maps

Each task model has a location on the intended proficiency scale, so it is not difficult to conceptualize different concentrations of task models at different locations along the scale. Key decision points along the scale would likely receive the highest density of task models. This task model density is also a type of measurement precision specification, not unlike an IRT target test information function used in test assembly. The distribution of task models is called a *task model map* (TMM) (Luecht et al., 2010). A TMM is a formal design specification for the intended scale. It details where precision is most needed to make critical decisions or where richer interpretations of performance are needed.

Each task model has a TMG statement or expression and these expressions incorporate the relevant procedural skill requirements, the content representation and declarative knowledge load, auxiliary tools, and contextual complexity indicators. The TMG becomes an *integrated* specification for a family of items that presents the same type of task challenges to every examinee under the same conditions, and where every item generated from the task model has approximately the same difficulty (and discrimination). This is the “something else” that could replace our current content blueprints. We can subsequently use templates and a variety of statistical quality control mechanisms to ensure that the difficulties of all items within each task-model family have sufficiently low variance.

Developing a TMM involves developing policies that dictate how and where the task models are allocated along the scale. Increasing the concentration of task models at any point along the scale obviously increases the richness of performance-based interpretations. Therefore, the relative concentrations of task models on the TMM serve a dual purpose of prioritizing which proficiency claims along the construct map require the most evidence and showing the intended distribution of psychometric measurement information along the scale. Figure 3 presents four different TMMs. Each TMM represents a different concentration of 35 task models. For a longer test, we would add more task models. However, the relative densities of the task models would remain the same because each configuration represents a different purpose requiring a different concentration of measurement information and more claim-based evidence within particular regions of the scale.

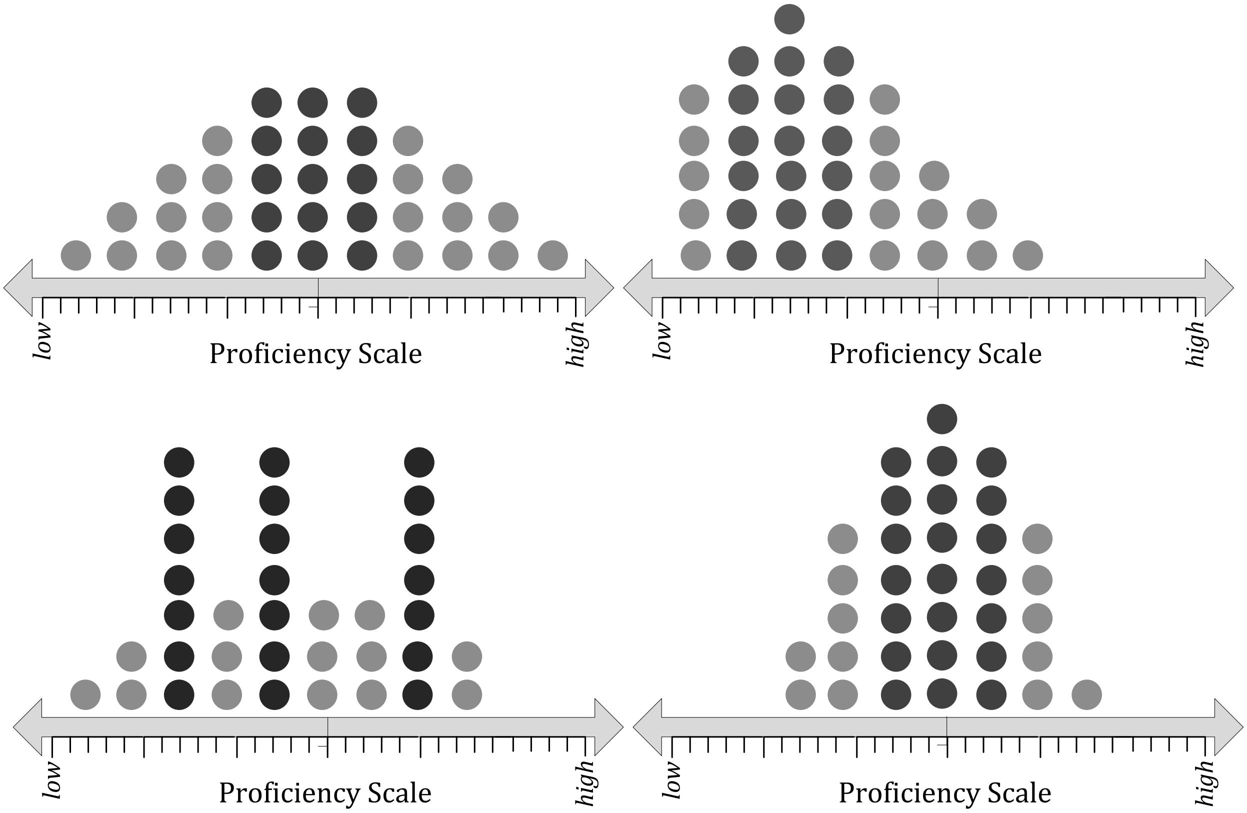


Figure 3. Four task model map configurations (35 task models each)

Each dot in Figure 3 is a task model with its own TMG expression and represents a potentially very large family of items. This approach to item creation as part of a larger task model family applies a strong engineering perspective to the design of task models and templates that allows items within statistically similar operating characteristics to be manufactured in large quantities within each task-model family. Provided that the items within each task model family maintain an acceptable degree of tolerance with respect to their locations (i.e., difficulty and minimum discrimination), both the templates associated with a particular task model and items associated with a particular template can be viewed as randomly exchangeable from both the perspectives of validity and statistical control (i.e., variation is assumed to be a random event that can be integrated out for scoring purposes). The templates must be empirically validated(possibly modified through iterative design and pilot testing) and ultimately shown to maintain the intended level difficulty and sensitivity within acceptable tolerances. An obvious benefit is that pilot testing of individual items can be relaxed and possibly eliminated altogether for well-designed templates and task models. In much the same way that a manufacturing organization retrieves schematics and builds new components as needed, the templates can be stored and used to generate new items as needed.

The links between the construct map, the TMM, and individual task model families made up of templates and items is depicted in Figure 4 for an easy, moderately difficult (complex), and difficult family. Ideally, there is one task model on the TMM for every test item. However, multiple templates can be created for each task model; moreover, multiple items can be generated from each template (Luecht, Burke, & Devore, 2009; Luecht, Dallas, & Steed, 2010).

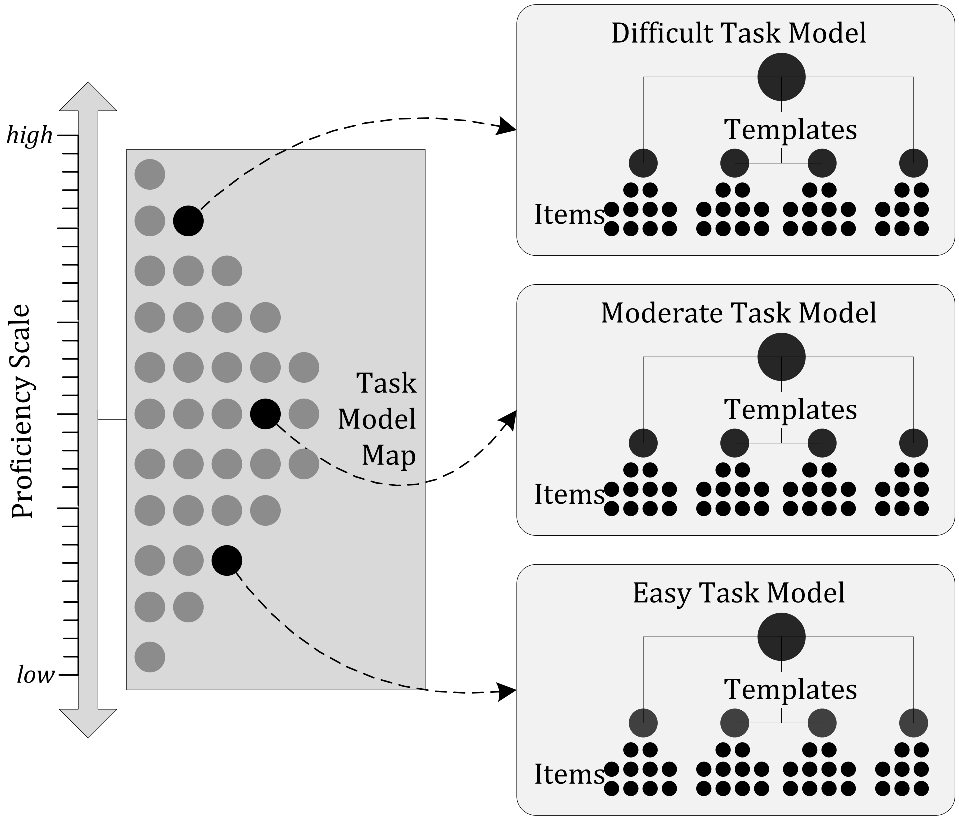


Figure 4. Hierarchical relationships between task model families, templates, and items

Locating the task models on the TMM accomplishes three important aims: (a) it makes it feasible to consider (hierarchically) calibrating the task models, rather than individual items; (b) it allows us to psychometrically evaluate the measurement precision associated with every task model and reconcile that precision with the interpretive utility of the task models in providing the intended evidence about specific claims along the construct map; and (c) it allows powerful, empirically based quality control mechanisms to be implemented to ensure that all items created for each task model family perform in a statistically similar manner. Operationally speaking, each task model is actually represented by multiple, exchangeable templates. In turn, each template can be used to generate many items.

Templates and Writing Items

The strong manufacturing-engineering aspects of AE demand that every component, every assembly of components, and every system of assemblies is carefully specified by design parameters and verified post-production through QC procedures. A task model is a carefully constructed specification for a family of items that all offer the same level of challenge to the examinees. The intended task challenge should be more or less difficult than other challenges and provide acceptable evidence about the proficiency claims of interest along the construct map. If the task model challenge is intentionally altered (i.e., made more or less complex), the difficulty of items associated with that task model should change accordingly.

Changes in difficulty should not be random outcomes that occur because creative item writers decide to add uncontrolled and sometimes unknown sources of complexity to the items. This again characterizes the fundamental problem with most traditional test specifications. Because content blueprints tend to ignore cognitive complexity as part of the description for each content category, they inherently fail to provide any concrete guidance to item writers about the intended difficulty of items generated within a particular content category. As a result, it becomes virtually impossible to hold item writers accountable for the adherence-to-design quality of their items because there was never a concrete design specification. At best, we might evaluate the statistical quality of their items after experimental pilot testing.

Under AE, the item writers are constrained to include the intended procedural skills and knowledge challenge implied by the task model. The hierarchical relationship between task models, templates, and items provides an important "manufacturing" improvement on item design and item writing because it provides a firm criterion—the quality in producing items that maintain the desired statistical and cognitive characteristics of the task model. Under AE, we depend heavily on ongoing quality assurance (QA) procedures to evaluate item writing procedures or outputs, attempting to detect aberrations that may signal a need to make alterations to the template design, the training of item writers, or quite possibly—leading to a decision to find new and "less creative" item writers. AE also requires strong and proactive adherence to statistical quality control (QC) criteria. The AE templates and item writers are part of an item manufacturing system, where QC and QA mechanisms represent ongoing efforts to make sure that the items in each task-model family are statistically behaving as intended.

From Task Models to Templates

Consider an example from the Common Core State Standards (CCSS):

> Calculate expected values and use them to solve problems: (S-MD.3.) Develop a probability distribution for a random variable defined for a sample space in which theoretical probabilities can be calculated; find the expected value. (CCSS Initiatives Project; www.corestandards.org/the-standards/mathematics/hs-statistics-and-probability/)

A plausible set of TMG statements to represent this standard might be created as shown in Figure 5. There are alternate ways to represent the types of tasks implied by the standard. These statements represent one possibility. Also, this task model is obviously generated outside the context of the broader CCSSs.

$$\text{Recall.formula.SRS\_uniform.discrete}\left[p_i = P_i\left(u_i = 1|a\right) = {}^{1}\!/_{a}\right]$$
$$\text{Recall.formula.expected\_value}\left[E\left(y\right) \doteq \bar{y} = \textstyle\sum_{i=1}^{n} p_i u_i\right]$$
$$\text{Apply.formula.sum\_products}\left[\bar{y} = \textstyle\sum_{i=1}^{n} p_i u_i = p_1 u_1 + p_2 u_2 + \cdots + p_n u_n\right]$$
$$\text{Apply.formula.simplify\_distributive}\left[\textstyle\sum_{i=1}^{n} p u_i = pn \middle| p = 1/a\right]$$
$$\text{Constraint.value.discrete\_int}\left[u_i \in \left(0,1,\ldots,u^{\max}\right)\right]$$
$$\text{Constraint.value.discrete\_int}\left[n \in \left(2,\ldots,n^{\max}\right)\right]$$
$$\text{Constraint.value.prob}\left[0.0 \le p \le 1.0\right]$$

Figure 5. TMG elements for CCSS S-MD.3

A sample item that requires the TMG elements in Figure 5 is displayed in Figure 6. Again, this is merely an example.

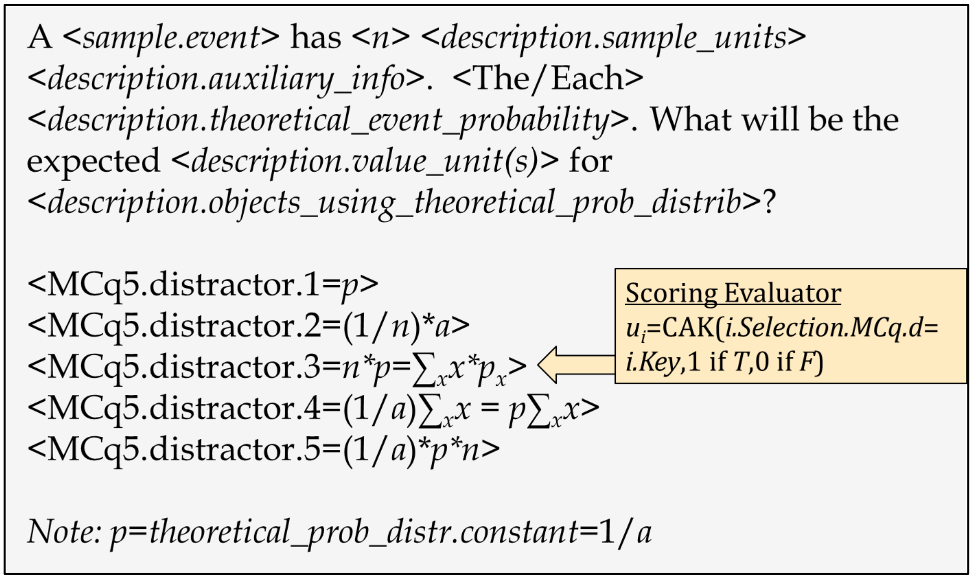
A test has five multiple-choice questions scored correct/incorrect. Each question has four possible options. What will be the expected number-correct score for students who guess the answers to all five of the questions?

A. 0.25
B. 0.80
C. 1.25
D. 3.75
E. 5.00

Figure 6. A sample item for CCSS S-MD.3, based on seven TMG expressions

With the sample item from Figure 6 and the more general TMG expressions in Figure 5, we can present the rendering model for a particular template that might generate a large number of items measuring the CCSS standard. AE task templates are actually comprised of three components: (a) the template *rendering model* controls the look-and-feel of the item and follows a specific item type format; (b) the template *scoring evaluator* controls which response or other data is collected and how it is used (i.e., the rubric format); and (c) the template *data model* contains all of the relevant data for presenting and scoring the items (i.e., the data used by the rendering model and scoring evaluators for that template). These three template components jointly specify the presentation format, manipulable content, and scoring rules for each template.

A possible rendering model associated with our prototype statistics item (Figure 6) is shown in Figure 7. This rendering model incorporates shells for each distractor as well as for the correct answer. The scoring evaluator—a correct answer key (CAK) one-best answer matching evaluator—is also depicted. The data model might constrain such elements as the numerical values, plausible lists of sample events, and acceptable probability distributions that could be included.



A <*sample.event*> has <*n*> <*description.sample_units*> <*description.auxiliary_info*>. <The/Each> <*description.theoretical_event_probability*>. What will be the expected <*description.value_unit(s)*> for <*description.objects_using_theoretical_prob_distrib*>?

<MCq5.distractor.1=p>
<MCq5.distractor.2=$(1/n)*a$>
<MCq5.distractor.3=$n*p=\sum_x x*p_x$>
<MCq5.distractor.4=$(1/a)\sum_x x = p\sum_x x$>
<MCq5.distractor.5=$(1/a)*p*n$>

Note: p=theoretical_prob_distr.constant=1/*a*

Scoring Evaluator
u_i=CAK(*i.Selection.MCq.d*= *i.Key*,1 if *T*,0 if *F*)

Figure 7. A Sample Template Rendering Model for CCSS S-MD.3

The templates may require extensive pretesting to validate their "controls." Although a particular task model is instantiated by multiple items, the *design* specification calls for all of those items to have a distinct location on the proficiency scale. It is the role of the template, combined with strong QA and QC mechanisms, to maintain each task-model's location and ensure that associated items do not vary in their operating characteristics. Realistically, some variation in the item statistics is expected and is fully acceptable at the level of templates and again at the level of items. These minimally acceptable levels of variation, in an engineering sense, are viewed as manufacturing *tolerances.* Acceptable tolerances can be empirically derived through simulation (Shu, Burke, & Luecht, 2010) or analytical means using pilot test results. Tolerances can be established relative to four sources of variation: (a) variance due to item difficulty and discrimination relative to other items generated from the same template (e.g., item discrimination, estimated "guessing" proportions, and covariances among the estimates); (b) variance in the template difficulties relative to other templates generated from the same task model; (c) intentional and unintentional scoring dependencies introduced via the scoring evaluators; and (d) residual covariance associated with templates and items that might lead to psychometric scaling complications (e.g., multidimensionality, differential item functioning).

Ultimately, AE depends on empirical field trials and pilot testing to experimentally detect statistical anomalies and manipulate the template controls to reduce the influences of these sources of variance. Once the tolerances are determined, they are incorporated into the data model for each template. If needed, the templates may need to be modified by tightening the rendering model, scoring evaluator rules, or data model controls to minimize variation and maintain the established engineering tolerances within templates.

Assessment engineering templates help ensure reusability and scalability of various data structures, content, and scoring components across items in that family. It is relatively straightforward to devise database structures to represent these reusable, scalable components of the templates. From a test development perspective, the use of templates reduces the possibility for item writers to "creatively" redefine the construct and/or task model complexity. Developing and empirically validating multiple templates for each task model can be very complicated, but is absolutely necessary. The templates turn each conceptual task model into real test items. The good news is that, once AE is implemented and working, item writing should be far less complicated and costly; it certainly will be more predictable than basing item-writing assignments on overly broad content specifications. An item writer's task has two possible responsibilities: (a) to provide *plausible* values for assessment task features (i.e., aspects of the task challenge, context, or resources allowed by the template) that effectively control for difficulty and/or eliminate sources of nuisance dimensionality; and (b) to provide surface-level, incidental changes to the rendering form of the item to provide multiple instantiations of each item.

Hierarchical Calibrations and Quality Control

Assessment engineering is not an exploratory endeavor to find a statistical signal (i.e., factor or latent trait) that appears to explain pattern of covariances among items. It requires a strong confirmatory perspective about what a scale is supposed to tell us about the examinees. Under AE, the statistical item characteristics we typically estimate are intentional (and predictable) outcomes based directly on the decisions that went into the principled design of the task models and templates. Strong QA and QC procedures are therefore required to ensure that the item manufacturing system is working as intended to maintain the score scale. If aberrations exist, we must iteratively modify the design until it does work as intended. That is engineering the assessment.

The idea of using task models and templates provides a convenient way of implementing design-based controls over the unwanted sources of statistical variation to ensure consistent interpretation and inferences about the construct from the task model challenge characteristics and from the data. Focusing on a task model as the primary unit of interest implies that we can modify the items and templates to ideally manage and variation within task models. The rather common statistical goal of minimizing random and systematic error variance suggests that these are credible QC criteria for evaluating the utility of multiple templates associated with a particular task model. That is, if the variation is sufficiently small, the task model and templates are considered "usable". If too much variation exists, the templates must be adapted until an appropriate minimum amount of variance can be consistently achieved within the target population.

In practice, empirical field trials and simulations based on real data can be employed to determine the acceptable tolerances at the levels of templates and items. Glas and van der Linden (2003) and Geerling, Glas, and van der Linden (2011) have demonstrated a promising hierarchical IRT calibration framework that serves AE purposes. IRT calibration also takes on a new role under AE: to provide empirically based, *quality-control evidence* that test developers can use to guide the redesign of the templates and ultimately reduce undesirable variation in the item operating characteristics for every item generated from a particular task model. This iterative redesign and QC feedback process may seem to be time-consuming and expensive. However, it pays off by an order of magnitude in the end because enough items can generated from every well-behaved task model to all but eliminate security/item over-exposure risks and pretesting. Furthermore, it is possible to use the natural hierarchy between task models, templates, and items to either collapse the data and/or employ hierarchical IRT calibration models that result in more stable estimates of the item statistics (Shu et al., 2010).

Discussion

In contrast to the many important advances in technology and psychometric modeling, innovations in test development have largely been non-existent—unless, of course, we count technology-enhanced item types and a recurring interest in performance-based items to be innovative. Measurement professionals continue to depend on SMEs to dictate what to measure and how to best measure those constructs, usually waiting until afterward the assessments are designed and data is collected to apply a particular psychometric scaling or calibration model to a large data set in the hope of extracting a clear and consistent statistical signal from the empirical data.

Assessment engineering represents a different way of thinking about the problem of designing test specifications—keeping the notion of a scale (i.e., the eventual and desired outcome) in mind from the beginning and articulating how the scale is to be interpreted, including the types of evidence needed to support those inferences and interpretations. It is only after developing the vision of the construct that we begin to carefully and systematically build our scale: from task models to templates to items. There are numerous checks and balances along the way, and iterative design changes are to be expected.

Will assessment engineering work? In the worst case, it provides a systematic way to approach item and test design. In that event, we build and design tests as we now do; albeit, with perhaps more detailed specifications to guide item writers. In the best case, we design well-articulated scales that can be maintained with an almost endless supply of items that do not require huge samples and complicated psychometric models maintain the scales. It seems worth the effort to try for the best case.

Assessment engineering requires a dedicated commitment to iterative design and refinement. From a practical perspective, the short-term benefits of considering assessment engineering as a replacement for conventional item writing, test assembly and calibration may seem small. The simple fact is that nobody is currently using this assessment engineering approach. But consider the more traditional alternative which is to write a limited supply of rather high-cost items to very vague content blueprints, where each item has a very limited shelf life. We then throw as much response data from those items as possible at a psychometric calibration model and hope that some it sticks to produce a scale. Finally, we add post hoc interpretations to the scale (e.g., writing performance level descriptors and setting cut scores). This traditional approach continually requires more high-cost items to replenish exposed items and change often takes years to implement. If we still built automobiles the way we build tests, cars would probably cost tens or hundreds times more than they do now. And if we treated notebook computers like we treat items, we would use the computer once and then either shelve it for limited reuse or discard it. There is a better way.

Considered over the longer term, the cost and psychometric benefits of adopting a more principled approach to item design and item writing—an approach that requires a more in-depth understanding of the underlying inferences being made all along the score scale as well as providing potentially large quantities of lower-cost items via empirically proven templating and associate quality control procedures—at least deserve a chance. Some of the more recent experiences the author has had with task modeling in mathematics, science, and reading comprehension suggests that task modeling is difficult and tedious, over requiring multiple attempts, feedback loops, and as much empirical modeling and experimentation as possible, That is to be expected because it is a new way of thinking about content and test design. Those same experiences are also beginning to show some solid potential for success.

As noted above, in the worst case, assessment engineering provides a better definition of the construct as a detailed progression of expected knowledge and skill, possibly adds more documentation and principled item and test development practices, but ultimately may still require pilot testing and calibrating every item using large samples. In short, under a worst case scenario, assessment engineering merely does what we do now. In the better case, it may help us actually understand what we are measuring—that is, what the numbers actually mean—as well as lowering the longer term costs of item production by an order of magnitude. Capitalizing on hierarchical calibrations of templates and task models, and investing iterative design and refinement, the less data is required for task models and templates that meet established quality control thresholds, When we stop treating items as high-cost, low use commodities with a limited shelf life, long-term costs decline. For example, if a testing program has historically spent on average $300 US per item (factoring in item writing, pilot testing publication and processing costs), and an assessment engineering template costs $600 US, the latter may not seem worthwhile. However, if the controlled production of items for a task model templates eventually eliminates much of the pilot testing of items and the associated template can generate 400 items, item exposure risks go down by an order of magnitude and costs per item drop to $1.50 US. That is the potential of assessment engineering.

References

Anastasi, A. (1986). *Psychological testing* (6th ed.). New York, NY: Macmillan.

Bejar, I., Braun, H. I., & Tannenbaum, R. J. (2007). A prospective, predictive, and progressive approach to standard setting. In Lissitz, R. (Ed.), *Assessing and modeling cognitive development in school: Intellectual growth and standard setting*. Maple Grove, MN: JAM Press.

Bejar, I. I., & Yocom, P. (1991). A generative approach to the modeling of isomorphic hidden-figure items. *Applied Psychological Measurement, 15*(2), 129-137.

Bloom, B. S., & Krathwohl, D. R. (1956). *Taxonomy of educational objectives: The classification of educational goals, by a committee of college and university examiners. Handbook 1: Cognitive domain*. New York, NY: Longmans.

Geerlings, H., Glas, C. A. W., & van der Linden, W. J. (2011). Modeling rule-based item generation. *Psychometrika, 76*, 337-359.

Glas, C. A. W., & van der Linden, W. J. (2003). Computerized adaptive testing with item cloning. *Applied Psychological Measurement, 27*, 247–261.

Guion, R. M. (1977). Content validity: The source of my discontent. *Applied Psychological Measurement*, *1*, 1-10.

Hambleton, R. K., & Pitoniak, M. P. (2006). Setting performance standards. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 433-470). Westport, CT: American Council on Education and Praeger.

Kane, M. T. (2006). Validation. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 17-64). Westport, CT: American Council on Education and Praeger.

Kintsch, W. (1988). The role of knowledge in discourse comprehension construction-integration model. *Psychological Review, 95*, 163-182.

Luecht, R. M. (2006). *Engineering the test: From principled item design to automated test assembly*. Invited paper presented at the annual meeting of the Society for Industrial and Organizational Psychology, Dallas, Texas.

Luecht, R. M. (2007, October). *Assessment engineering: An integrated approach to test design, development, assembly, and scoring*. Invited keynote and workshop presented at the Performance Testing Council Summit, Scottsdale, AZ.

Luecht, R. M. (2008a, February). *Assessment engineering*. Session paper at Assessment Engineering: Moving from Theory to Practice, Coordinated panel presentation at the Annual Meeting of the Association of Test Publishers, Dallas, TX.

Luecht, R. M. (2008b, February). *The application of assessment engineering to and operational licensure testing program*. Paper presented at the Annual Meeting of the Association of Test Publishers, Dallas, TX.

Luecht, R. M. (2008c, October). *Assessment engineering in test design, development, assembly, and scoring*. Invited keynote address at the Annual Meeting of the East Coast Organization of Language Testers (ECOLT), Washington, D.C.

Luecht, R. M. (2009, June). *Adaptive computer-based tasks under and assessment engineering paradigm*. Paper presented at the 2009 GMAC Conference on Computerized Adaptive Testing, Minneapolis, MN (published online in *Proceedings*. http://www.psych.umn.edu/psylabs/catcentral/).

Luecht, R. M. (2010, April). *Controlling difficulty and security for complex computerized performance exercises using assessment engineering.* Paper presented at the Annual Meeting of the National Council on Measurement in Education, Denver, CO.

Luecht, R. M. (2011, March). *Assessment design and development, version 2.0: From art to engineering*. Invited, closing keynote address at the Annual Meeting of the Association of Test Publishers, Phoenix, AZ.

Luecht, R. M., Burke, M., & Devore, R. (2009, April). *Task modeling of complex computer-based performance exercises.* Paper presented at the Annual Meeting of the National Council on Measurement in Education, San Diego, CA.

Luecht, R. M., Dallas, A., & Steed, T. (2010, April). *Developing assessment engineering task models: A new way to develop test specifications*. Paper presented at the Annual Meeting of the National Council on Measurement in Education, Denver, CO.

Luecht, R. M., & Masters, J. (2010, February). *The efficiency of calibrating multiple-item templates and task models using a hierarchical calibration model*. Paper presented at the Annual Meeting of the Association of Test Publishers, Orlando, FL.

Masters, J. S., & Luecht, R. M. (2010, April). *Assessment engineering quality assurance steps: Analyzing sources of variation in task models and templates*. Paper presented at the Annual Meeting of the National Council on Measurement in Education, Denver, CO.

Messick, S, (1989). Validity. In R. L. Linn (Ed.), *Educational measurement* (3rd ed., pp. 13-103). New York, NY: American Council on Education and Macmillan.

Mislevy, R. J. (1994). Evidence and inference in educational assessment. *Psychometrika, 59*, 439-483.

Mislevy, R. J. (2006). Cognitive psychology and educational assessment. In R. L. Brennan (ed.), *Educational measurement* (4th ed., pp. 257-305). Westport, CT: American Council on Education and Praeger.

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

Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (2003). On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives, 1*, 3-67.

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

Shu, Z., Burke, M., & Luecht, R. M. (2010, April). *Some quality control results of using a hierarchical bayesian calibration system for assessment engineering task models, templates, and items*. Paper presented at the Annual Meeting of the National Council on Measurement in Education, Denver, CO.

Webb, N. L. (April, 2005). *Issues related to judging the alignment of curriculum standards and assessments.* Paper presented at the Annual Meeting of the American Educational Research Association Meeting, Montreal, QB, Canada.

Wilson, M. (2005). *Constructing measures*. Mahwah, NJ: Lawrence Erlbaum.