Assessment Engineering in Test Design, Development, Assembly, and Scoring

Richard M. Luecht, Ph.D.
University of North Carolina at Greensboro
07 November, 2008
East Coast Organization of Language Testers (ECOLT) Conference
Assessment Engineering (AE)

- AE provides an integrated framework with replicable, scalable solutions for assessment design, item writing, test assembly, and psychometrics.
- Possible applications are being explored for multidimensional, K-12 classroom formative assessments.
- Current applications are actually being developed for large-scale, summative assessment applications (e.g., the Uniform CPA Examination, AP, and the PSAT).
Assessment Engineering (AE)

- AE begins with the development of one or more **construct maps** that describe concrete, **ordered** performance expectations at various levels of a proposed **scale**
- *Empirically driven* **evidence models** and **cognitive task models** are developed at specified levels of each construct, effectively replacing traditional test blueprints and related specifications
- Multiple assessment **task templates** are **engineered** for each task model to control item difficulty, covariance, and residual errors of measurement
- Psychometric procedures are used as **statistical quality assurance mechanisms** that can **directly** and **tangibly** hold item writers and test developers accountable for adhering to the intended test design
Why is AE Useful? Necessary?

- Psychometric models are “data hungry”
  - Sparse data is a serious problem for IRT and other psychometric models re calibration
  - AE can reduce item exposure risks by expanding item banks in a principled way
  - AE assessments capitalize on replication to reduce item production costs and overall pretesting costs
- Strong, empirically based quality control (QC) mechanisms can be implemented to improve test development in a concrete way
- AE is fully consistent with advanced psychometric models for calibration, equating, and scaling (e.g., hierarchical Bayes estimation and so-called cognitive diagnostic models and related constrained latent class models)
Recent Developments

- Task design frameworks are making progress
  - Evidence-centered design (ECD, Mislevy & Almond)
  - Integrated test design, development, and delivery (ITD³, Luecht)
  - AE design of accounting simulations (Luecht, Gierl, and Devore, 2007; Luecht, Burke, & Devore, 2008)
  - Language testing (Kenyon, 2007; Tucker, 2008)
- Applications to diagnostic testing are emerging
  - Attribute-hierarchy model (AHM, Gierl & Leighton)
  - ECD-like applications (Huff; Perlman)
  - Principled assessment designs for inquiry (PADI, Wilson & Mislevy)
  - Task modeling (Luecht, Burke, & Devore, Masters & Luecht, Gierl & Leighton; Luecht & Gierl)
Five AE Processes

- Construct mapping
- Evidence modeling
- Task modeling and construct blueprinting
- Template design and item writing
- Psychometric QC/QA, Calibration, and Scoring
Traditional Views of the Transition from Construct Spaces to Latent Trait Scales

Semantic Relations

Reading Comprehension

Encoding

Syntactic Management

What was the author’s purpose?
A. To inform
B. To illustrate
C. To persuade
D. To obsfuscate

You are here!

Proficiency Scale

\[ U_j = \left( u_{1j} = 1, u_{2j} = 1, u_{3j} = 0, \ldots, u_{nj} = 0 \right) \]

\[ = (110 \ldots 1) \]

\[ \frac{\partial L}{\partial \theta} = \sum \left( u - P \right) \frac{\partial P}{\partial \theta} \]
K-12 Language Proficiency

<table>
<thead>
<tr>
<th>Bridging</th>
<th>Expanding</th>
<th>Developing</th>
<th>Beginning</th>
<th>Entering</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Linguistic Complexity</td>
<td>Vocabulary Usage</td>
<td>Language Control</td>
<td>L5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L1</td>
</tr>
</tbody>
</table>

Kenyon, D. (Nov., 2007). *Examining a large-scale language testing project through the lens of assessment engineering: What can language testers learn?* Keynote address at the Sixth Annual ECOLT Conference, Washington, DC
Drilling Down…

1. Determining the meaning of words and sentences

1.a Word affixes, roots, and cognates

1.b Syntax

1.c Context

1.d Word recognition

Morphing through Dimensionally More Complex States

State 1: Pronunciation and vocabulary function autonomously

State 2: Grammar emerges

State 3: grammar, pronunciation and vocabulary merge; fluency and sociolinguistics emerge

Due to automaticity, pronunciation and vocabulary become indistinguishable

2008 R.M. Luecht

Tying Complexity to Cognition

- Language contexts
  - **Tasks:** more communication tasks → greater complexity
  - **Topics:** more topics → greater complexity
  - **Information density:** higher structural density of text or speech samples → greater complexity

- Cognitive task challenges
  - **Conceptual Knowledge:** facts, rules, regulations that form the core database for the practitioner
  - **Process Skills:** concrete applications and “how to do [this]”
  - **Evaluation and Synthesis:** reasoning, comparing, contrasting and making inferences or deductions (includes meta-cognition)
A “New” Perspective on Complexity and Dimensionality
AE and Construct-Based Design

- Constructs should be articulated in terms of ordered, hierarchical levels of procedural knowledge and skills, or in terms of levels of cognition applied to well-defined content strands.
  - We call the ordered statements that define a construct claims” or “assertions”
  - Claims are in service of particular decisions along an ordered continuum (fail → pass; 50, 51,…,100, etc.)
- Higher-level claims subsume lower-level claims
- All salient constructs should be specified, along with the expected patterns of relationships among the constructs.
- Ultimately...focus on the proficiency claims we wish to make with respect to a specific number of useful, interpretable score scales
A Construct Map (Wilson, 2005)

Increasing X

Respondents

High Level of Construct X

High Item Response Scores Require Highest Level of X

Item Responses

Item Response Scores Require Moderate Level of X

Mid-level Level of Construct X

Decreasing X

Low Level of Construct X

Response Scores Require Lowest Level of X

“X” can represent a continuum or an ordered set of latent classes

Item locations denote score properties of multiple items with similar characteristics
Binet & Simon (1905): Intelligence

Respondents
- 2-3 years old
  - Follow five verbal orders (e.g., touching nose, mouth, eyes)
  - Name familiar objects in a picture
- 7-8 years old
  - Describe pictures
  - Count mixed coins
  - Compare two objects from memory
- 9-10 years old
  - Arrange weights
  - Answer to comprehensive questions
  - Form sentences from 3 words
  - Interpret pictures
  - Create rhymes

Item Responses
- Increasing Intelligence
  - Arrange weights
  - Answer to comprehensive questions
  - Form sentences from 3 words
  - Interpret pictures
  - Create rhymes
- Decreasing Intelligence
  - Follow five verbal orders (e.g., touching nose, mouth, eyes)
  - Name familiar objects in a picture

2008 R.M. Luecht
Claims: Examples

- Can *evaluate* the basic distinctions among different <types of entities>
- Can *compare* the effectiveness of components of a system in a specific context
- Can *perform* <appropriate analysis> procedures to assess risk
- Can *prepare* documentation of an operational procedure
What is Construct Mapping (Wilson, 2005; Luecht, 2007)?

- Benjamin Bloom (1956) defined a well-known progression of cognitive skills: knowledge $\rightarrow$ comprehension $\rightarrow$ application $\rightarrow$ analysis $\rightarrow$ synthesis $\rightarrow$ evaluation

- Marzano (2000) reformulated the progression as conceptual knowledge (declarative knowledge), process skills (procedural knowledge), and evaluation and synthesis (includes meta-cognition, both declarative and procedural)

- Construct mapping amounts to clearly documenting a progression of ordered claims about proficiencies and skills and the required observable evidence needed to make those claims.
Claims and Construct Maps

**Increasing Proficiency**

**Profiency Claims**
- Can evaluate complex systems
- Can analyze mid-level systems
- Can define Audit concepts

**Task Models**
- Evaluates(Analyzes(Researches(system1,system2 | tools)))
- Analyzes(system1,system2 | tools)
- Match(concept,definition)

**Decreasing Proficiency**
Construct-Based Validation is NOT New (Messick, 1994)

“A construct-centered approach [to assessment design] would begin by asking what complex of knowledge, skills, or other attributes should be assessed, presumably because they are tied to explicit or implicit objectives of instruction or otherwise valued by society. Next, what behaviors or performances should reveal those constructs, and what tasks or situations should elicit those behaviors?” (p. 16)
Scores ➔ Scales ➔ Construct Maps

<table>
<thead>
<tr>
<th>Pattern #</th>
<th>Assessment Tasks</th>
<th>Increasing Proficiency</th>
<th>Task Models</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Performance</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4 4 4 4 4 4 4 4</td>
<td>Evaluates, interprets,</td>
<td>XXX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>researches, and analyzes</td>
<td>XXXXX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>multivariable systems</td>
<td>XXXXXXX</td>
</tr>
<tr>
<td>2</td>
<td>3 4 4 4 4 4 4 4</td>
<td></td>
<td>XXXXXXXXXX</td>
</tr>
<tr>
<td>3</td>
<td>3 3 4 4 4 4 4 4</td>
<td></td>
<td>XXXXXXXXXXX</td>
</tr>
<tr>
<td>4</td>
<td>3 3 3 4 4 4 4 4</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>5</td>
<td>3 3 3 3 4 4 4 4</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>6</td>
<td>3 3 3 3 3 4 4 4</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>7</td>
<td>3 3 3 3 3 3 4 4</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>8</td>
<td>3 3 3 3 3 3 3 4</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>9</td>
<td>3 3 3 3 3 3 3 3</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>10</td>
<td>2 3 3 3 3 3 3 3</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>11</td>
<td>2 2 3 3 3 3 3 3</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>12</td>
<td>2 2 2 3 3 3 3 3</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>13</td>
<td>2 2 2 2 3 3 3 3</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>14</td>
<td>2 2 2 2 2 3 3 3</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>15</td>
<td>2 2 2 2 2 2 3 3</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>16</td>
<td>2 2 2 2 2 2 2 3</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>17</td>
<td>2 2 2 2 2 2 2 2</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>18</td>
<td>1 2 2 2 2 2 2 2</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>19</td>
<td>1 1 2 2 2 2 2 2</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>20</td>
<td>1 1 1 2 2 2 2 2</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>21</td>
<td>1 1 1 1 2 2 2 2</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>22</td>
<td>1 1 1 1 1 2 2 2</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>23</td>
<td>1 1 1 1 1 1 2 2</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>24</td>
<td>1 1 1 1 1 1 1 2</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>25</td>
<td>1 1 1 1 1 1 1 1</td>
<td></td>
<td>XXXXXXXXXXXX</td>
</tr>
</tbody>
</table>

2008 R.M. Luecht
Evidence Models
Construct Maps and Evidence Models

Claims

- Handling complex operations on complex systems of objects
- Exceptional accuracy/speed performing moderately complex operations
- Handling moderately complex operations on simple systems
- Average accuracy/speed performing relatively simple operations
- Performing simple operations on small sets of objects
- Low accuracy/speed performing simple operations

Evidence

- Complex work products done very quickly with perfect accuracy
- Highly challenging work tasks done very quickly with perfect accuracy
- Routine work products done very quickly with near perfect accuracy
- Moderately challenging work: average time and accuracy
- Simple, isolated tasks done with marginal accuracy
- Very simple tasks done taking too much time and with marginal accuracy

2008 R.M. Luecht
Evidence Models

- An evidence model is a documented specification of the universe of tangible actions, responses, and/or products that would qualify as evidence for a particular proficiency claim...it is a repository of plausible performance tasks for every claim.
- Each claim should have one or more evidence models.
- Task models are composed directly from the evidence models.
- Components of an evidence model include:
  - Valid settings or contexts
  - The plausible range of challenges for the target population
  - Relevant actions that could lead to a solution
  - Dangerous or inappropriate actions
  - Legitimate auxiliary resources, aids, tools, etc. that can be used to solve the problem
  - Concrete exemplar products of “successful performance”
Using Practice Analysis Skills (S) and Tasks (T) to Map Evidence Models to a Research & Analysis Construct for Accounting

<table>
<thead>
<tr>
<th>Skill=</th>
<th>Constr=RAI</th>
<th>LOW n(Tasks)=</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>S435</td>
<td></td>
<td>2:</td>
<td>T088  T089</td>
</tr>
<tr>
<td>S430</td>
<td></td>
<td>18:</td>
<td>T072  T073 T076 T083 T084 T085 T086 T087 T088</td>
</tr>
<tr>
<td>S447</td>
<td></td>
<td>2:</td>
<td>T087  T110</td>
</tr>
<tr>
<td>S432</td>
<td></td>
<td>20:</td>
<td>T065  T070 T074 T075 T077 T078 T080 T081 T097</td>
</tr>
<tr>
<td>S431</td>
<td></td>
<td>14:</td>
<td>T065  T072 T082 T083 T084 T087 T089 T097 T103</td>
</tr>
<tr>
<td>S433</td>
<td></td>
<td>9:</td>
<td>T069  T072 T085 T086 T088 T089 T093 T094 T109</td>
</tr>
<tr>
<td>S437</td>
<td></td>
<td>6:</td>
<td>T074  T085 T086 T093 T105 T109</td>
</tr>
<tr>
<td>S459</td>
<td></td>
<td>5:</td>
<td>T089  T091 T094 T095 T122</td>
</tr>
<tr>
<td>S439</td>
<td></td>
<td>20:</td>
<td>T069  T073 T074 T075 T076 T077 T078 T080 T082</td>
</tr>
<tr>
<td>S441</td>
<td></td>
<td>5:</td>
<td>T071  T072 T085 T105 T106</td>
</tr>
<tr>
<td>S445</td>
<td></td>
<td>3:</td>
<td>T089  T095 T109</td>
</tr>
<tr>
<td>S446</td>
<td></td>
<td>8:</td>
<td>T086  T087 T091 T095 T103 T105 T109 T118</td>
</tr>
<tr>
<td>S426</td>
<td></td>
<td>5:</td>
<td>T070  T076 T085 T093 T106</td>
</tr>
<tr>
<td>S414</td>
<td></td>
<td>10:</td>
<td>T065  T069 T070 T073 T074 T076 T085 T092 T103</td>
</tr>
<tr>
<td>S440</td>
<td></td>
<td>13:</td>
<td>T071  T072 T080 T089 T091 T093 T094 T095 T096</td>
</tr>
<tr>
<td>S448</td>
<td></td>
<td>19:</td>
<td>T080  T081 T082 T083 T084 T086 T087 T089 T092</td>
</tr>
<tr>
<td>S442</td>
<td></td>
<td>30:</td>
<td>T065  T069 T070 T073 T074 T075 T077 T078 T080</td>
</tr>
<tr>
<td>S438</td>
<td></td>
<td>23:</td>
<td>T063  T073 T080 T081 T082 T083 T084 T089 T091</td>
</tr>
<tr>
<td>S458</td>
<td></td>
<td>22:</td>
<td>T063  T071 T072 T080 T085 T086 T087 T089 T092</td>
</tr>
<tr>
<td>S443</td>
<td></td>
<td>21:</td>
<td>T082  T083 T084 T086 T089 T092 T093 T094 T095</td>
</tr>
</tbody>
</table>

The *Trajectory* of a Claims and Evidence Along a Construct

Cognitive Skills Trajectory

Advanced Knowledge Utilization
Comprehending
Identifying/Applying Simple Relations, Procedures or Calculations
Description, Recall, or Recognition

Incremental Cognitive Skills
Incremental Knowledge/Skills Mixture

Multiple Objects with Complex Properties
Multiple Objects with Complex Relations
Multiple Objects with Complex Relations and Relations Among Properties

Vocabulary and Simple Concepts

Declarative Knowledge

Align the evidence models with a particular trajectory through this skill-by-knowledge object space

2008 R.M. Luecht
Task Modeling and Construct Blueprinting
Construct Maps and Targeted Measurement Information

- Measurement information is largely a function of two statistical characteristics of assessment tasks
  - The **difficulty** of each item (i.e., its “location” with respect to some score scale)
  - The **sensitivity** of the item to the underlying construct being measured (i.e., discriminating power of the item)
- We can **TARGET** measurement information where it is needed most by controlling the **difficulty** of the assessment tasks
- Under AE, we must jointly control sensitivity to the construct of interest and “nuisance” dimensionality via **task models and templates**
Features of Test Measurement Information

- Each item contributes a unique amount of information at specific score values.
  - The item information functions are independent of one another for different items.
  - A TIF does not depend on any particular items being included in the test.

- Under scaling methods such as IRT, the test information functions are directly proportional to the error variance associated with the estimates of θ (EAPs or MLEs).
Target Information Functions Tied to *Decisions*

![Graph showing target information functions tied to decisions]

- Test 1
- Test 2
- Test 3

- Beginning vs. Developing
- Developing vs. Expanding

2008 R.M.

31
How AE Works to Target Measurement Information

- Measurement precision is targeted to specific regions of the construct map.
- Evidence models define the universe of knowledge and skill tasks that might provide credible, observable, and concrete evidence about the proficiency claims at various levels of the construct.
- Task models are composed from evidence model components and are stacked in the greatest numbers where to approximate the density of measurement precision needed.
- Multiple task templates are constructed and empirically validated for each task model.
- Task templates are used by item writers to generate exchangeable performance assessment tasks to meet demands.
Density of Task Models Proportion to Measurement Precision Needs

Performance

Integrates and interprets discourse-level text

Interprets sentential level text

Encodes, recognizes, and interprets salient lexical patterns

Encodes and defines whole words

Spelling and letter/symbol identification

Increasing Proficiency

Task Model

XXX

XXXXXXXXXXXXX

XXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXX

Each “X” denotes a class of items that perform similarly and provide similar evidence about a claim

Decreasing Proficiency
A Construct Blueprint with a Task Model Distribution of Measurement Opportunities for Writing Proficiency

Performance

- Fluent written expression with detail regarding most common topics
- Creates routine social correspondence and documentary materials required for most limited work requirements
- Sufficient control of writing system to meet most survival needs and limited social demands
- Has sufficient control of the writing system to meet limited practical needs using familiar concepts
- Writes using memorized material and set expressions
- No functional writing ability

Task Model

- Decreasing Proficiency
- XXX
- XXX
- XXX
- No functional writing ability

Increasing Writing Proficiency

- XX
- XXX
- XXXXXXXXXX
- XXXXXXXXXX
- XXXXXXX
- XXX
- XXX
- XXX
- No functional writing ability

2008 R.M. Luecht
Task Models: A New Way to Blueprint

- Task models describe THREE characteristics in terms of conjunctive performance statements stated on a particular construct map:
  - Objects and their properties
  - Nature of relationships among objects and their persistence (e.g., hierarchical, directional, causal)
  - Functional clauses represent the action required on the objects and any specified conditions; e.g., $\text{Action}$(Object1, Object2) or $\text{Action}$(Object $|$ conditions)

- Cognitively complex tasks can be represented by higher-order functional clauses (e.g., “Maintains()” versus “Updates” or as nested primitive functional clauses)

- A useful task model should be capable of producing multiple templates; however, all of the templates for a given task model should be empirically shown to behave similarly in terms of their psychometric properties.
Task models aligned on the construct map replace traditional content blueprints. For example, NO MORE....

<table>
<thead>
<tr>
<th>Content Areas</th>
<th>Knowledge &amp; Concepts</th>
<th>Applications</th>
<th>Evaluation &amp; Synthesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>8%</td>
<td>10%</td>
<td>12%</td>
</tr>
<tr>
<td>B</td>
<td>6%</td>
<td>6%</td>
<td>8%</td>
</tr>
<tr>
<td>C</td>
<td>8%</td>
<td>10%</td>
<td>12%</td>
</tr>
<tr>
<td>D</td>
<td>6%</td>
<td>6%</td>
<td>8%</td>
</tr>
</tbody>
</table>
Defining and Validating Task Models

- Task models differ in *location* (difficulty) along the construct map.
- Each model provides *measurement information* in a particular region of the construct map.
- Deficits or gaps are filled by adding more task models.
- Ordering of task models must be *empirically* confirmed.
Cognitive Elements of Task Models

- **Declarative knowledge manipulatives**
  - Vocabulary/popularity of words
  - Number of objects (numeric entities, actors, concepts, or idea units) and extent of details
  - Relationships among objects
  - Relationships among properties of objects

- **Procedural-skill manipulatives**
  - Describing using objects simple recognition and recall
  - Interpretation, translation, calculations, procedures-by-rote, or identifying simple systems of relations
  - Comprehension: relating knowledge structures and predicting outcomes
  - Advanced utilization of knowledge (synthesis, evaluation, and advanced applications using complex knowledge structures)
Building Task Models that Control Difficulty and Dimensionality

- Controlling the number of key objects
- Identifying key properties of the objects relevant to the task (facilitative or distractive)
- Controlling the number of objects to be acted upon or manipulated
- Constraining the number and nature of the relationships
- Specifying and controlling the cognitive level of the action(s) or manipulation(s) required by the task
- Explicitly defining the nature and nesting of relations among objects
- Explicitly defining the nature and hierarchical sequencing of functional clauses
### Task Model Specification Worksheet

<table>
<thead>
<tr>
<th>Construct Identifier:</th>
<th>Applied statistics and educational measurement statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level(s) of Construct:</td>
<td>Basic</td>
</tr>
<tr>
<td>Primary Context:</td>
<td>Effect size, $d$</td>
</tr>
<tr>
<td>Competency Claim:</td>
<td>Computes and interprets an effect size as a standardized difference between groups or levels of an independent variable</td>
</tr>
</tbody>
</table>

#### Evidence Documentation

1.Successfully computes $d$, given two means and std. deviations from a common population.

2.Successfully computes $d$, given two means and std. deviations from independent populations (i.e., using the pooled variances).

3.Correctly interprets $d$, given two means and std. deviations from a common population.

4. Correctly interprets $d$, given two means and std. deviations from independent populations (i.e., using the pooled variances).

#### Conceptual Task Models

<table>
<thead>
<tr>
<th>Specific Tasks</th>
<th>Expected Mastery Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. <em>interprets</em> (single pop. means)</td>
<td>Plausible choice from options</td>
</tr>
<tr>
<td>2. <em>interprets</em> (separate pop. means)</td>
<td>Plausible choice from options</td>
</tr>
<tr>
<td>3. <em>interprets</em> (levels of indep. variable)</td>
<td>Plausible choice from options</td>
</tr>
<tr>
<td>4. <em>computes</em> ($\mu_1$, $\mu_2$, $\sigma_1$, $\sigma_2$)</td>
<td>Correct value</td>
</tr>
<tr>
<td>5. <em>interprets</em> (computes) ($\mu_1$, $\mu_2$, $\sigma_1$, $\sigma_2$)</td>
<td>Plausible choice from options</td>
</tr>
<tr>
<td>6. <em>interprets</em> (computes) (scatter plot)</td>
<td>Plausible choice from options</td>
</tr>
<tr>
<td>7. <em>interprets</em> (generates) (scatter plot)</td>
<td>Plausible choice from options</td>
</tr>
</tbody>
</table>

#### Manipulable Features of Complexity/Difficulty

- Magnitude of $d$ (low, moderate, high)
- Standardization of variables
- Number of groups (two or more)
- Sign of the effect size
- Formulas provided
- Software/calculator access/training
- Graphic facilitators (depictions of probability distributions)

#### Features Irrelevant to Complexity/Difficulty

- Variable labels
- Magnitude of scale
- Compute vs. interpret vs. interpret(compute())
Rules for Building Task Models

- Task models should be **incremental**—that is, ordered by **complexity**
  - Numbers knowledge objects
  - Depth of salient knowledge object properties
  - Extent of salient relations among objects
  - Sequential or simultaneous actions required to successfully complete the task

- Task models are the same level must reflect be **conjunctive performance**

- Higher performance assumes that lower level knowledge and skills have been successfully mastered
Task-Model Grammar (TMG)  
(Luecht in progress)

- **Knowledge objects** and their properties describe key task entities
  - Format: `Object.property.value="data"`
  - Drivers
    - Number of objects
    - Number of manipulated properties
    - Popularity/familiarity of the objects

- **Relational operations** link two or more objects
  - Format: `IsRelated(Object1, Object2, Nature_of_relationship)`
  - Drivers
    - Number of objects related
    - Nature of the relationship
    - Nesting of relations

- **Functional clauses** express an action or operation
  - Format: `Action(Object1, Object2)` or `Action(Object | conditions)`
  - Drivers
    - Number of arguments
    - Complexity of the function
    - Nesting of functions
Language-Based Task Design

Drivers to Consider Under TMG

Knowledge
- Unique vocabulary/TTR
- Discipline-specific vocabulary
- Grammatical structures
- Semantic relations
- Number of “idea units”
- Key properties of objects
- Nature of relations
- Graphic complexity
- Contextual constraints/setting details
- Formula familiarity
- Auxiliary language

Cognitive Skills
- Auxiliary aids
- Training/direction
- Calculation complexity
- Persistence of relations
- Mental manipulations of images and visual objects
- Derivation or manipulation of formulas
- Functional constraints on applications (e.g., open-ended functionality vs. tight scripting)
Calibrating Task Models

- The task model is treated as a family of items with similar operating characteristics.
- A hierarchical Bayesian framework can be used to estimate the task model parameters.
  - Hyperparameters are employed.
  - Uncertainty is automatically factored in.
- Scoring uses the joint probability distribution.
  - Less statistically efficient than separate item parameters.
  - More efficient in terms of operational scoring.
From Task Models to Templates

- Each task model should yield **multiple templates**
- Templates are elaborated “item models” used to **render** and **score** the items in a “family”
  - Each template has a formal data structures that captures the fixed and variable features of the task model
  - Each template “scoring evaluators” that specify how measurement opportunities are converted to “scores” such as 1=correct, 0=incorrect
- Templates must be empirically validated to ensure that they are controlling difficulty and extraneous sources of noise
Template Design and Item Writing
AE-Based Templates

- Each **task model** can be represented by **multiple, exchangeable templates**

- A **template** has three components
  - **Rendering model**: detailed presentation format data and constrained interactive components for each task (e.g., LaDuca, 1994; Case & Swanson, 1998; Luecht, 2001, 2006)
  - **Scoring evaluator**: produces item- or measurement-opportunity-level scores from a performance (Luecht, 2001, 2005, 2006)
  - **Data model**: represents the rendering model, scoring evaluator, associated difficulty drivers (radicals), and incidental surface-level manipulables in database structures that can be used/activated by item writers to generate two or more items
Item Model (LaDuca, 1994)

A 19-year old archeology student comes to the student health service complaining of severe diarrhea, with large-volume watery stools per day for 2-days. She has no vomiting, hematochezia, chills, or fever, but she is very weak and very thirsty. She just returned from a 2-week trip to a remote Central American archealogical research site. Physical examination shows a temperature of 37.2 degrees Centigrade (99.0 F), pulse 120/min., respirations 12/min., and blood pressure 90/50 mm Hg. Her lips are dry and skin turgor is poor. What is the most likely cause of her diarrhea?

A. Anxiety and stress from traveling
B. Inflammatory disease of the bowel
C. An osmotic diarrheal process
D. A secretory diarrheal process
E. Poor eating habits during her trip

A Rendering Template

“comes to” <Setting.description> “complaining of”
<Patient.ailment.symptom1>
  <Patient.ailment.symptom1.duration>
<Patient.ailment.symptom2>
  <Patient.ailment.symptom2.duration>
<Patient.history.activity.recent>
<Patient.physicalexam.temp=# C, (convert(C,F))>
<Patient.physicalexam.pulse=#/min>
<Patient.physicalexam.respiration=#/min>
<Patient.physicalexam.bp=#1/#2>
<Patient.physicalexam.symptom1>
<Patient.physicalexam.symptom2> “What is the most likely cause of <Patient.ailment.prime_symptom> “?”
A Rendering Template for Simple Statistics

A <setting.container> holds <object1.count=x> <object1.description> <object2.count=y> <object2.description>, and <object3.count=z> <object3.description>. If we select <task.action.select.object_count=k> <task.action.select.objectdescription> from <setting.container>, what is <task.response_object> that the <task.action.select.objectdescription> is <object1.description>?

<task.answer.distractor1=1/n, n=x+y+z> <task.answer.distractor2=1/{x,y, or z}> <task.answer.distractor3={x,y, or z} /{(x+y),(x+z),(y+z)}> <task.answer.correct={x,y, or z} /(x+y+z)>
Scoring Evaluators

- A **scoring evaluator** is a software or human “agent” that converts an examinee’s response(s) into a numerical score; this is conceptually similar to Wilson’s (2005) “outcome spaces”
- Single-key scoring evaluators typically resolve to a dichotomous or binary score
  - $y_{ij} = f(r_{ij}, a_i)$ for single responses
  - $y_{ij} = f(r_{ij}, a_i)$ for vectors of response variables
  - $y_{ij} \in \{0,1\}$
- Correct answer key (CAK) scoring evaluators use the “correct” answer key(s)
- Incorrect key evaluators are useful for diagnostic scoring (Luecht, 2005, 2006)
- AI-based evaluators are possible (e.g., automated essay scoring)
Data Models

- A data model is a structured representation of the salient rendering template task components and related information needed to compose, administer, and score items that are generated from a particular template.
- Plausible values to plug into the rendering template using look-up values or ranges of values.
- Constraints on use of tools or auxiliary resources (e.g., calculators, measuring devices) are specified.
- Parameters are specified for factors that directly or indirectly affect task difficulty (e.g., extent of intermediate calculations required, information density limits, etc.).
- Content, contexts, and other coded data in the task model are specified.
- Values, rubrics, or scripts used by the scoring evaluator become part of the data model.
- Automated, adaptive item construction using scripts or callable agents/routines is theoretically possible.
Possible Additional Fields in the Data Model for Capturing Task Difficulty and Complexity

- Complexity and difficulty fields based entirely on empirical statistics (e.g., $p$-values, IRT statistics, dimensionality weights, etc.)
- Complexity and difficulty fields based on item writer/test designer judgments (e.g., taxonomic “cognitive” codes)
- Template data features linked to complexity and difficulty are based on derived, replicable, cognitively relevant task models
  - A representational grammar is used to capture the salient model features
  - Data models are developed and empirically link the features to difficulty and complexity indicators
Empirically Validate the Stored Components for Each Template and Associated Task Model

- Try out prototypes to detect which components affect changes in difficulty
- Use statistical quality control (QC) analysis to identify potential sources of “error” and implement template-level controls to reduce such covariance
- Templates should account for a large proportion of explained item difficulty variance
Templates and Item Writing

- All item writing is funneled through one or more templates (i.e., item writers do NOT create their own templates)
- Component palettes can be restricted for each template
- Subtle variations in templates, component palettes, and content/context → lots of possible templates, and by extension, even more items, all with similar psychometric characteristics
Other Engineering Steps

- Create **pricing sheets** to evaluate costs of new templates and component palettes
- Use **cost-benefit analysis** to evaluate
  - The information-per-unit-of-time for costly components
  - Real costs ($$$) per unit of information
- **Maximize** the number of measurement opportunities and **minimize** the costs
- Make automated test assembly easier
  - Task models match ALL specification (demands)
  - Plenty of simulations (supply) can be generated
Psychometric QC/QA, Calibration and Scoring
Supporting Psychometrics

- Task models and/or templates can be calibrated instead of individual items, using a hierarchical Bayes framework (Glas & van der Linden, *APM*, 2003)
- Treat the hyperparameters as “super parameters” for the *task model*
- Estimate one set of common means and variance-covariances for the entire family
  - Less pretesting needed, once templates are verified
  - Fewer parameters leads to robust estimation
  - Misfit can be minimized if families are “well formed”
- Hierarchical framework is extensible as a QC mechanism
  - Minimize posterior variance associated with individual items within templates
  - Minimize posterior variance associated with templates with task models
QC via the Posterior Distributions for Task Models = P(a, b | U)

Lower Quality Task Model or Template

Higher Quality Task Model or Template
From Construct Maps to Items

Increasing Proficiency

Proficiency Claims

Can evaluate complex systems

Can analyze mid-level systems

Can prioritize key concepts

Decreasing Proficiency

Task Models

Evaluates(Ana|lyzes(Researches(system1,system2|tools)))

Analyze|ses(system1,system2|tools))

Prioritize(Select(procedures|risk,efficiency,effectiveness criteria))

Isolates(key.components)

2008 R.M. Luecht
Scoring Paradigms

- **Hierarchical Bayes**
  - Calibrated item statistics can exist at the item, template, or task-model level
  - Integrate over the joint distribution of parameters (see Glas & van der Linden, 2003)

- **Multidimensional scoring**
  - Separate ability metrics can be maintained
  - Augmented scoring can “steal” collateral information (e.g., *Test Scoring*, Wainer et al, 2001, Ch. 9) but induces a regression bias
  - Full-information MIRT scoring avoids the bias (Segall, 1996, 2000, Luecht, 1996, van der Linden, in progress, Luecht, Gierl, and Ackerman, in progress)
  - Cognitive diagnostic (constrained latent-class) models (e.g., Henson and Templin, 2006, 2007, 2008)
Traditional View of the Assessment Process

- Item Field Testing
- Blue Printing
- Practice or Domain Analysis
- Item Specifications
- Item Writing
- Item Analysis
- Operational Test Administration
- Test Assembly
- Scoring
- Equating
- Item Calibration
- Scoring Reporting & Interpret. Materials
- PLD Definitions
- Standard Setting