ABSTRACT

Title of Dissertation:	A FRAMEWORK FOR THE PRE-CALIBRATION OF AUTOMATICALLY GENERATED ITEMS.
	Shauna Jayne Sweet, Doctorate of Philosophy, 2018
Dissertation directed by:	Drs. Gregory R. Hancock and Jeffrey R. Harring, Department of Human Development and Quantitative Methodology

This paper presents a new conceptual framework and corresponding psychometric model designed for the pre-calibration of automatically generated items. This model utilizes a multi-level framework and a combination of crossed fixed and random effects to capture key components of the generative process, and is intended to be broadly applicable across research efforts and contexts. Unique among models proposed within the AIG literature, this model incorporates specific mean and variance parameters to support the direct assessment of the quality of the item generation process. The utility of this framework is demonstrated through an empirical analysis of response data collected from the online administration of automatically generated items intended to assess young students' mathematics fluency. Limitations in the application of the proposed framework are explored through targeted simulation studies, and future directions for research are discussed.

A FRAMEWORK FOR THE PRE-CALIBRATION OF AUTOMATICALLY GENERATED ITEMS

By

Shauna Jayne Sweet

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctorate of Philosophy
2018

Advisory Committee:
Dr. Gregory R. Hancock, Co-Chair
Dr. Jeffrey R. Harring, Co-Chair
Dr. Ji Seung Yang
Dr. Hong Jiao
Dr. Colleen O'Neal

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

> Technological innovation in education need not stay forever young. And one important change in the market for education technology is likely to accelerate its maturation markedly within the next several years. For the first time…states are working together… to create a new generation of assessments that will genuinely assess college and career-readiness.
>
> The development of common standards and shared assessments radically alters the market for innovation … the adoption of common standards and shared assessments means that education entrepreneurs will enjoy national markets where the best products can be taken to scale.
>
> In this new market, it will make sense for teachers in different regions to share curriculum materials and formative assessments. It will make sense for researchers to mine data to learn which materials and teaching strategies are effective for which students – and then feed that information back to students, teachers, and parents.
>
> – Joanne Weiss, 2011

As framed by Joanne Weiss, then Chief of Staff to U.S. Secretary of Education Arne Duncan, the widespread adoption of the Common Core State Standards (CCSS) was a keystone development for the future of educational assessment. The adoption of common standards promised a host of opportunities for innovation in educational technology through its creation of a national marketplace for computer-based testing (Weiss, 2011). Although the Common Core has not been the sole driver, Weiss's predictions were not incorrect. The implementation of CCSS and testing requirements mandated by No Child Left Behind, along with advances in computer technology and the advent of continuous testing through online test administration, have converged to produce a steady and rising demand for newly designed assessments aligned to the new standards (Hagopian, 2014). Driven by demand, these opportunities for innovation also present significant challenges for those responsible for developing increasingly varied and highly specialized assessments. Among the greatest challenges is how to effectively meet an ever-increasing demand for high-quality items in computer-based testing environments.

Several authors have called attention to the challenges of item development for continuous testing, particularly in high-stakes environments, where exponential increases in the number of items available are required for only linear increases in item security (see, e.g., Wainer, 2002). In computer-based testing, the cost of item writing using traditional methods is second only to expenses associated with test administration and comprises approximately 10-15% of the total budget not accounting for costs associated with pre-testing (Irvine & Kyllonen, 2002; Wainer, 2002). While the approximate cost per item necessarily depends on the item type and nature of the construct being tested, conservative estimates for development range between $1500 and $5000 per item (Irvine & Kyllonen, 2002; Rudner, 2010). Using traditional item-development methods, the cost of item bank development for use in high-takes computer-based testing is becoming prohibitive (Rudner, 2010).

With technology integration and an increased emphasis on data-driven decision-making and accountability in classrooms, there is a growing need for high-quality items for use in lower-stakes environments as well. Several providers of annual summative assessments have begun efforts to develop modules appropriate for interim or formative assessment of student proficiency. Ease of administration via mobile devices, tablet apps, or web-based platforms is part of the appeal of these products. Unfortunately, ease of use translates to low item security and higher rates of item exposure, only increasing demands on item and test development teams responsible for refreshing those materials. The growing popularity of online courses presents a similar set of opportunities and corresponding challenges. Textbook publishers are developing systems to deliver comprehensive online course support in the form of electronic textbooks and all of the resources needed to develop tailored (book-specific) assessments that can be administered during the course. With hundreds of instructors administering thousands of tests,

quizzes, and homework assignments to tens of thousands of students covering the same material, the success of these systems may be a double-edged sword for assessment developers.

Across contexts, platforms, and purposes, the current and anticipated future demand for high-quality items threatens to strain organizational capacity, timelines, and budgets. Model-based automatic item generation has been proposed as a cost-effective method for successfully populating item banks that are sufficiently large and also suitably diverse to satisfy and unprecedented demand for high-quality items that can support the construction of adaptive, customizable test forms (Arendasy & Sommer, 2007; Arendasy, Sommer, Gittler, & Hergovich, 2006; Embretson, 1999; Gierl & Haladyna, 2013; Irvine & Kyllonen, 2002).

1.1 A Paradigm Shift

As the name suggests, 'automatic item generation' (AIG) is an iterative approach to item development whereby items are constructed mechanically and, to the extent that it is technologically and practically possible, without human intervention (Bejar et al, 1993, 2002) using computer algorithms to integrate content into carefully engineered templates. Model-based item generation can be envisioned as a three-step process that begins with cognitive model development and template specification, followed by the identification of relevant content to be integrated into those templates and the definition of rules governing that integration, and finally the algorithmic integration of content into the item templates (Gierl & Lai, 2013). A visual representation of this process is shown in Figure 1, illustrating the production of multiple *templates* from *cognitive task models* that are associated with particular *educational objectives*; *essential and variable content* (and specified ranges/sets for the variable content) is specified for each template, and the algorithmic integration of content within the template, per

constraints/rules specified in the (various layers of the) *item model* yields (a very large number of) individual *instances* which are the actual tasks presented to students on an assessment.

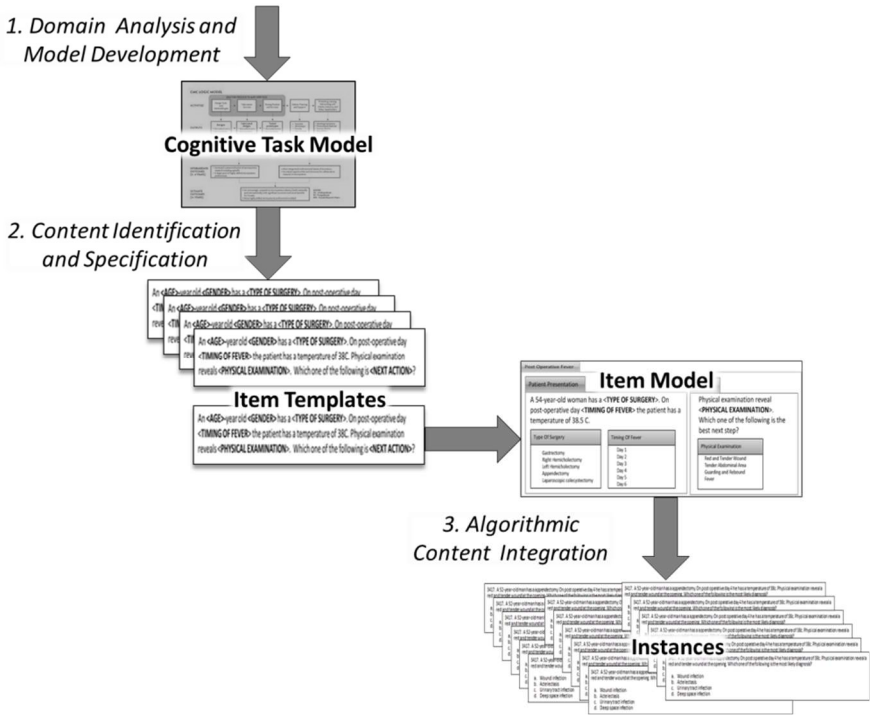

Figure 1. Three-step, multi-component AIG process

Increased processor speed and greater programming flexibility make it possible to imagine and implement fully automated item generation systems, a level of generativity that even ten years ago was viewed as exceptional (Bejar, Morely, Lawless, Bennett, & Revuelta, 2002; Ferreyra, Fabiana & Backhoff-Escudero, 2016; Gierl & Lai, 2013). But while automation of the item writing process can reduce the cost of item bank development through increased efficiency, the real promise of AIG lies in successful engineering of templates and processes of item development, not simply improved automation. Ideally, AIG processes are engineered with a precise alignment between specific features of the cognitive task models, the structural and variable elements of item templates, and the structure of the corresponding psychometric models used in calibration.

As explained by Gierl and Haladyna (2013), when AIG processes are well-designed, "templates are aligned to the task models, [and] the *items generated from the templates allow both cognitive inferences and predictable psychometric characteristics*" of generated items (Gierl & Haladyna, 2013, p. 6, emphasis added). This alignment is critical to the success of AIG and to the argument for its adoption. With this alignment, AIG is expected to drastically reduce pre-testing costs as a result of being able to predict item properties *a priori* (Irvine & Kyllonen, 2002). This is the promise of pre-calibration: that an automatic generation process is sufficiently well-designed and aligned to a corresponding psychometric model, that it is possible to estimate the parameters governing that generation process well enough to reliably predict the characteristics of items generated via that process, even if those items or templates would be the product of novel combinations of characteristics which may not yet have been directly observed.

A promised reliance on model-predicted values rather than the empirical calibration of item properties represents a paradigm shift (Gierl & Haladyna, 2013; Kuhn, 1962) and significant departure from current item writing and test development practice (Embretson & Daniel, 2008; Irvine & Kyllonen, 2002); but assessment is also rapidly changing. As Weiss (2011) predicted, along with several other critical cultural and technological factors, the "development of common standards and shared assessments [has] radically [altered] the market for innovation in… assessments." Advances in computer technology permit greater flexibility with, and capability for, on-demand item generation; and the economic and logistical realities of test development demand some level of automation in item writing. There is widespread interest in developing new items that can measure complex cognitive response processes, a desire for items that support more finely-grained inferences about examinees' knowledge, skills, and abilities, and an interest in developing items for a wide range of context including within games-

based assessments (DiCerbo, Mislevy, Behrens, O'Neil, Baker, & Perez, 2016; Hagopian, 2014; Irvine & Kyllonen, 2002). Together, these factors are powerful drivers behind continued investment and interest in the successful development and engineering of AIG systems (Gierl & Haladyna, 2012) and the specification of appropriate psychometric models for use in the pre-calibration of generated items (e.g. Cho, DeBoeck, Embretson, Rabe-Hesketh, 2014).

1.2 Current Successes and Shortcomings

There is a growing body of empirical research highlighting the successful development of AIG systems and the use of automatically generated items. Among the earliest examples of template-based item generation was a model-based system for the generation of figural matrix items (Embretson, 1999) and this work was later parlayed into the development of models for generating quantitative reasoning items (Embretson & Daniels, 2008). Building on cognitive models for language comprehension, systems have also been developed for the automated construction of multiple choice cloze items (Liu, Wang, & Gao, 2005) and vocabulary items (Brown et al., 2005) for inclusion on language proficiency tests. AIG systems have been developed for the generation and selection of items for inclusion on a general competency exam (Ferraya et al, 2016). In-depth domain analysis has also supported the development of multiple-choice items for use on medical licensure examinations (Gierl & Lai, 2014) and the development of items designed to assess young students' fluency with a range of mathematical operations (Kellogg, Rauch, Leathers, Simpson, Lines, & Bickel, 2015).

Unfortunately, despite these systems clearly demonstrating proof of concept, the adoption of AIG is remains limited because the precise alignment of cognitive and psychometric models which is critical to its success has continued to prove difficult to achieve in practice (Irvine & Kyllonen, 2002; Luecht, 2013). Even with well-defined cognitive task models and carefully

engineered item templates, additional review of and identification of constraints on combinations of content, beyond what is suggested by the cognitive task model, is often necessary. Gorin (2005) identified as the most significant challenge for implementing AIG "the development and verification of a viable cognitive model and an associated task feature model… [that] contains features that realistically can be manipulated to affect processing in such a way that item difficulty is reliably predicted" (p. 351). Full automation may be technologically feasible but the consistent generation of high-quality items still requires that human intervention be part of the generation process (Arendasy & Sommer, 2007; Embretson, 1999; Irvine & Kyllonen, 2002; Liu et al., 2005). Improvements in technology have not resolved the challenges inherent in defining the set and range of range of feature manipulations that will have well-understood impacts on item properties (Irvine & Kyllonen, 2002; Luecht, 2013). Perhaps most importantly, researchers have yet to identify a modeling framework suitable for pre-calibration such that "item generation and psychometric modeling are completely intertwined in such a way that it becomes possible to not only generate items but also 'parse' any item to characterize its psychometric properties" (Bejar et al, 2002, p. 202).

1.3 A Need to Close the Gap

Difficulties in consistently predicting properties of generated items highlight the need for research into both appropriate model specification and detection procedures for items that perform contrary to expectation. In applied settings, there are only a few cases in which item characteristics have been successfully predicted using item or template features; and even then, success has been only moderate and heavily dependent on domain (Arendasy & Sommer, 2007; Embretson & Daniel, 2008; Gorin, 2005; Irvine & Kyllonen, 2002; Liu et al., 2005).

AIG research efforts into model specification for use in pre-calibration feature psychometric models with increasingly complex item and family mean structures designed to capture as completely as possible the nuances of cognitive response processes and wider arrays of item features (e.g., Cho et al., 2013; Cho et al., 2014). But even as newly developed integrated modeling frameworks (Cho et al., 2013; Geerlings et al., 2011) are promising for use in AIG contexts, there is a continued need to examine the performance of these models under conditions that more closely resemble those encountered by applied researchers and test developers.

But alongside the development of model specifications and algorithms capable of estimating these more complex models, there has not been an investigation of the impact or the detection of items that do not perform according to model expectations. Psychometric models intended for use as pre-calibration models in the context of AIG are first and foremost confirmatory models, and there is a lack of simulation research in the context of AIG that investigates issues related to model misspecification. There is a pressing need to understand the effects, on item parameter estimates and inferences about examinee proficiency, when generated items are not 'well-behaved' (Luecht, 2012). In fact, among the authors who have worked to detail principles of AIG item development, there are several for whom issues of model selection for item pre-calibration are 'out of scope' when addressing questions of implementation (e.g., Alves, Gierl, & Lai, 2010; Huff, Alves, & Pellegrino, 2013).

Especially the interest in innovative item types increases, the desire for items that are more specifically targeted and items which elicit evidence of complex skills becomes more widespread, and requirements for item generation processes necessarily become more complex, there is a growing need for research into the specification and performance of psychometric models for use in pre-calibration. This paper seeks to fill this gap in AIG research, presenting a

new and broadly applicable conceptual framework and corresponding psychometric model designed for the pre-calibration of automatically generated items. Unique among models proposed within the AIG literature, this model incorporates specific mean and variance parameters to support the direct assessment of the quality of the item generation process. The utility of this framework is demonstrated through an empirical analysis of response data collected from the online administration of automatically generated items intended to assess young students' mathematics fluency. Recognizing the importance of understanding the impacts on model parameter estimates of poor or incomplete model specification, and the need for more of this work within the AIG literature, targeted simulation studies explore possible limitations in the application of the proposed framework and the interpretation of parameter estimates.

1.4 Overview

Chapter 2 provides an overview of the statistical frameworks that have been proposed for use in the AIG context to pre-calibrate items generated through a mechanical template-based process. There is an array of models that have been proposed and a diversity of perspectives that inform this research, but there is also an unfortunate lack of coherence in the AIG literature. Chapter 3 seeks to inform the AIG literature by proposing a coherent conceptual framework that is sufficiently flexible to accommodate a range of approaches to automation and item calibration but clear in providing researchers with a common vocabulary for item generation and evaluation. Also in Chapter 3, a new mathematical modeling framework is proposed for use in the calibration pre-calibration of automatically generated items and its relationship to and extensions beyond previously proposed frameworks are highlighted. Chapter 4 provides a demonstration of the utility of the proposed framework through an analysis of item response data collected from the online administration of algorithmically generated items designed to assess elementary

school students' computational fluency. Results from a targeted simulation study, with conditions designed to map onto a range of real as opposed to ideal implementation conditions, aid in the interpretation of parameter estimates. Chapter 5 presents additional simulation work which examines the impact of model misspecification on item parameter and ability estimates. Chapter 6 considers key take-aways from the work presented in the previous chapters and outlines future directions for research into the possibility of successful pre-calibration.

Chapter 2: Approaches to Pre-Calibration

In 2002, ETS researchers were engaged in a multi-pronged research program to "generate many assessment tasks efficiently and effectively… to automatically generate [*pre-*]*calibrated* items so that costs can be reduced and validation is built into test development. Items are generated from templates that describe a content class. Each template contains both fixed and variable elements. The variable elements can be numeric or linguistic. Replacing the template's variables with values results in a new item" (Gitomer & Bennett, 2002, p. 9).

In the fifteen years following the publication of Gitomer and Bennett's technical report outlining in the promise of a system for generating pre-calibrated items, advances in computing made these systems possible to implement in operational and not just research contexts (e.g. Gierl & Lai, 2014; Kellogg, Rauch, Leathers, Simpson, Lines, & Bickel, 2015). In fact, a number of AIG systems have been successfully designed to support the algorithmic production of items across a wide range of domains, including figural reasoning (Embretson, 1998, 1999), reading comprehension (Gorin, 2005), sentence completion (Sheehan & Mislevy, 2002), algebra and quantitative reasoning (Arendasy et al, 2006; Arendasy & Sommer, 2007; Embretson & Daniel, 2008), and K-12 mathematics (Simpson, Elmore, Bickel, & Price, 2015). Algorithmic item generation procedures have also been used to create items for inclusion on exams for medical and dental licensure (Gierl & Lai, 2012, 2013a, 2014) and for testing second language proficiency (Liu, Wang, & Gao, 2005).

Despite demonstrated success in architecting item generation systems, researchers have continued to struggle with the challenge of pre-calibrating items, and this remains a barrier to widespread adoption of AIG. Pre-calibration remains a barrier because even though it is a difficult problem to solve, within the AIG context finding a solution is necessary. Even with a

limited number of templates and relatively few manipulated features, item generators have readily produced thousands (Simpson et al., 2015) and even tens of thousands of items (Gierl & Lai, 2014). The algorithmic construction of items is absolutely possible, but calibrating all of those items directly is not. Pre-calibration is necessary to take full advantage of all that AIG has to offer.

2.1 The Challenge of Pre-Calibration

Automatic item generation is a template-based process whereby a pre-defined range of content can be algorithmically integrated into a generic item form or shell in order to create a set of unique items aligned to a common educational objective (Gierl & Haladyna, 2013). Pre-calibration is the process by which the parameters governing the generation process are estimated well enough to reliably predict the characteristics of items generated via that process, even if those items had not been seen by any examinees. Pre-calibration is really the calibration of higher-order design features within the generation process.

Within the AIG literature, the challenge of pre-calibration is viewed first and foremost as a challenge of engineering: how to develop templates to effectively structure items, and how to isolate, specify, and combine variable features within those templates in such a way that the impact of those manipulations is well understood. Given a well-engineered item generation process, the challenge is then one of model specification and: how to parameterize a psychometric model to ensure sufficient alignment to the generation process to support the prediction of the properties of generated items. Interestingly, AIG researchers frequently assume that the generation process is well-engineered and focus their attention on model specification.

This chapter provides an overview of the psychometric models that have been proposed for use by AIG researchers. Within the broader psychometric literature, it is well-understood that

generating items from common templates induces local dependencies that need to be accounted for in order to ensure accurate and precise parameter estimates (Chen & Wang, 2007; Cohen et al., 2008; Jiao et al., 2005; Jiao et al., 2008). All of the calibration models that have been proposed for use within the context of AIG are therefore aligned to some version of a template-based generation process, and typically utilize multi-level structures (e.g. Cho et al., 2014; Gierl & Lai, 2012; Kellogg et al., 2015) or otherwise include parameters intended to account for variance across and dependencies among items generated from the same templates (e.g., Embretson & Daniel, 2008). The models reviewed in this chapter are organized according to their features, and specifically the use of either fixed or random effects or both to capture the relevant features of the generation process. Models are also presented roughly in chronological order.

The first models discussed in this chapter exclusively feature fixed effects and were also the first to appear in the AIG literature: researchers sought to identify the set of design manipulations which would determine item properties and calibrate those features. Random effects models appeared later in the AIG literature, and these models typically use multilevel structures to group similar items together as "item families" and support the calibration of prototypical instantiations. The third set of models discussed are those which have appeared most recently in the literature and blend the first two approaches, accounting for differences between items in terms of cognitively relevant design features while also accounting for dependence among items generated from common templates. A review of these models provides the necessary background for the presentation of a new conceptual and statistical framework for calibrating automatically generated items which will be presented in the third chapter.

2.2 Fixed Effects Models: Calibrating Feature Manipulations

The first type of model accounts for inter-item dependencies resulting from the influence of design decisions on the generation of items that are assumed to be directly related to the cognitive skills being tested (Embretson, 1998; 1999; Embretson & Daniel, 2008; Geerlings, Glas, and van der Linden, 2011). Characterized as *cognitive-psychometric models,* these fixed-effects models calibrate common design principles rather than modeling the psychometric properties of individual items, (Embretson, 1999). These models featured heavily in ETS research programs in the late 1990s through the middle of the following decade, where the goal was to develop principled item design approaches to automatic item generation and to determine the feasibility of estimating the impact(s) of design decisions in order to reliably predict the difficulty of items based on a subset of cognitively relevant item features , thereby reducing the need to collect calibration data in future test administrations (e.g. Gitomer & Bennett, 2002). The success of this approach to item generation and pre-calibration hinged on researchers; ability to identify cognitive response processes and link them to "observable features of [items] that can be systematically coded and entered into statistical analyses to test the impact of the [proposed response] process on item difficulty" (Gorin, 2006, p. 24). By explicitly linking item response probabilities to design principles or item features that were hypothesized to impact the processing requirements of items, these models also could provide item-level evidence of construct validity (Embretson & Daniel, 2008; Gorin, 2006).

2.2.1 Fixed Effects Models Proposed for Pre-Calibration

The linear logistic test model (LLTM; Fischer, 1973) was proposed for use in this context. The LLTM is an extension of the Rasch model (Rasch, 1960) which decomposes item difficulty into a weighted linear combination of j attributes. Where X_{ip} is a response

dichotomously scored as 0 or 1, the Rasch model represents the probability of an individual *p* with ability θ_p correctly answering item *i* using the following functional form:

$$Pr(X_{ip} = 1|\theta_p, \alpha, \beta_i) = \frac{e^{\alpha(\theta_p - \beta_i)}}{1+e^{\alpha(\theta_p - \beta_i)}}. \quad (2.1)$$

The item-specific difficulty is represented by β_i and α is the common discrimination parameter for all of the items. The LLTM model builds on the Rasch model by parameterizing item difficulty as a linear combination of *J* item attributes $q_{i1}, \ \ldots, \ q_{ij}$, where π_j represents the effect of attribute *j* on the difficulty of item *i:*

$$\beta_i = \pi_1 q_{i1} + \pi_2 q_{i2} + \cdots + \ \eta_J q_{iJ} = \sum_{j=1}^{J} \pi_j q_{ij}$$

The log odds of a correct response can therefore be written as:

$$\eta_{ip} = \theta_p - \sum_{j=1}^{J} \pi_j q_{ij}, \text{ where } \theta_p \sim N(0,1) \text{ to ensure identifiability.}$$

In the context of pre-calibrating automatically generated items, the goal is to collect an initial set of item responses to items featuring as many relevant combinations of *j* attributes as possible in order to estimate the effects, π_j, of those attributes in order to be able to predict the difficulty of future items based on their design alone.

Gorin (2005) effectively used an LLTM model to investigate the extent to which the difficulty reading comprehension items could be manipulated by varying items' (1) propositional density and syntax, (2) the presence of negative or passive voice, (3) the order of information, or (4) response alternatives. There are, however, many cases in which a Rasch model is not sufficient to model item response functions. In her work on the generation and calibration of figural matrix items, Embretson (1999) found that item features were predictive of both item discrimination and difficulty. She proposed the constrained two-parameter logistic (C2PL) as an extension of the linear logistic test model, in which she related item design features to both item

discrimination and difficulty. The two-parameter logistic (2PL) model (Lord & Novick, 1968) has the following item response function:

$$Pr(X_{ip} = 1|\theta_p, \alpha_i, \beta_i) = \frac{e^{\alpha_i(\theta_p - \beta_i)}}{1+e^{\alpha_i(\theta_p - \beta_i)}} \quad (2.2)$$

where X_{ip} is again a response to item *i* by individual *p* that is dichotomously scored as 0 or 1, β_i is the item-difficulty and α_i is the item-specific discrimination or slope parameter of the item response function. In the C2PL model, both item discrimination and difficulty are specified as a to be a linear combination of *J* common design features, such that for item *i,*

$$\alpha_i = \delta_1 q_{i1} + \delta_2 q_{i2} + \cdots + \delta_J q_{iJ} = \sum_{j=1}^{J} \delta_j q_{ij}$$

$$\beta_i = \pi_1 q_{i1} + \pi_2 q_{i2} + \cdots + \pi_J q_{iJ} = \sum_{j=1}^{J} \pi_j q_{ij}$$

The log odds of a correct response can therefore be written as:

$$\eta_{ip} = \sum_{j=1}^{J} \delta_j q_{ij} \cdot (\theta_p - \sum_{j=1}^{J} \pi_j q_{ij}), \text{ where } \theta_p \sim N(0,1). \quad (2.3)$$

In her analysis of data from figural matrix items, Embretson coded as design features the number of rules incorporated in the design of each item, the abstract correspondence, and the overlay, fusion, and distortion of figures (Embretson, 1998; 1999). Embretson successfully applied the C2PL model to these data, and demonstrated that this model offered a better data-model fit than the LLTM. Results were consistent with theory that the number of rules governing the figural matrix patterns and the complexity of those rules would affect examinees' ability to infer those rules and also apply them correctly.

2.2.2 Limitations of Fixed Effects Models

Empirical results support rule-based item generation as a promising approach to item generation for use in operational contexts (e.g. Morley, Bridgeman, & Lawless, 2004; Embretson & Daniel, 2008; Geerlings, Glas, & van der Linden, 2011), but there is only limited support for utilizing fixed effects models for the pre-calibration of resulting items, even when using well-designed templates (De Boeck, 2008). As illustrated in Embretson's work (e.g. Embretson & Daniel, 2008), which relied on in-depth feature coding by a panel of experts, the accurate calibration design features using fixed effects models may come at too high a cost both in terms of money and time. This cost is increasingly prohibitive given anticipated operational demands for wider arrays of items and items that are more specifically targeted to educational objectives (Gierl & Lai, 2013). Moreover, rule-based item generation is not necessarily feasible in all situations. Template-based item generation and the specification of cognitive models which support the specification of LLTM and C2PL models is limited to application in "narrow domains where cognitive analysis is feasible and where well-developed theory is more likely to exist" (Bejar et al, 2002, p. 5,). Unfortunately, well-defined domains may be more of the exception than the rule (Rupp, diCerbo, Levy, Benson, Sweet, Crawford, Fay, Kunze, Calico, & Behrens, 2012), and even in those well-defined domains "experts sometimes have blind spots regarding the cognitive processes used by the respondents to solve the given tasks" (Arendasy & Sommer, 2007, p. 380).

2.3 Random Effects Models: Calibrating Prototypical Items

As an alternative to the highly structured rule-based approach to item generation, Bejar and colleagues (2002) advocated for a template-based approach. Item developers would create prototypical instances (also known as *parent items*) and from these prototypical instances, derive

an array of items which look sufficiently different from the parent item and from one another to prevent transfer of solution strategies but whose essential characteristics and psychometric properties are unchanged. Items modeled after the same parent item are referred to as *siblings*, with each parent item and its siblings comprising an item f*amily*.

Given the importance of aligning the item generation process and the psychometric model used for pre-calibration, a template-based approach to item generation necessarily warrants a different approach to item pre-calibration. This second category of psychometric models proposed for use in AIG contexts facilitate pre-calibration of items via the calibration of prototypical items. Unlike the fixed effects models discussed previously, the random effects model structure is not designed to estimate the impact(s) of specific design decisions. Items derived from the same prototypical instance are designed (and are subsequently assumed) to have similar psychometric properties. As such, family-level parameters can be estimated from responses to instances randomly sampled from within each family. Using a hierarchical model structure, these models estimate the characteristics of families of related items and incorporate random effects to account for the dependence within and (limited) variation among items within the same family (Bejar et al, 2002; Sinharay & Johnson, 2005; Sinharay, Johnson, and Williams, 2003; Geerlings et al, 2011; Geerlings, 2012; Gierl & Haladyna, 2013).

2.3.1 Random Effects Models Proposed for Pre-Calibration

One approach to estimating the variance in item responses due to family membership is to utilize the linear logistic test model with error (LLTM-R; De Boeck, 2008; Janssen et al, 2004). The LLTM-R is an extension of the LLTM that permits imperfect prediction of item difficulty by item features. For use in the context of calibrating item families, the LLTM-R can be formulated using dummy variables to represent family membership. Building on the LLTM

presented earlier, the log odds of an individual *p* responding correctly to item *i* is represented by the equation:

$$\eta_{ip} = \theta_p - \sum_{j=1}^{J} \pi_j q_{ij} + \varepsilon_i$$

Within this framework, persons and items are modeled as crossed-random effects (De Boeck, 2008), and latent abilities are again typically assumed to be distributed $\theta_j \sim N(0,1)$ to ensure the model is identified and the measurement error term specified as $\varepsilon_i \sim N(0, \sigma_\varepsilon)$. Unfortunately, although this approach was suggested by Embretson & Daniel (2008), it doesn't appear in any studies calibrating item families and as such it is unclear how well this approach might work for the pre-calibration of automatically generated items.

Another approach to pre-calibrating items generated using a template-based process is to use a two-level random-effects model. Alternately referred to as the Related Siblings Model (RSM; e.g. Sinharay & Johnson, 2005) or the Item Cloning Model (ICM; Glas & van der Linden, 2003), this approach uses random effects to model an association structure among the items within an item family. The first-level model of the RSM or the ICM is an IRT model, such as the Rasch model (Equation 2.1). At Level 1, the log odds of a correct answer to item *i* within family *j* is written as follows:

$$\eta_{ip} = \theta_p - \beta_{ij}$$

Within each item family the effects of changing the incidental or surface features of items are assumed to be minor and unsystematic. This is reflected in the Level 2 specification of the RSM, where the difficulty of generated items is specified as the family mean (the difficulty of the prototypical item and some random error:

$$\beta_{ij} = \pi_{0j} + \varepsilon_{ij}$$

The odds of a correct response to item *i* within family *j* can therefore be written as:

$$\eta_{ipj} = \theta_p - (\pi_{0j} + \varepsilon_{ij}), \text{ where } \theta_p \sim N(0,1) \text{ and } \theta_\varepsilon \sim N(0, \sigma_\varepsilon) \quad (2.4)$$

Again, both persons and items are treated as random. For the purposes of model identification, the population distribution for the latent abilities is typically assumed to be normally distributed such that $\theta_p \sim N(0,1)$.Using this framework, family-level parameters can be estimated from responses to instances randomly sampled from within each family, and the family-level parameters are used to predict the characteristics of generated items.

2.3.2 Limitations of Random Effects Models

The primary limitation of random effects models is that they lack explanatory power. Although pre-calibration of generated items is possible using hierarchical models, the models provide no insight into the quality of the item generation process. There are no parameters which reflect the relative success of design features or a need for their improvement.

A second limitation these models is that they are relatively untested within the AIG context. Unlike the fixed effects models, which were proposed in the context of real data analysis and tested through empirical research, the work on random effects models draws primarily on simulation work. Unfortunately, in the simulation-based work that has informed the development of the more complex random-effects models (e.g., Cho et al., 2013), authors rarely explore issues of poor model-data fit. By and large, built into the design of model simulations is the assumption that items have been consistently successfully generated, meaning that the items behave as expected according to the theoretical model (Luecht, 2013).

A few AIG simulation studies have investigated model performance when the generation process is "unsuccessful" (Leucht, 2013) and produce instances with high variability (e.g. Bejar et al., 2003; Sinharay & Johnson, 2005). Results from those studies suggested that non-

isomorphicity within families has minimal impact on item parameter and ability estimates, provided variability is appropriately accounted for (Glas & van der Linden, 2003; Sinharay, Johnson, & Williamson, 2003; Sinharay & Johnson, 2005; Leucht, 2013), but thresholds for levels of within-family variation are not well-defined in the AIG literature and often hover near zero. There is a need to examine the performance of these models under more realistic conditions.

2.4 Modeling Differences and Similarities with Integrated Frameworks

Evolving technologies are increasingly capable of more diverse item generation (Gierl & Lai, 2013a, 2013b, 2014). Increasingly nuanced understandings of cognitive response processes and the growing desire for more finely-grained inferences about examinees' knowledge, skills, and abilities, have yielded increasingly complex task models for item development. It is therefore unsurprising that the models being proposed for use in AIG contexts feature increasingly complex item and family means structures in an effort to align as completely as possible with emerging AIG processes (Cho et al., 2013; Geerlings et al., 2011; Liu, Wang, & Gao, 2005). These models feature combinations of fixed feature combinations of fixed and random effects in order to account both for specific design features and variability among items generated from the same templates.

2.4.1 Linear Item Cloning Model

Geerlings and colleagues (2011) developed the linear item cloning model (LICM) to be applied when item developers utilize "a combination of the two methods of automated item generation" (p. 337). This model aligns to a template-based approach to item generation where prototypical items are generated via the application and combination of a set of features that are

intended to impact item properties. For each combination of design rules there is a family of items which are generated from the prototypical or parent item through minor changes to non-essential surface features. A graphical representation of this item generation process is shown below in Figure 2.

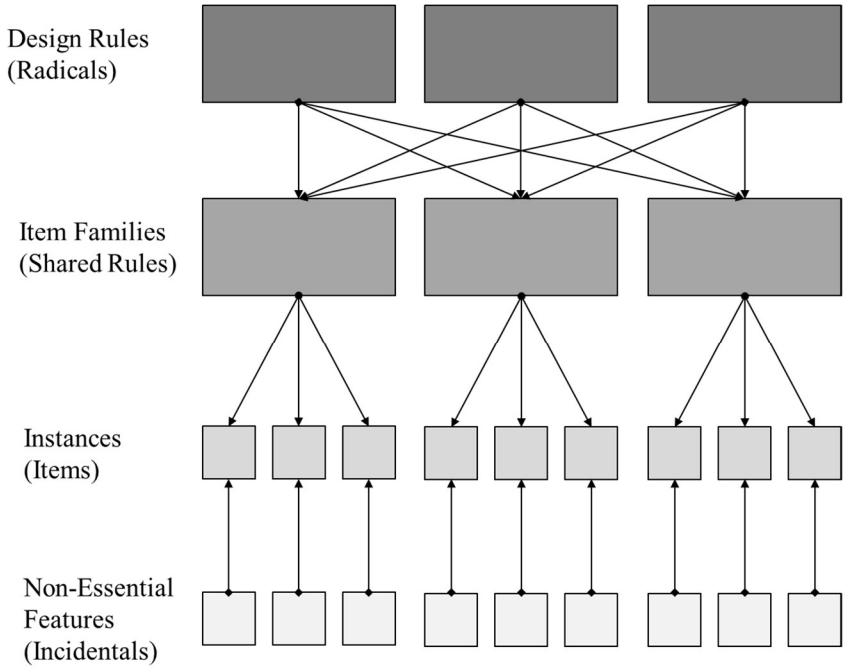

Figure 2. Item generation process for LICM model

The LICM is a two-level model which extends the work of Sinharay and Johnson (2008) and the development of the RSM (Equation 2.4) by placing structure on the mean difficulty of each item family at Level 2. Random effects at Level 1 account for dependencies in response probabilities among instances within item families generated using common sets of manipulated features.

The LICM utilizes the to specify the probability of a correct response to item *i*. Although a 2PL or 3PL model could be used for the first-level model, Geerlings et al. (2011) presented the LICM using the three-parameter normal-ogive (3PNO) model at Level 1, requiring a number of

assumptions to be made about the independence of residuals, as discussed in the previous section. For simplicity of exposition here we assume a Rasch model as the item response model, and the odds of a correct response to item *i* within family *j* can therefore be written as:

$$\eta_{ipj} = \theta_p - (\sum_{k=1}^{K} \pi_{00k} q_{0jk} + \varepsilon_{ij}), \text{ where } \theta_p \sim N(0,1) \text{ and } \varepsilon_{ij} \sim N(0, \sigma_{\varepsilon_j}) \quad (2.5)$$

where π_{00k} denotes the effect of design feature k on the difficulty of item families, and q_{0jk} is a design variable which captures the use of that feature in the generation of the parent item for family *j*. It is worth noting that in the specification of the LICM, Geerlings et al. (2011) departed from most other AIG researchers by exploring alternative methods for encoding design variables. Features are typically encoded as binary (present or not present), but Geerlings and colleagues demonstrated the possibility of including a range of values to communicate either the extent to which a feature was manipulated or the number of times a rule was applied in the generation process. The effects of these design features are consistent across families; within-family variation can be estimated as a common or as a family-specific variance parameter as shown in the notation for Equation 2.5.

2.4.2 Additive Multilevel Item Structure Model

Cho and colleagues (2014) proposed a multi-level mixed effects IRT model which aligns more closely to a rule-based approach to item generation than the one explored by Geerlings et al. (2011). A graphical depiction of the additive multilevel item structure (AMIS) model and the logic of the corresponding generation process is shown in Figure 3. Like the LICM, prototypical items are generated via the application and combination of a set of features that are intended to impact item properties. Unlike the LICM, within item categories that are determined by common design features, items are generated as variants of one another instead of being engineered as siblings. This difference also accounts for the differences in terminology used by the authors to

describe groupings of items at Level 2, categories (Cho et al, 2014) versus families (Geerlings et al, 2011).

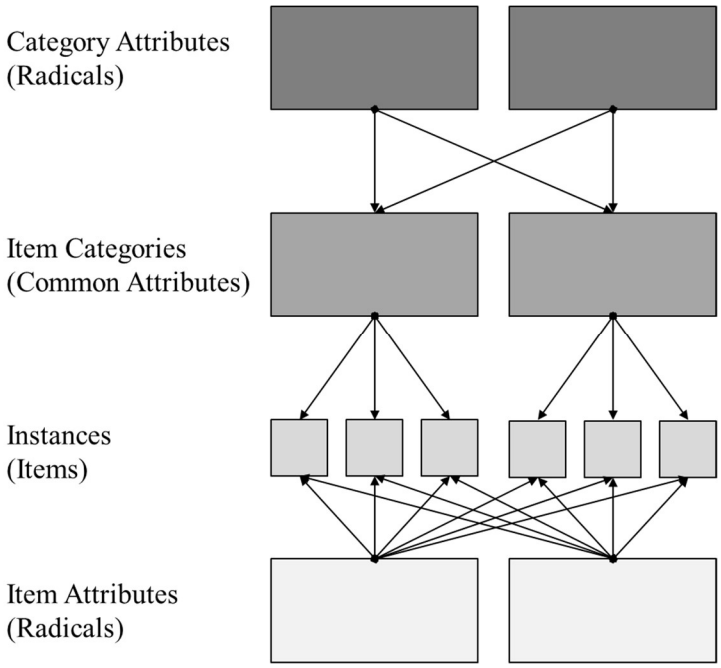

Figure 3. Item generation process for AMIS model

The AMIS model decomposes the mean family-wise discrimination and difficulty parameters (in the second-level model) as a weighted combination of effects and also models systematic variation in within-family discrimination and difficulty parameters (in the first-level model). The discrimination and difficulty parameters have a common structure: each is decomposed into a linear combination of an overall mean, the weighted sum of *d* item-specific attributes and the weighted sum of *t* category-specific attributes, a category-specific residual and an item-specific residual.

Cho et al. (2013) assume that a 2PL model (see Equation 2.2) is appropriate for modeling the item response function and present the full equation for the logit of person *p* to respond correctly to item *i*, which generated as a member of category *c,* as follows:

$$\eta_{ipc} = \left(\mu_\alpha + \sum_d \gamma_{\alpha d} Q_{id} + \varepsilon_{\alpha i}^{(1)} + \sum_t \delta_{\alpha t} R_{ct} + \varepsilon_{\alpha c}^{(2)}\right) \cdot \left(\theta_p - \left(\mu_\beta + \sum_d \gamma_{\beta d} Q_{id} + \varepsilon_{\beta i}^{(1)} + \sum_t \delta_{\beta t} R_{ct} + \varepsilon_{\beta c}^{(2)}\right)\right).$$

The population distribution for the latent abilities is assumed to be normally distributed such that $\theta_p \sim N(0,1)$. Cho and colleagues (2013) also assume that within- and between-family variances are equal, and the assumption is also made that residuals are independent. Specifically,

$$\varepsilon_{\alpha i} \sim N(0, \sigma_\alpha) \text{ and } \varepsilon_{\alpha c} \sim N(0, \sigma_\alpha)$$

$$\varepsilon_{\beta i} \sim N(0, \sigma_\beta) \text{ and } \varepsilon_{\beta c} \sim N(0, \sigma_\beta)$$

It is interesting to note that despite being far more complex in terms of its notation, the AMIS model has striking similarities to the C2PL model (Equation 2.3): the same set of attributes impacts both discrimination and difficulty, and the effects of those attributes are homogeneous across items and across families. While it is true that the integrated modeling frameworks show considerable promise, at least in their initial application of the AMIS model, Cho and colleagues (2013) do not attempt to exercise the flexibility afforded by the multilevel modeling structure. The ubiquity of simplifying assumptions in the AIG literature, despite models which have increasingly complex means structures hints at some of the possible limitations of these integrated frameworks.

2.4.3 Limitations of Integrated Modeling Frameworks

Estimation. Although the integrated modeling frameworks promise better alignment to the underlying item generation processes, increased model complexity does present additional challenges for estimation. Estimating fixed effects models for use in AIG contexts is relatively simple. For the initial calibration of the LLTM, design effects are typically estimated using marginal maximum likelihood estimates (Embretson & Daniel, 2008); and Embretson (1999) used a joint maximum likelihood approach was used to estimate design effects specified using the C2PL framework. Researchers who have proposed the use of random effects models have used both maximum likelihood and Bayesian approaches to the estimation of item parameters for

the initial calibration of item families. Glas & van der Linden (2003) used Markov Chain Monte Carlo (MCMC) methods and also marginal maximum likelihood techniques (Glas & van der Linden, 2003; Geerlings, Glas, & van der Linden, 2011) to estimate item family means. Sinharay and colleagues (2008; 2013) also used EAP estimates to describe the mean behavior of items within a family. In contrast, researchers who have proposed the more complex models have also noted challenges in estimation and mentioned some workarounds. In their discussion of the LICM, Geerlings et al. (2011) proposed the three-parameter normal-ogive (3PNO) model instead of using a 3PL model to facilitate sampling from the conditional posterior distributions of family-wise parameters when estimating abilities. Citing the computational burden of MCMC estimation approaches and the possibility of slow convergence as a result of possible correlations in joint posterior distributions of estimated parameters, Cho and colleagues describe an extended alternating imputation-posterior algorithm (AIP) with adaptive quadrature to estimate item and person parameters (Cho & Rabe-Hesketh, 2011; Cho et al., 2013). It is true that there is ongoing investment in MCMC estimation hierarchical IRT models (Stan Development Team, 2017), but as part of any research effort, attention needs to be paid to the conditions under which these more complex models can be estimated.

Assumptions. The need for simplifying assumptions may be driven in part by concerns about the successful estimation of model parameters (e.g. Cho et al, 2013). However, as models proposed for item pre-calibration become more complex and more assumptions need to be made in order to meet the requirements of the model or to facilitate estimation, it is important to pay attention to the implication those assumptions have for the quality of the underlying generation process and how well that process needs to be understood in order to warrant those assumptions.

Simulation studies designed to demonstrate the promise of hierarchical models for use within an AIG context, with few exceptions (notably Geerlings, 2012; Geerlings et al., 2011) routinely set residual covariances equal to zero, with minimal variances on the diagonals that govern the variability of instances within item families. This is consistent with assumptions routinely made about homoscedasticity and the independence of residuals in the presentation or application of models proposed for use in the pre-calibration of AIG items. Many authors examining the performance of hierarchical models for use in an AIG context note that while the proposed models can arguably accommodate non-zero covariances between parameters at level one (Embretson & Daniel, 2008) or at level two (Geerlings et al., 2011), they routinely make simplifying assumptions about covariances between item parameters within and across families. As noted in the previous section, Additionally, although the AMIS model can be extended to account for heteroscedastic residuals and a bivariate distribution for the residuals of item discrimination and difficulty, Cho and colleagues (2013) assume that within- and between-family variances are equal, and the assumption is also made that residuals are independent. But this independence also assumes that the generation process is well understood and all of the relevant features are identified, which flies in the face of the lessons learned through empirical research into the performance of fixed effects models for pre-calibration (e.g. Gorin, 2005).

In a small simulation study Luecht (2013) clearly highlighted, within a limited range of conditions, the potential impact of unmodeled residual covariances between item parameters within families using a 2PL model. Luecht did not specify the origin of this covariance between parameters otherwise unaccounted for in the model, but in a limited simulation study in which he varied test length (10 versus 40 items), estimation error of item parameters resulting from family-level calibration (none using the generating parameters, low, moderate, and high), and

conditional covariance between discrimination and difficulty parameters (low, moderate, high), he demonstrated increased error and bias in ability estimates. Although increasing test length was shown to ameliorate the effects of the loss of efficiency resulting from family-level calibration in the absence of residual (level-1) covariance, simply increasing test length failed to address the bias that resulted from the presence of even low residual covariances when there was a high degree of within-family variability (Luecht, 2013).

Little attention has been paid to the modeling or accurate estimation of covariances between item parameters, or to the possible impacts of misspecification. As researchers look to develop and test the feasibility of increasingly complex models for use in AIG contexts, this needs to be kept in mind.

2.4.4 Looking Ahead

On their face, these integrated modeling framework with sets of both fixed and random effects appear to better capture the complexity of item generation processes and as such offer the most promising approach to pre-calibration of automatically generated items (e.g. Cho et al., 2013; 2014; Geerlings, Glas, & van der Linden, 2011). However, this integration of modeling frameworks needs to go beyond the inclusion of more complex means structures to include the establishment of a common conceptual frame that will support continued model development and evaluation, along with a re-examination of some the fundamental model assumptions that arguably limit the applicability and utility of these otherwise promising models.

Chapter 3: A New Framework for Pre-Calibration

The AIG literature is punctuated with numerous efforts to specify psychometric models appropriate for the pre-calibration of generated items (see Gierl & Haladyna, 2013; Irvine & Kyllonen, 2002), but the literature provides little evidence of forward momentum despite continued and growing interest in solving the problem of successful pre-calibration. A thorough review of the AIG literature suggests that the absence of a common language and conceptual framework to support current (and future) modeling efforts is likely a contributing factor.

Automatic item generation is fundamentally an engineering problem (Gierl & Lai, 2012), and pre-calibration is first and foremost one of design. How the various components of the item generation process are related to or integrated with one another to produce individual test items necessarily informs the pre-calibration model's structure and the resulting parameter interpretations in the AIG context. For that reason, attention must be paid to how the item generation process is envisioned and the terminology used to describe it, because these are the building blocks of a conceptual framework and vocabulary for specifying an appropriate psychometric model and also for its evaluation. Unfortunately, although there is a common logic motivating the structure and specification of the models which appear in the AIG literature, there is not yet a coherent conceptual framework that spans research efforts.

AIG researchers have converged on the problem of item generation from a range of different disciplines, and they often use subtly (and not-so-subtly) different words to describe the item generation process and to define corresponding statistical models, using terminology borrowed from cognitive psychology (e.g., Embretson & Daniel, 2008; Gorin, 2005), assessment engineering (e.g., Arendasy & Sommer, 2007), and elsewhere. Researchers' vocabularies arguably reflects their background and training rather than being native to the AIG context.

Difficulties stemming from inconsistent language are compounded by vagueness in model definitions. Constructs that are central to model specification, namely *radicals*, *incidentals*, and any mention of *parents*, *siblings*, or *item families* are not consistently defined vis a vis the item generation process and definitions of these constructs are often incomplete (M. Gierl, personal communication, December, 2015; Gierl & Lai, 2013a; Sinharay & Johnson, 2013). For AIG researchers, effective synthesis of findings first requires translation. Competing models are consequently difficult to evaluate, compare, and apply to contexts other than those for which they were immediately developed, and even when estimation is possible, the resulting parameters are difficult to interpret (Alves, Gierl, & Lai, 2010; Cho et al., 2014; Gierl & Lai, 2013a, 2013b; Huff, Alves, & Pellegrino, 2013). It is difficult to say, across research efforts, exactly what is working and what might be emerging as best practices for pre-calibration.

In an effort to address a persistent gap in the AIG literature, this chapter presents a conceptual framework and corresponding psychometric model that will support the pre-calibration automatically generated items. The proposed framework looks to provide a clear vocabulary for describing the item generation process and also a roadmap for specifying an appropriate pre-calibration model.

3.1 A New Conceptual Framework

The conceptual framework presented in the following sections embraces the complexity of the recently implemented integrated approaches to item generation and pre-calibration (i.e. Cho et al., 2014). The goal is that the framework is flexible enough to accommodate the complexities of layered and multistage item generation processes at level of generality that allows it to be broadly applicable to a range of topics within the AIG context (e.g., Gierl & Lai, 2012; Gierl, Lai, & Turner, 2012). In an effort to promote clarity, particularly in the specification

of pre-calibration models, the proposed framework draws less from the realm of assessment engineering and more from language consistent with evidentiary argumentation and evidence-based assessment (e.g. Mislevy & Riconscente, 2005).

3.1.1 Components of the Item Generation Process

Figure 4 visually summarizes the whole of the item generation process. This conceptual diagram highlights each of the components of this process, including a series of nested templates (*item forms, item models,* and *parent items*) and the variable content (*form-level characteristics, primary* and *secondary content*) to be integrated into those templates. Each of these components will be described in detail, beginning with the *educational objectives* which serve as the focus for sets of generated items, through to the algorithmic generation of specific *instantiations* that will appear as test items on a particular assessment. Each of these components and the relationships between them are described in detail in the sections that follow, beginning from the highest level and working down.

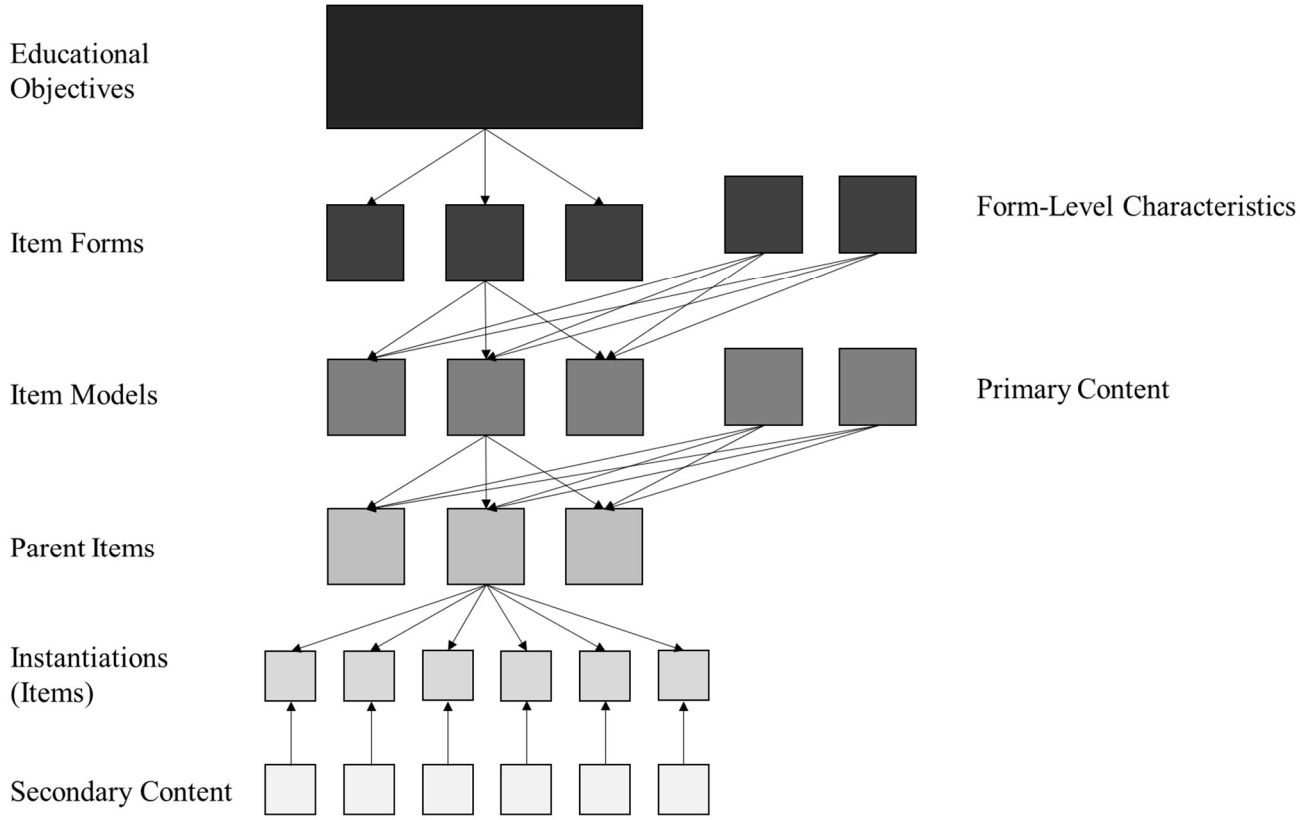

Figure 4. Item Generation Process for the Generative Process Model

Educational Objectives and Cognitive Task Models. Whether automatically generated or developed manually, items are intended to elicit evidence about underlying competencies, skills, or abilities. At the core of the item generation process is therefore a set of *educational objectives* which motivate item development and inform the design of any automatic item generation process. Aligned to each objective is a *cognitive task model* which characterizes, at a fairly high level of abstraction, the assessment environment in which examinees will say, do, or create something.

Effectively defining a cognitive task model requires both a comprehensive understanding of the construct being measured and a detailed understanding of the response processes governing examinees' demonstration of proficiency (Gorin, 2005). Both are necessary for specifying a set of generative rules, or grammar (Bejar et al., 2003; Irvine, 2002), that serves to

structure the item development process from the top down. These models specify the characteristic features of each task that are essential to eliciting the desired evidence about what examinees know or are able to do (Mislevy & Riconscente, 2005; Mislevy, Steinberg, & Almond, 2003). Within the context of AIG, these task models identify those features of the assessment environment that will be systematically manipulated in order to produce items designed to target the same proficiency but with varying degrees of difficulty and evidentiary focus (Gierl & Lai, 2013; Mislevy & Riconscente, 2005).

Item Forms. Cognitive models provide subject matter experts and test developers the framework necessary for developing wide array of assessment tasks targeting specific objectives. *Item forms* are each a unique realization of one possible structure for these tasks. Item forms are very abstract: each form is a template designed to accept a range of content that can be presented to the test taker in a variety of ways that nonetheless preserve essential task characteristics and maintain adequate alignment to the target objective(s).

As an example of an item form, first imagine a set of items which could be designed to assess students' understanding of probability. There are a number of ways that one might formulate a question about probability, including a question that is likely to be familiar to many, asking about drawing marbles from a jar. But a question about drawing marbles from a jar could also be formulated as a question about picking socks from a drawer or selecting balls from an urn. Each of these problems could be derived from a suitably generic and flexible template for a set of problems that involve the calculation of probability based on the selection of any object from a container among a group of similar objects with varying traits. This template is an example of an item form (Figure 5).

[NAME] has [INTEGER 1] [OBJECTS]in [CONTAINER]There are [INTEGER 2] [TRAIT 1] [OBJECTS], [INTEGER 3] [TRAIT 2] [OBJECTS], [INTEGER 4] [TRAIT 3] [OBJECTS]. [NAME] draws from the [CONTAINER] and selects [INTEGER 5] [OBJECTS]. What is the probability that [PROPORTION] of the[OBJECTS] are [TRAIT 1]?

Figure 5. Example item form

What here is referred to as an item form, a generic template which represents the first layer of an automated item generation process, appears elsewhere in the AIG literature as a template (Luecht, 2013) or an item shell (Deane, Graf, Higgins, Futagi, & Lawless, 2006). Unfortunately, neither of these terms are clearly defined in the literature relative to other components of the item generation process. In addition, "item shell" has a negative connotation, as it is most commonly used in the AIG literature to describe the shared characteristics of item clones (e.g., Gierl & Lai, 2012); and "template" is ambiguous given that algorithmic item generation is a template-based process.

Form-level or Display Characteristics. Each item form specifies the structure for the assessment task, including the format and grammatical structure of the item stem, while making allowances for variation in presentation, including the response format and number of and dependencies among response options (Gierl & Lai, 2012) as well as the presence, content, and format of any auxiliary information such as tables, graphs, or images. Within the current framework, the variable features which determine the presentation of information are collectively referred to as *form-level characteristics.* Unique combinations of form-level characteristics should be designed to produce sets of items that address a common educational objective at levels of difficulty and complexity that are appropriate for learners of different abilities (e.g., Luecht, 2002, 2013).

Item Models. The cross-classification of an item form with a particular combination of form-level characteristics produces an *item model*. Item models are still best understood as

templates, but they are more concrete than item forms, in that they fully describe the structure of the assessment task, its form and format, absent any of the content. In the context of automatic item generation, item models are of particular importance, because item models are the first *generative product* in any AIG process and represent a level at which pre-calibrated items could be banked. Item models are also an intermediate generative product, more well-specified than an "item shell" but also more abstract than a prototypical or parent item, for which there isn't a good analog in the AIG literature. Figure 6 shows two example item model which could be generated from the item form shown previously.

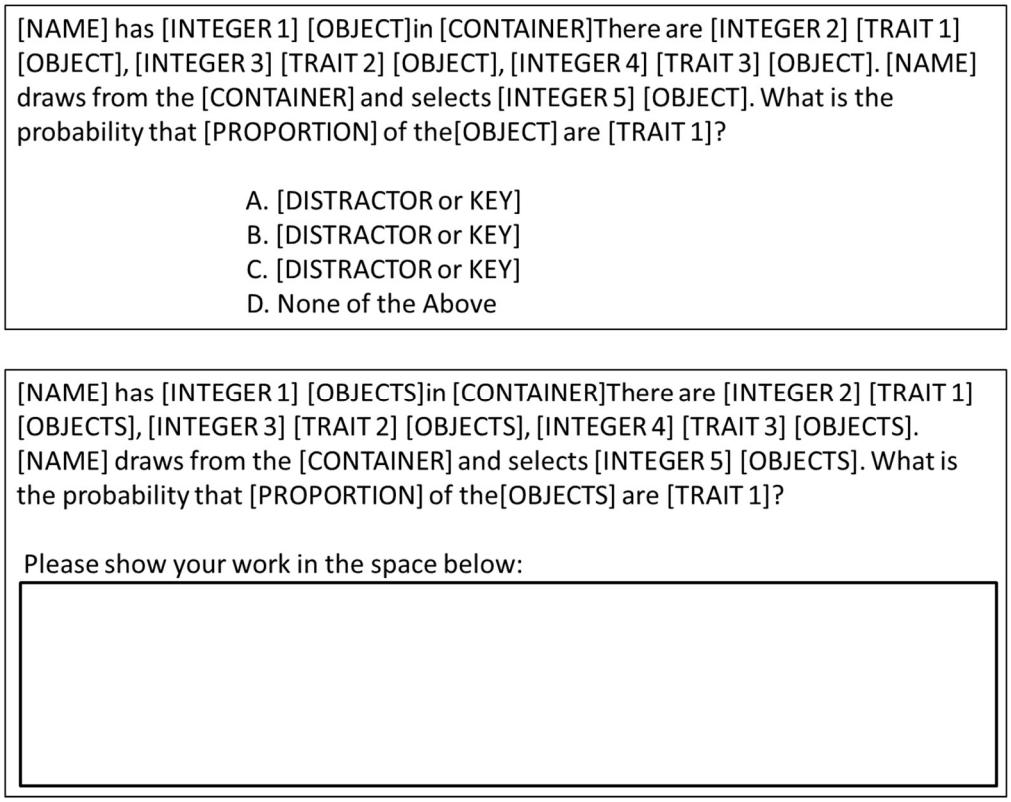

Figure 6. Example item models

Primary Content Integration. It is at this stage in the item generation process that content begins to be integrated into the item model. As discussed in the previous chapter, AIG researchers typically account for primary content integration through the inclusion of fixed

effects in a pre-calibration model (e.g., Cho et al., 2013; Embretson & Daniel, 2008; Sinaharay & Johnson, 2003). What is described here as *primary content,* is information that is directly relevant for the solution process and describes particular ranges or combinations of values that are expected to impact item difficulty. Importantly, these are not specific values but instead are specific categories or ranges of content. It is here, perhaps more than anywhere else, where the importance of a well-defined cognitive task model is evident. Primary content is referred to elsewhere in the AIG literature as design manipulations (Embretson, 1998; 1999), key content (Simpson et al., 2015), radicals (Gierl & Haladyna, 2013), or systemic manipulations (Gorin, 2005).

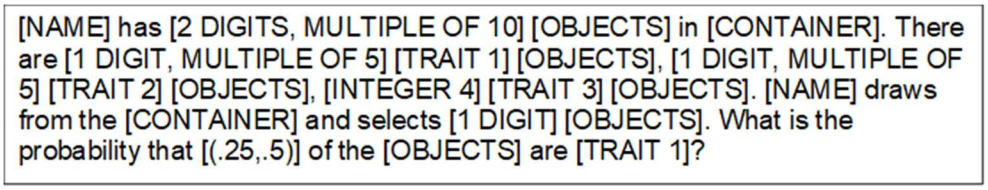

Figure 7. Example Parent Item

Parent Items and Item Families. The algorithmic integration of primary content into item models produces *parent items* which share the same form but differ in their difficulty and complexity as a result of their specific content. This language of parent items aligned with item families, where instantiations generated within families are psychometrically equivalent or nearly equivalent, is intuitive and also prevalent throughout the AIG literature (e.g., Sinharay & Johnson, 2013) and is retained within this framework.

Secondary (Specific) Content Integration. The final step in the item generation process is the integration of specific and *secondary content* to generate specific instantiations of each parent item. *Secondary content*, also referred to as incidentals or as surface features within the

AIG literature (e.g. Sinharay & Johnson, 2003; 2013), describes the specific values that are integrated into a template but are not relevant to obtaining a correct solution to the item.

Bob has 13 marbles in his jar. There are 6 blue marbles, 4 green marbles, and 3 red marbles. Bob reaches into the jar and selects 3 marbles. What is the probability that all of the marbles are blue?

Anita has 13 cupcakes in her display case. There are 6 vanilla cupcakes, 4 chocolate cupcakes, and 3 red velvet cupcakes. Anita reaches into the display case and selects 3 cupcakes. What is the probability that all of the cupcakes are vanilla?

Jose has 13 pieces of candy in his desk. There are 6 blue pieces of candy, 4 green pieces of candy, and 3 red pieces of candy. Jose reaches into the desk and selects 3 pieces of candy. What is the probability that all of the pieces of candy are blue?

Figure 8. Three example isomorphs with secondary content highlighted

Secondary content, as illustrated in Figure 8 above, although not critical to the solution, is still necessary to produce a usable test item. Here, the secondary content includes the person who is acting as the agent in the problem (Bob), the particular objects and the container in which they are in (marbles in a jar), and the characteristics or traits (color) of those objects. Important within the AIG context, each of these contextual elements can be manipulated, either individually or in concert, in order to create items which are are designed to be equivalent to and will be pre-calibrated with the assumption that they are exchangeable with all other items created from the same parent.

It is important to note that the decision to avoid the language of "radicals" and "incidentals" when talking about content integration was deliberate. Both of those terms describe components of the item generation process, but within the literature both are defined vis-à-vis a pre-calibration model, e.g. a radical is a manipulation that has a systematic impact on item properties. A core tenet of this framework is to propose a vocabulary that will maintain a

distinction, both conceptually and linguistically, between the generation process and the psychometric model(s) that could be used to pre-calibrate the generated items.

3.1.2 Summary

Together, these different components and the relationships between them describe the key features and steps in item generation processes commonly described though perhaps not clearly or completely defined in the AIG literature. Each *cognitive task model* is aligned to a specific educational objective, and provides a framework guiding the item generation process. From each cognitive model, a set of unique *item forms* are developed using those specifications. From each form, multiple *item models* are generated by specifying different combinations of *form-level characteristics,* including the number of response options or the presence of graphic support. Item models describe completely the structure of the task absent any content. In the context of automatic item generation, item models and parent items are of primary importance because it is these intermediate generative products, rather than individual items, which are pre-calibrated and banked. The algorithmic integration of *key content* into item models produces *parent items* which share a common form but differ in their difficulty and complexity as a result of their specific content. From each parent item a family of individual test items, or *instantiations,* are generated via the integration of *secondary content* which provides the context and color for the task.

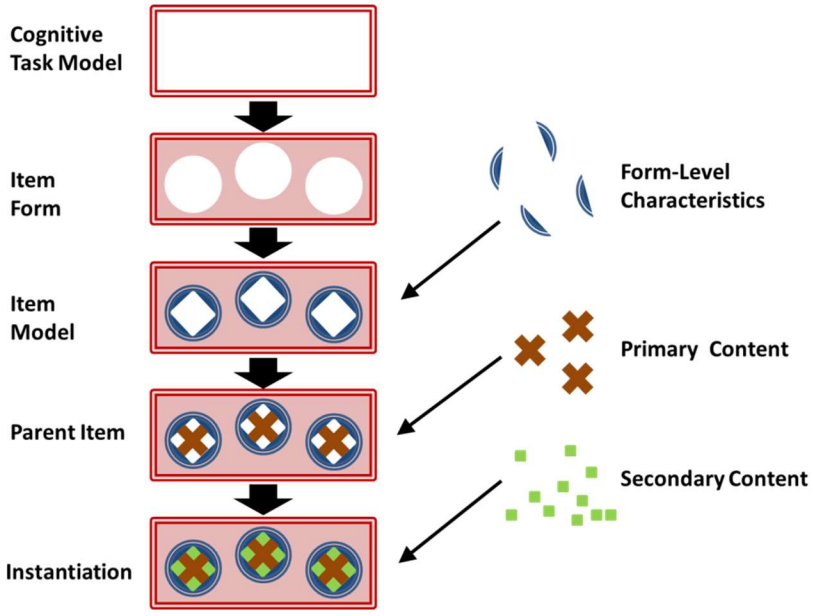

Figure 9. Notional graphic illustrating the item generation process underlying the Generative Process Model

3.2 The Generative Process Model

The proposed conceptual framework can be readily translated into a pre-calibration model for automatically generated items. Consistent with other models proposed for use in the AIG context, the proposed Generative Process Model (GPM) uses a multi-level framework to account for dependencies introduced by the item generation process and a combination of fixed and random effects to capture primary components of the generative process as described above. Unique among models proposed within the AIG literature, this model specifically incorporates parameters corresponding to intermediate generative products and the close alignment between model specification and the generative process is intended to support the direct assessment of the quality of the item generation process.

The structure of the GPM is presented below using a Rasch model as the item response model at Level 1. A 2PL or 3PL model could also be estimated within this framework, though additional assumptions would need to be made in order to ensure identification and convergence

of estimates. A simple model was selected for this exposition in order to highlight the alignment between the conceptual framework and the parameterization of the mean structure of the GPM.

3.2.1 Components of the Generative Process Model

Level 1: The Item Response Model. The first-level model specfies the predicted log odds of success of person p on instantiation i, $\eta_{ip(j_1,j_2)(k_1,k_2)t}$, as a function of that person's ability, and that particular item's characteristics which are defined by the various components of that item's generative process.

$$\eta_{ip(j_1,j_2)(k_1,k_2)t} = \theta_p - \beta_{i(j_1,j_2)(k_1,k_2)t}$$

It is important to note that not all examinees will interact with every item. However, missing responses to particular instantiations are considered to be missing at random and to improve readability without loss of generality, the pattern of missing data is not made explicit in the model notation (Geerlings, 2011).

Level 2: Item Family Mean and Within-Family Variation. Recall from the previous section that instantiations are differentiated from one another through the integration of secondary content. This integration of secondary content and the specification of different plausible values is expected to have some minor impacts on the properties of the resulting items, but not in a systematic way, and each instantiation generated from the same parent item is thus assumed to have the same average psychometric properties. Within each item family the effects of integrating secondary content are assumed to be random, and this is reflected in the Level 2 specification of the GPM:

$$\beta_{i(j_1,j_2)(k_1,k_2)t} = \mu_{0(j_1,j_2)(k_1,k_2)t} + e_{i(j_1,j_2)(k_1,k_2)t}, \text{ where } e_{i(j_1,j_2)(k_1,k_2)t} \sim N\left(0, \sigma_e\right)$$

Persons as well as items are assumed to be random. The ability parameter, $\theta_p$is therefore also defined as a random effect, with a population mean of μ_{θ_0} and a variance of σ_θ^2.

$$\theta_p = \mu_{\theta_0} + u_{\theta_p}, \text{ where } u_{\theta_p} \sim N(0, \sigma_\theta\ )$$

Items and persons are cross-classified, and the person residuals and item residuals are assumed to be independent. For identifiability in estimation, the assumption is that abilities are normally distributed with a mean of zero and standard deviation of one, $\theta_p \sim N(0,1)$.

Level 3: Item Model Mean and Primary Content Integration. Secondary content is assumed to have only minor and unsystematic impacts on item properties. In contrast, assigning different values to primary content elements is expected to systematically impact the psychometric properties of generated items, yielding parent items which differ from one another depending on the item model used to generate the parents and the particular combination of content features. This is reflected at Level 3 of the GPM, where the difficulty of each parent item is decomposed into the difficulty of the corresponding item model and the combined effects of the content integration:

$$\mu_{0(j_1,j_2)(k_1,k_2)t} = \pi_{00(k_1,k_2)t} + \pi_{01(k_1,k_2)t}X_1 + \pi_{02(k_1,k_2)t}X_2 + u_{0j_1(k_1,k_2)t} + u_{0j_2(k_1,k_2)t},$$

$$\text{where } u_{0j_1(k_1,k_2)t} \sim N(0, \sigma_{j_1}) \text{ and } u_{0j_2(k_1,k_2)t} \sim N(0, \sigma_{j_2})$$

$X_1$and $X_2$are design variables set equal to 1 when key content manipulations are present, and 0 otherwise. Only two effects are shown here for simplicity of exposition. This could be the summation of many more effects (e.g. Cho et al, 2014). As noted before, these effects reflect specific hypotheses about how variation in primary content will impact response processes. The effects included in the model to capture the impact of content integration are selected from a universe of possible effects; and because they summarize across values the coded effects are unlikely to fully capture the impact of primary content integration. There is consequently likely

to be some variation above and beyond what can be predicted by the specified fixed effects, and residuals are also included in the pre-calibration model at this level. Although a number of the hierarchical models proposed for use in AIG frameworks include second-level residuals (e.g. Cho et al., 2013; Geerlings, 2011), the rationale is typically statistical rather than acknowledging explicitly the potential "gap" between the pre-calibration model and the item generation process and the underlying response process.

Level 4: Item Form Mean and Form-Level Characteristics. The structure of Level 4 of the GPM mirrors that of Level 3. The difficulty of each item model is decomposed into the difficulty of the corresponding item form and the combined effects of the particular combination of display characteristics.

$$\pi_{00(k_1,k_2)t} = \gamma_{000t} + \gamma_{00} \quad Z_1 + \gamma_{0002}\ Z_2 + v_{00k_1t} + v_{00k_2t},$$

$$\text{where } v_{00k_1t} \sim N\left(0, \sigma_{k_1}\right) \text{ and } v_{00k_2t} \sim N\left(0, \sigma_{k_2}\right)$$

Z_1 and Z_2 are design variables set equal to 1 when key form-level characteristics are present, and 0 otherwise. Again, only two effects are shown in this discussion, representing, for example, the inclusion of graphic support on the one hand or the utilization of four randomly generated response options instead of using, as shown in Figure 6, an open-format response.

Level 5: Grand Mean and Variation Across Item Forms. The difficulty of each item form is parameterized at Level 5 as being drawn from a larger population all possible tasks that could be used to gather evidence about the targeted educational objective:

$$\gamma_{000} = \omega_{0000} + w_{000}\ , \text{where } w_{000t} \sim N(0, \sigma_w).$$

3.2.2 Summary

The Generative Process Model incorporates cross-classified fixed and random effects within a hierarchical structure that captures as completely as possible the item generation process. Using a Rasch model as the item response model at Level 1, the log odds of a correct response to an item generated via a multi-layered generation process can be written as follows:

$$\eta_{(i,j)p(m,k)(t,c)} = \theta_p - (\omega_{0000} + \gamma_{0001t}Z_1 + \gamma_{0002t}Z_2 + \pi_{01(k_1,k_2)t}X_1 + \pi_{02(k_1,k_2)t}X_2 + e_{i(j_1,j_2)(k_1,k_2)t} + u_{0j_1(k_1,k_2)t} + u_{0j_2(k_1,k_2)t} + v_{00k_1t} + v_{00k_2t} + w_{000t}) \quad (3.1)$$

where $\beta_{i(j_1,j_2)(k_1,k_2)t}$ is the difficulty of the *i*th generated item within an item family defined by the corresponding template *t* and unique combinations of key content and display characteristics. Item difficulties are randomly distributed around their family means, $\mu_{0(j_1,j_2)(k_1,k_2)t}$. Each family mean is decomposed into the mean of the item model and the cross-classified primary content manipulations as well as the random effects $u_{0j_1(k_1,k_2)t}$ and $u_{0j_2(k_1,k_2)t}$. The mean of each item model, $\pi_{00(k_1,k_2)t}$, is jointly defined by its template mean, γ_{000t}, and display characteristics as well as the random effects, $u_{00\ _1t}$ and u_{00k_2t}. The item forms used to seed the item generation process represent a sample of all possible tasks that could be used to gather evidence about the targeted educational objective.

3.2.3 Additional Considerations

An Extension. The framework presented in this chapter is designed to support a program of research whereby "validation is built into test development" (Gitomer & Bennett, 2002). The *cognitive task model* provides the guiding theoretical framework for distinguishing between primary and secondary content; and the corresponding fixed and random effects represent testable hypotheses derived from that framework. Within this framework, item generation is re-imagined a template-based process where content is iteratively integrated, and the proposed pre-

calibration model leverages a cross-classified structure in order to support the estimation of the properties of intermediate generative products, *parent items*, *item forms,* and *item models.* The granularity at which the generation process is defined and the corresponding statistical model is specified is unique within the AIG literature, as is the use of cross-classification to capture content integration. The proposed benefits of this framework are its flexibility, and its utility. Using this framework, estimated parameters of any of the generative products could be used for item banking. Additionally, by virtue of the close alignment between the generative process and the parameterization of the Generative Process model, the variance components at each level also could also inform the evaluation and refinement of the item generation process.

Future Application. The proposed definitions of the various process and model components presented in this chapter are intended to strike a critical and delicate balance: specific enough to support the interpretation of model parameters but at a level of abstraction that allows the framework to be broadly applicable to a range of topics within the AIG context. While this balance is achievable in theory, it is worth demonstrating how a conceptual framework that is sufficiently abstract to be generalizable can also be successfully applied. The goal of the next chapter is to explore the utility of the proposed framework through an analysis of item response data collected from the online administration of algorithmically generated items designed to assess elementary school students' computational fluency with addition, subtraction, and multiplication during the Summer Math Challenge Program (Simpson, Elmore, Bickel, & Price, 2015). The analysis is informed by a series of targeted simulation studies which examine the performance of the proposed mathematical model under a limited set of conditions which more closely resemble studies in the applied AIG literature versus the idealized conditions typically featured in the simulation literature.

Chapter 4: A Targeted Simulation and an Empirical Illustration

The AIG literature is, by definition, forward-thinking. Generation approaches and pre-calibration models are both being designed with the future in mind. In this future, item generation systems are well-oiled machines, both literally and figuratively, producing tens of thousands or even hundreds of thousands of items. That imagined future, however, does not map well onto the current reality of applied AIG research, where the total number of items being generated is within the range of tens or hundreds, albeit with a few notable exceptions (e.g., Gierl & Lai, 2012; Gierl, Lai, & Turner, 2012). And even in those cases where large numbers of items are being generated, the number of templates used to produce those items is not large, and item generation processes typically feature relatively few manipulations. In fact, particularly when a domain is well-understood by subject matter experts and the generation process is well-defined, AIG researchers may be forced to contend with small sample sizes.

Recent work by Simpson and colleagues (2015) is an illustrative case in point. In their initial evaluation of items generated for the Summer Math Challenge Program, Simpson and colleagues (2015) demonstrated that a large proportion of the variance in generated item difficulty could successfully be explained through a limited number of characteristics, coded at a relatively coarse grain-size. In fact, three features were identified as key drivers for the generation of thousands of like items. Drawing on that work, the item generation process was refined to systematically manipulate only a few key features that could be clearly defined and also resonated with content experts. These features were coded as binary design variables: sets of items were generated to have operands with a maximum of three digits or two, operands could be multiples of 10 or not, operands were paired so that students needed to employ regrouping as a solution strategy or not. The relative simplicity of this solution to the thorny problem of domain

modeling is compelling but it also highlights what could be a tension in AIG research: simple engineering solutions may present estimation challenges for those looking to pre-calibrate generated items, particularly given the ubiquity in AIG literature of increasingly complex hierarchical models.

The primary objective of this chapter is to demonstrate the applicability of the Generative Process Model through an analysis of item response data collected during the Summer Math Challenge Program (Simpson et al, 2015). The characteristics of these data additionally inform a series of targeted simulation studies which examine the performance of the proposed mathematical model within sample size constraints that AIG researchers may need to consider more deeply as they look to advance models for pre-calibration and understand the conditions under which model estimates can be meaningfully interpreted. More detail on the items generated for use during the Summer Math Challenge Program is provided in the next sections.

4.2 The Summer Math Challenge Program and the Math Item Generator

During the summer of 2014, more than 1,500 students participated in the MetaMetrics Summer Math Challenge Program (MetaMetrics, Inc., 2015). This elective online program provided students with supplemental math instruction during the summer months. The program offered students helpful hints for problem-solving, the opportunity to play math-centric online games, and provided weekly fluency exercises targeting specific constellations of math skills. The Math Item Generator (MIG), a template-based algorithmic item generation system, was used to produce the items included on those weekly exercises (Kellogg et al, 2015; Simpson et al, 2015) using a template-based approach that is illustrated in Figure 10 and described in detail in the following sections.

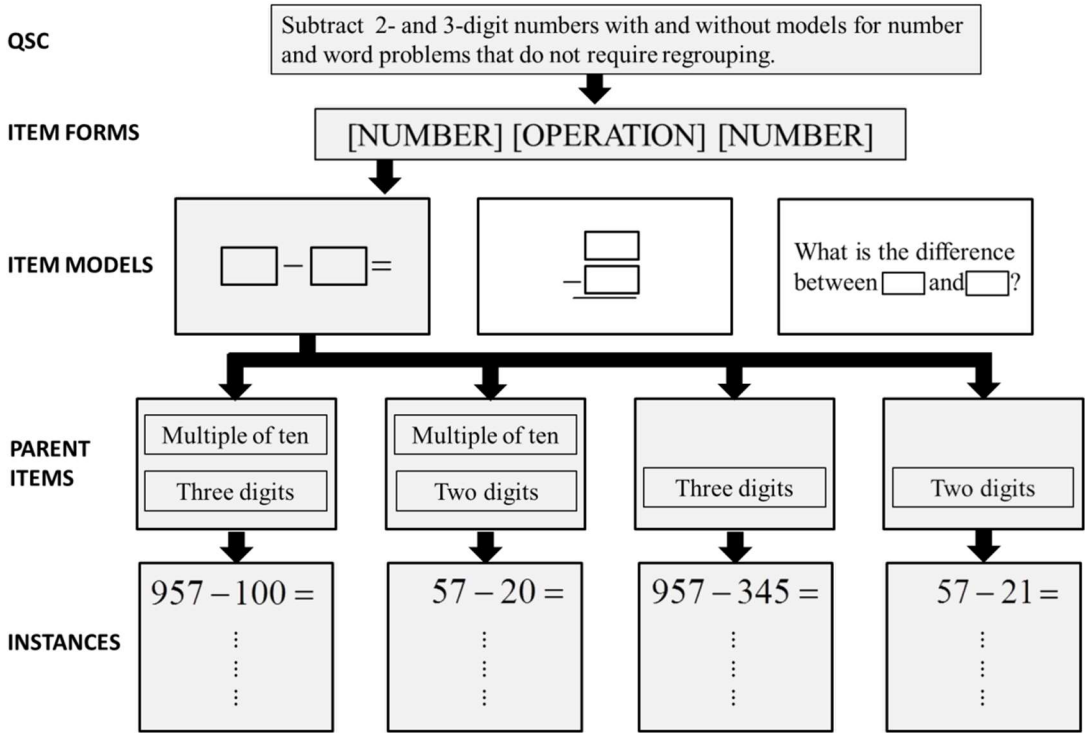

Figure 10. Illustration of generative process and products for example QSC, Item Form, and Item Model

4.2.1 Components of the Item Generation Process

QSCs and Educational Objectives. The core content and structure of the MIG is provided by the Quantile® Framework for Mathematics (MetaMetrics, Inc., 2011), which is comprised of approximately 550 K-12 mathematical skill and concept bundles that are aligned to Common Core Standards and also to grade-level knowledge and performance standards for selected states. Each of these Quantile Skills and Concept (QSC) bundles is a detailed and operationalizable description of mathematical skills and concepts that have been validated and empirically scaled (MetaMetrics, Inc., 2011). Each bundle describes a specific educational objective which the items generated by the MIG are designed to assess. The Summer Math Challenge Program fluency exercises feature 902 items aligned to one of *eleven* QSCs, as shown in Table 1 below.

Table 1. Distribution of Items and Observations by QSC

QSC	Description	Items		Observations	
		N	*Pct*	*N*	*Pct*
1	Add 3 single-digit numbers in number and word problems.	34	4%	3,190	4%
2	Use addition and subtraction facts to 20.	30	3%	2,116	3%
3	Add 2- and 3-digit numbers with and without models for number and word problems that do not require regrouping.	124	14%	9,245	11%
4	Use multiplication facts through 144.	94	10%	8,851	11%
5	Rewrite and compare decimals to fractions (tenths and hundredths) with and without models and pictures.	66	7%	6,391	8%
6	Find the fractional part of a whole number or fraction with and without models and pictures.	125	14%	7,136	9%
7	Know and use division facts related to multiplication facts through 144.	143	16%	20,684	25%
8	Estimate and compute products of whole numbers with multi-digit factors.	95	11%	6,103	7%
9	Add and subtract fractions and mixed numbers with like denominators (without regrouping) in number and word problems.	21	2%	1,681	2%
10	Estimate and compute sums and differences with decimals.	54	6%	5,856	7%
11	Subtract 2- and 3-digit numbers with and without models for number and word problems that do not require regrouping.	116	13%	10,131	12%

Item Forms. Aligned to the eleven QSCs, a total of *fifteen item forms* were identified, where each item form is defined – as illustrated in Figure 10 - by the unique combination of an educational objective and a particular mathematical operation. **Error! Reference source not found.** provides a brief description of each item form, along with the corresponding QSC and focal mathematical operation. Consistent with any template-based generation process, descriptions at this level are relatively generic, allowing for a range of different items to be generated from each form.

Table 2. Distribution of Items and Observations by Item Form

Item Form	QSC	Operation	Description	Items		Observations	
				N	*Pct*	*N*	*Pct*
1	1	Addition	Add 3 single-digit numbers in number and word problems.	34	4%	3,190	4%
2	2	Addition	Use addition facts to 20.	15	2%	1,089	1%
3	2	Subtraction	Use subtraction facts to 20.	15	2%	1,027	1%
4	3	Addition	Add 2- and 3-digit numbers with and without models for number and word problems that do not require regrouping.	124	14%	9,245	11%

5	4	Multiplication	Use multiplication facts through 144.	94	10%	8,851	11%
6	5	Decimals	Rewrite and compare decimals to fractions (tenths and hundredths) with and without models and pictures.	28	3%	2,618	3%
7	5	Fractions	Rewrite and compare decimals to fractions (tenths and hundredths) with and without models and pictures.	38	4%	3,773	5%
8	6	Multiplication	Find the fractional part of a whole number or fraction with and without models and pictures.	125	14%	7,136	9%
9	7	Division	Know and use division facts related to multiplication facts through 144.	143	16%	20,684	25%
10	8	Multiplication	Estimate and compute products of whole numbers with multi-digit factors.	95	11%	6,103	7%
11	9	Addition	Add fractions and mixed numbers with like denominators (without regrouping) in number and word problems.	6	1%	583	1%
12	9	Subtraction	Subtract fractions and mixed numbers with like denominators (without regrouping) in number and word problems.	15	2%	1,098	1%
13	10	Addition	Estimate and compute sums with decimals.	28	3%	2,972	4%
14	10	Subtraction	Estimate and compute differences with decimals.	26	3%	2,884	4%
15	11	Subtraction	Subtract 2- and 3-digit numbers with and without models for number and word problems that do not require regrouping.	116	13%	10,131	12%

Form-level Characteristics and Item Models. From each item form, the MIG produces sets of items that were either rendered as word problems or presented in numerical format, and displayed either horizontally or vertically (see Figure 11 below). Given the expectation that display format will systematically impact item difficulty (e.g., Simpson et al., 2015), display format is understood as the next layer of the generation process that will be parameterized as a fixed effect within the pre-calibration model. Each unique combination of a QSC, a mathematical operation, and a particular display format is conceptualized an item model, yielding *27 unique item models*. provides an illustration of how item models might align with an item form and an overarching educational objective.

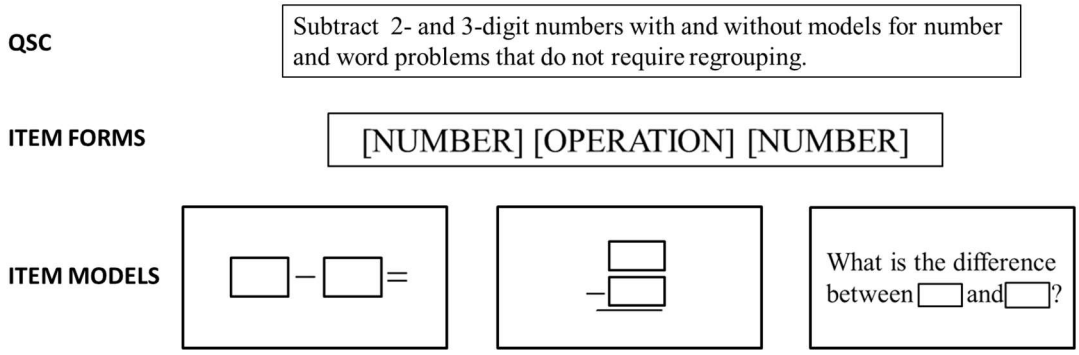

Figure 11. Example QSC, Item Form, and Item Models

Content Integration. As noted at the beginning of the chapter, the item generation process used by MetaMetrics was developed to systematically manipulate only a few key features, yielding sets of items with a maximum of two digits between the two operands, those with a maximum of three digits, items featuring numbers that are multiples of 10, and items requiring students to employ regrouping. These features, crossed with each of the 27 item models yields *45 parent items*, each featuring key content displayed in a particular way and aligned to a specific QSC and mathematical operation. **Error! Reference source not found.** shows the combinations of each of the four characteristics, QSCs, and mathematical operations that characterize each item family identified in the response data from the Summer Math Fluency exercises. For readability, not shown in the table are the display formats of these items.

Table 3. Distribution of Items across QSC, Operation, and Coded Content Features

QSC	Operation	Multiple of 10	Max Digits = 2	Max Digits = 3	Regrouping	Number of Items
1	Addition	0	0	0	0	34
2	Addition	0	1	0	1	15
2	Subtraction	0	1	0	1	15
3	Addition	0	1	0	0	25
3	Addition	0	0	1	0	26
3	Addition	1	1	0	0	23
3	Addition	1	0	1	0	23
4	Multiplication	0	0	0	1	94
5	Decimals/Fractions	1	1	0	0	9

5	Decimals/Fractions	1	0	1	0	19
5	Decimals/Fractions	1	1	0	0	9
5	Decimals/Fractions	1	0	1	0	20
6	Multiplication	0	0	0	1	125
7	Division	0	0	0	1	143
8	Multiplication	1	1	0	1	35
8	Multiplication	0	1	0	1	31
8	Multiplication	0	0	1	1	29
9	Addition	0	0	0	0	6
9	Subtraction	0	0	0	0	15
10	Addition	0	0	0	1	15
10	Addition	0	0	0	1	13
10	Subtraction	0	0	0	1	12
10	Subtraction	0	0	0	1	14
11	Subtraction	0	0	1	0	4
11	Subtraction	1	1	0	0	4
11	Subtraction	1	0	1	0	4
11	Subtraction	0	1	0	0	24
11	Subtraction	0	0	1	0	20
11	Subtraction	1	1	0	0	20
11	Subtraction	1	0	1	0	18

Item Families. The total number of items generated from each parent item (within each item family) ranges from 1 to 186. These families of items generated from each parent item are expected to have similar if not identical psychometric characteristics. Figure 10 illustrates how a variety of items might be generated from one item form aligned to a particular QSC through the variation of secondary content.

4.2.2 A Challenge for Pre-Calibration

Using the MIG, over 6,000 items were generated for use in the Summer Mathematics Program; and each generated item is the product of particular types of numbers, arranged and displayed in a particular format, combined using one or more mathematical operations, in order to assess students' proficiency relative to a specific set of mathematical skills and concepts which align to targeted grade-level standards.

The item generation process was designed by curriculum experts and psychometricians who worked to identify the key semantic and syntactic components of tasks which would, according to current theory and research, be likely to impact task difficulty (Kellogg et al, 2015). Researchers identified the set of features which could be manipulated to either affect students' problem representation or to increase the number of steps required to reach a desired solution, thereby increasing the difficulty of the task. Within constraints designed to ensure generated items' alignment to stated educational objectives, the MIG system was designed to systematically manipulate these key features as well as additional secondary or surface-level item characteristics to produce a large number of high-quality items of varying degrees of difficulty which target the same set of skills.

By all accounts, the MIG is a well-designed item generation system, generating thousands of items, but those items are the product of fewer item families, which are derived from fewer item models, which are aligned with only 15 item forms. The majority of the items available in the initial calibration sample align to only three of these forms, which target addition, multiplication, and subtraction (Figure 12).

Throughout the AIG literature there seems to be an assumption that pre-calibration will be aided by better engineering, but given the complexity of the model proposed for pre-calibration, the reality of the structure of these data raise some important questions. The next section describes a targeted parameter recovery study designed to examine how well generating values can recovered when sample sizes at the upper levels of the model are small, in line with both the motivating example of the Summer Math Challenge Program and the applied AIG literature more broadly.

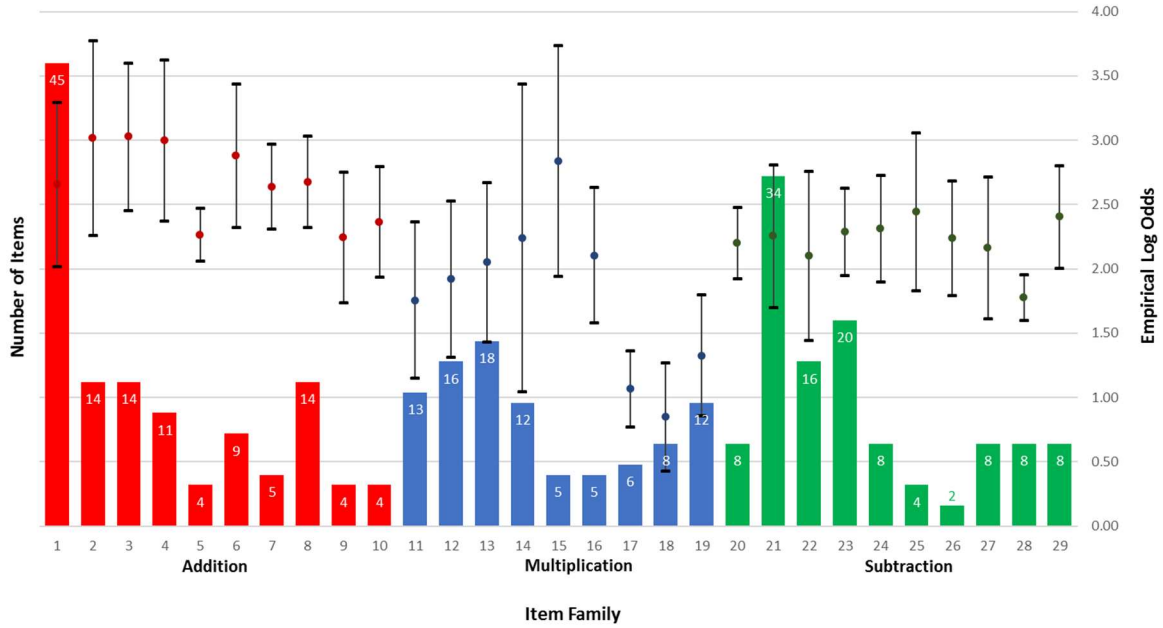

Figure 12. Distribution of Items and Mean Empirical Log Odds by Item Family and Item Form

4.3 Simulation Design

4.3.1 Simulation Objectives

Overall, the simulation study was designed to answer a simple question: given a limited number of item forms and a very efficient item generation process, how well does the Generative Process Model perform in an initial item calibration? Can the model successfully recover the generating parameters well enough that the estimates obtained during the initial calibration be sensibly used in pre-calibrating generated items? Finally, might the parameter estimates possibly provide some insight into the quality (consistency or inconsistency) of the underlying generation process?

4.2.2 Simulation Conditions

In thinking about the study design, it was necessary to balance a few different considerations. On the one hand, it was necessary to have some degree of alignment with the characteristics of the motivating data and context. But the study parameters cannot be so

narrowly defined that the study fails to be generalizable beyond that immediate context. In an effort to balance these concerns, the study was designed so that the number of components of the generation process and the complexity of that process were matched to the applied data. In order to promote generalizability, an effort was made to define study conditions that challenge what is often the central assumption underlying most AIG As outlined in Table 4: item difficulties and item response data were generated as if items were automatically generated through a well-designed and well-understood process, through a process that is poorly designed and not well understood, and through a process that generates items of variable quality.

Table 4. Generation Process Quality

		Generation Process				
		Well-Designed	Poorly Designed	Inconsistent Quality		
				Item Form 1	*Item Form 2*	*Item Form 3*
Within-family variance		0.1	0.4	0.1	0.1	0.4
Variance Explained by Design Features	*Level 2*	75%	40%	75%	75%	40%
	Level 3	75%	40%	75%	75%	40%

This variations in the quality of the item generation process were achieved by applying different sets of parameters to the data generation process: different degrees of within-family variation and adjusting the percentage of total variation explained by the fixed effects at each level. Given that a well-designed item generation process is typically assumed, the values were selected based on a review of applied AIG research. Across modeling approaches (including the LLTM and its multi-level variants), the percentage of variance in item difficulty successfully predicted by AIG researchers seeking to link response probabilities to item features typically ranges between 50% and 90% (e.g., Enright et al., 2002; Gorin, 2005), informing the percentage of variance explained

in this study. In studies on item cloning, within-family variation is commonly set between 0.1 and 0.5 (Geerlings, Glas, & van der Linden, Glas & van der Linden, 2003; Sinharay, Johnson, Williamson, 2003). Variations in the quality of the item generation process were crossed with factors denoting the complexity of the item generation process to yield a set of 27 conditions for the parameter recovery study.

Table 5. Summary of Simulation Conditions

Factor	Levels	Number of Levels
Quality of Generation Process	Well-Designed, Poorly Designed, Inconsistent	3
Number of Design Features at Level 3	2, 3, 4	3
Number of Design Features at Level 2	2, 3, 4	3
Number of conditions		27
Number of replications per condition		50

4.2.1 Data Generation

Statistical investigations of AIG are unique in that a population of items needs to be generated so that a sample of items can be drawn and administered, rather than thinking only about drawing a sample of examinees from a larger (theoretical) population. Because items are conceptualized as random and also parameterized as random in pre-calibration models, it is Important to the AIG context, items must be generated as a population. This was accomplished by first generating a matrix of item difficulties, then calculating a complete set of response data, and assembling data for each replication from that response matrix. This approach is similar to the methodology for generating response data detailed by Leucht (2012) and used elsewhere in the AIG literature (e.g. Geerlings, Glas, & van der Linden, 2011).

As a first step, people and items are generated separately. First, an ability vector, θ_p, was drawn for 250 simulees according to a normal unit distribution. This number of people was selected to be consistent with the number of students included in the data received from the

Summer Math Challenge Program. Second, a complete matrix of item difficulties was calculated consistent with the Generative Process Model (

Important to the AIG context*, items are generated as a population* rather than as a limited sample of items so that they might be appropriately modeled as random. The total number of items generated for the simulation therefore far exceeds the number that is considered in any single replication or even in the study as a whole: a total of 160,000 unique items were generated. These items were derived from 1,000 simulated item forms, and between two and four display features were manipulated on each of those forms to yield between 8 and 16 item models per form (8,000 – 16,000 total item models), and between 8 and 16 prototypical items were generated from each form (64,000 – 225,000 total item families). From each prototypical item, 10 instantiations were generated which differ from one another only in surface features to yield between 640,000 and 2,250,000 items per condition.

The next step is to calculate a complete *response probability matrix* for every person-item combination using Equation 3.1, with the appropriate number of fixed effects as described in Table 5. Each of the design features are all binary, with the value of each coefficient defined following Dardick & Harring (2008) so that the desired proportion of the variation between item families and between item models (40% or 75%) is explained by the linear combination of those features. As noted in the previous section, the within-family variation, is a simulation condition, and so σ_e^2 is either equal to .1 or .4. Item forms were defined as which are normally distributed with a mean of $\omega_{0000} = 0$ and a variance σ_w^2=1. The total variances at Levels 2, 3 were set at 0.6, 0.8, which is consistent with values found elsewhere in literature on the calibration of item families (DeBoeck, 2008; Sinharay & Johnson, 2005).

Following the calculation of the log odds of a correct response for each person-item combination, the inverse logistic function is used to transform that probability matrix into a matrix of *dichotomous response data*, where each row represents a complete response vector for each person. Response data are generated in this way once per condition to ensure consistency of response data within each condition (so that if a person "encounters" the same item more than once, the response vector is not inconsistent by chance), and random seeds were specified within the generation code to facilitate comparisons across conditions by minimizing sources of sampling variability.

Response data for each replication within each condition was assembled by drawing two samples: the first from among the available item pool, and the second from available simulees. For each replication, three item forms were selected at random and without replacement from the 1,000 available. For every item derived from each of those templates, the responses from 75 simulees were selected at random from the response matrix. Each person could only encounter each item once, though no restrictions were placed on how many or which items each person might see that were derived from the same item form and/or shared common design features. These values were again chosen to align with the empirical data, where only three item forms are available for investigation, and there are few restrictions on how students choose to interact with practice items.

Within each condition, the response data used for each replication was therefore comprised of (3 item forms) x (8-16 item models) x (8 x 16 prototypical items) x (10 instantiations per family) x (75 observations per item) = 11,390 – 45,560 rows, with each row containing a unique identifier denoting the person, an identifier for the item, and a dichotomous score variable. Each response was also indexed by relevant features of the generation process: an

identifier for item form and either a "1" or a "0" denoting the presence or absence of each of the design features denoting the integration of particular display characteristics or content. As noted above, a total of fifty replications were completed for each condition.

4.2.3 Analytic Model

For data generated under the well-designed and poorly designed conditions, the analytic model was specified to include template estimates as fixed effects, where the log odds of a correct response, $\eta_{(i,j)p(m,k)(t,c)}$, can be written as,

$$\eta_{(i,j)p(m,k)(t,c)} = \theta_p - (\gamma_{0001} + \gamma_{0002} + \gamma_{0003} + \gamma_{00010}Z_1 + \gamma_{00020}Z_2 + \pi_{01000}X_1 + \pi_{02000}X_2 + e_{i(j_1,j_2)(k_1,k_2)t} + u_{(j_1,j_2)(k_1,k_2)t} + v_{0(k_2,k_1)t})$$

For data generated under the Heterogeneous condition, a vector of values is estimated for each of the coefficients and each of the variance components to allow for variation across items generated from each template. In this condition, the log odds of a correct response is written as,

$$\eta_{(i,j)p(m,k)(t,c)} = \theta_p - (\gamma_{0001} + \gamma_{0002} + \gamma_{0003} + \gamma_{0001}\ Z_1 + \gamma_{0002t}Z_2 + \pi_{01(k_1,k_2)t}X_1 + \pi_{02(k_1,k_2)t}X_2 + e_{i(j_1,j_2)(k_1,k_2)t} + u_{(j_1,j_2)(k_1,k_2)t} + v_{0(k_2,k_1)t})$$

4.2.4 Verification of Generation Process and Analytic Model

Additional analyses were conducted prior to conducting the simulation in order to confirm that the data were being generated correctly. Using the full set of generated items, item difficulties were examined using a hierarchical model coded using the lme4 package to confirm that the random effects could be successfully recovered. As a second step, item parameters were estimated from a complete set of 12,000,000 generated responses (1,000 templates, two covariates each at levels 2 and 3, with 10 items per family and 75 observations per item). Parameter recovery was attempted under this "large sample" condition given the expectation that

small sample sizes are unlikely to support accurate parameter recovery, even if the data generation model and analytic model were correctly coded and applied. The generating model was used as the analytic model, and estimated under the same conditions as are outlined above. Generating parameters were satisfactorily recovered, as shown in Table 6 below.

Table 6. Generating Values and Parameter Estimates from "Large Sample" Condition

	True Value	Est. Mean	95% HDI	Eff N	R-hat
σ_e	0.316	0.307	(0.3, 0.314)	2951	1.001
σ_u	0.387	0.386	(0.369, 0.404)	1991	1.003
σ_v	0.447	0.434	(0.39, 0.477)	1008	1.005
σ_w	1	1.078	(0.938, 1.248)	768	1.003
π_{0100}	0.949	0.915	(0.874, 0.957)	1502	1.007
π_{0200}	0.949	0.883	(0.845, 0.923)	1781	1.003
γ_{0010}	-1.095	-1.004	(-1.097, -0.908)	2098	1
γ_{0020}	-1.095	-1.099	(-1.194, -1.006)	2020	1
ω_{0000}	0	0.051	(-0.22, 0.305)	279	1.03

4.2.5 Estimation

Estimation was performed using RStan to facilitate the estimation of cross-classified fixed and random effects (Stan Development Team, 2015). Each model was estimated using six chains with 5,000 burn-in iterations and 1,000 samples after warm-up. Each chain was initialized with random starting values. A non-centered parameterization was used when estimating the variances for each model. Half-normal priors were specified for each variance parameter that was estimated, with the upper bound of those priors estimated using half-normal *N(*0,.5) hyperpriors. In all cases, the ability parameter, θ_m is specified a normal variate, with a mean of 0 and standard deviation equal to 1. Following estimation, trace plots and sampling parameters for each chain were examined for convergence, in addition to monitoring divergences and Rhat values for each parameter.

4.3 Simulation Results

4.3.1 Study 1: Varying the Number of Item Models and Families

The results from the first simulation study are shown in **Error! Reference source not found.** and Table 8, which contain the relative bias of the parameter estimates and the empirical variation of those estimates, calculated across replications. Consistent with the literature examining the impact of small samples in hierarchical modeling and latent modeling frameworks, generating parameters are not well-recovered, particularly at the upper levels of the model.

As shown in **Error! Reference source not found.**, fixed effects are consistently underestimated. The bias is most severe for the estimates of template means, which were treated as incidental clusters; estimates of fixed effects at level two are the least biased, particularly when there are more covariates included at level 3 in the model. Similar to the estimates of fixed effects, variance estimates at level 3 are the most biased and the estimates of within family variance most closely match the generating parameters. In addition, the estimated variances more closely match the generating parameters when more covariates are included at level 3. Unlike the fixed effect estimates, variances are not consistently underestimated. Specifically, the variance between families within item models (calculated at level 2) is consistently overestimated.

4.3.2 Study 2: Varying the Quality of the AIG Process

Consistent with expectation, when items are poorly designed, meaning there is greater variability among items within item families and the manipulated design features explain less of the total variance at the upper levels of the model, there is a negative impact on the quality of the resulting estimates. What was unexpected was that only the recovery of fixed effect parameters appears to be impacted Table 9 and Table 10). Assuming the number of covariates in the model

are the same, the bias in the estimates of fixed effects under the poorly designed condition is approximately twice what it would be if the items were well designed and the calibration model was well-aligned to that process. The relative bias and variability of both the estimates of within-family variation and residual variances at the upper level of the model are similar in both the well-designed and poorly designed conditions. Similar to the pattern of results in the first study, what does seem to affect parameter recovery is the number of covariates in the model. In both the poorly designed and well-designed condition, when there are three covariates at levels two and three, meaning there are eight item models within each template and eight families within each item model, the residual variance at level 3 is no longer under-estimated, and the variances at levels 1 and 2 are consistently well-recovered.

Table 7. Relative Bias for Template Means, Fixed and Random Effects Across Conditions, Study 1

Number of Covariates		Template Means			Fixed Effects								Random Effects		
Level 3	Level 2	t1	t2	t3	c13	c23	c33	c43	c12	c22	c32	c42	u	v	w
2	2	-139.6	-197.6	-146.0	-36.4	-3.8			-18.3	16.1			0.7	24	-51.7
	3	-4.2	5.3	-448.7	5.6	-16.7			-9.9	-6.4	2		-2.4	1.8	-86.5
	4	-90.5	-827.6	-56.7	29.8	-46.6			-2.6	-9.6	-12.5	24.4	-1.3	11.8	14.8
3	2	-57.0	-1713.5	-110.3	-13.4	-4.3	-9.4		-1.5	7.9			-4.6	9.7	-27.9
	3	-147.0	-158.0	-235.1	3.6	-4.5	-25.3		2.2	-6	8.3		1.5	1.6	35.2
	4														
4	2	-75.2	-752.4	-17.1	-55	-3.6	-29	-10.3	-1.8	-13.1			0.1	11.9	21
	3														
	4														

Table 8. Variability of Estimates for Template Means, Fixed and Random Effects Across Conditions, Study 1

Number of Covariates		Template Means			Fixed Effects								Random Effects		
Level 3	Level 2	t1	t2	t3	c13	c23	c33	c43	c12	c22	c32	c42	u	v	w
2	2	0.031	0.034	0.029	0.024	0.019			0.023	0.022			0.014	0.013	0.023
	3	0.027	0.028	0.026	0.018	0.021			0.016	0.019	0.018		0.012	0.012	0.006
	4	0.018	0.022	0.019	0.015	0.014			0.014	0.012	0.012	0.013	0.007	0.007	0.009
3	2	0.019	0.022	0.020	0.016	0.014	0.014		0.015	0.013			0.010	0.012	0.015
	3	0.017	0.019	0.019	0.014	0.015	0.015		0.010	0.012	0.013		0.007	0.008	0.007
	4														
4	2	0.018	0.019	0.017	0.012	0.014	0.014	0.012	0.013	0.010			0.008	0.007	0.009
	3														
	4														

Note: Not all combinations of factors were fully explored. The shaded areas of the table indicate where no data is available.

Table 9. Relative Bias for Template Means, Fixed and Random Effects Across Conditions, Study 2

	Number of Covariates		Template Means			Fixed Effects								Random Effects		
	Level 3	Level 2	t1	t2	t3	c13	c23	c33	c43	c12	c22	c32	c42	u	v	w
Well-Designed	2	2	-139.6	-197.6	-146.0	-36.4	-3.8			-18.3	16.1			0.7	24	-51.7
		3	-4.2	5.3	-448.7	5.6	-16.7			-9.9	-6.4	2		-2.4	1.8	-86.5
	3	2	-57.0	-1713.5	-110.3	-13.4	-4.3	-9.4		-1.5	7.9			-4.6	9.7	-27.9
		3	-147.0	-158.0	-235.1	3.6	-4.5	-25.3		2.2	-6	8.3		1.5	1.6	35.2
Poorly Designed	2	2	-145.4	-176	-156.4	-77.3	-9.6			-42.7	36.7			-1.6	21.4	-62.9
		3	100.7	3	-480.1	6.1	-39.8			-22.3	-12.1	2.9		-4.4	3.7	-88
	3	2	-45.7	-1679.7	-105.5	-26.5	-11.8	-18.3		-2.9	19.1			-3.6	12	-30.2
		3	-150.7	-162.4	-216.2	11.3	-5.8	-55.4		6.9	-12.8	17.2		1.5	2	32.2

Table 10. Variability of Estimates for Template Means, Fixed and Random Effects Across Conditions, Study 2

	Number of Covariates		Template Means			Fixed Effects								Random Effects		
	Level 3	Level 2	t1	t2	t3	c13	c23	c33	c43	c12	c22	c32	c42	u	v	w
Well-Designed	2	2	0.031	0.034	0.029	0.024	0.019			0.023	0.022			0.014	0.013	0.023
		3	0.027	0.028	0.026	0.018	0.021			0.016	0.019	0.018		0.012	0.012	0.006
	3	2	0.019	0.022	0.020	0.016	0.014	0.014		0.015	0.013			0.010	0.012	0.015
		3	0.017	0.019	0.019	0.014	0.015	0.015		0.010	0.012	0.013		0.007	0.008	0.007
Poorly Designed	2	2	0.031	0.034	0.029	0.024	0.019			0.023	0.022			0.014	0.013	0.023
		3	0.027	0.028	0.026	0.018	0.021			0.016	0.019	0.018		0.012	0.012	0.006
	3	2	0.019	0.022	0.020	0.016	0.014	0.014		0.015	0.013			0.010	0.012	0.015
		3	0.024	0.021	0.023	0.017	0.017	0.014		0.012	0.013	0.015		0.006	0.009	0.007

Note: Not all combinations of factors were fully explored. The shaded areas of the table indicate where no data is available.

Table 11. Relative Bias for Template Means, Fixed and Random Effects Across Conditions, Study 2

Number of Covariates			Template Means			Fixed Effects							Random Effects		
Level 3	Level 2		t1	t2	t3	c13	c23	c33	c43	c12	c22	c32	c42	u	v	w
2	2	Well- Designed	-139.6	-197.6	-146.0	-36.4	-3.8			-18.3	16.1			0.7	24	-51.7
		Poorly Designed	-145.4	-176	-156.4	-77.3	-9.6			-42.7	36.7			-1.6	21.4	-62.9
		Heterogeneous	-194.6	-250.8		0.45	-41.6			-20.85	41.3			10.75	1.5	-1.5
					-274.5	-42.5	40.3			-64.4	-25.1			-43.05	-29.95	-36.6
2	3	Well- Designed	-4.2	5.3	-448.7	5.6	-16.7			-9.9	-6.4	2		-2.4	1.8	-86.5
		Poorly Designed	100.7	3	-480.1	6.1	-39.8			-22.3	-12.1	2.9		-4.4	3.7	-88
		Heterogeneous	-10.7	2.2		20.05	-8.6			-20.1	-27.3	13.5		-19.6	18.25	-3.65
					-281.7	-21.8	-171.4			54.8	-14.55	37.15		-53.35	-13.05	-37.3
3	2	Well- Designed	-57.0	-1713.5	-110.3	-13.4	-4.3	-9.4		-1.5	7.9			-4.6	9.7	-27.9
		Poorly Designed	-45.7	-1679.7	-105.5	-26.5	-11.8	-18.3		-2.9	19.1			-3.6	12	-30.2
		Heterogeneous	-44.6	-1957.5		-3.65	3.5	15.6		1.5	12.55			-2.55	8.85	22.1
					-98.9	-87.4	67.4	-42.6		44.3	-9.05			-51.95	-28.05	-23.7

Note: Not all combinations of factors were fully explored. The shaded areas of the table indicate where no data is available.

Table 12. Variability of Estimates for Template Means, Fixed and Random Effects Across Conditions, Study 3

Number of Covariates			Template Means			Fixed Effects							Random Effects		
Level 3	Level 2		t1	t2	t3	c13	c23	c33	c43	c12	c22	c32	c42	u	v	w
2	2	Well-Designed	0.031	0.034	0.029	0.024	0.019			0.023	0.022			0.014	0.013	0.023
		Poorly Designed	0.031	0.034	0.029	0.024	0.019			0.023	0.022			0.014	0.013	0.023
		Heterogeneous	0.058	0.052		0.063	0.074			0.076	0.074			0.027	0.007	0.025
					0.053	0.061	0.062			0.068	0.067			0.023	0.009	0.011
2	3	Well-Designed	0.027	0.028	0.026	0.018	0.021			0.016	0.019	0.018		0.012	0.012	0.006
		Poorly Designed	0.027	0.028	0.026	0.018	0.021			0.016	0.019	0.018		0.012	0.012	0.006
		Heterogeneous	0.075	0.066		0.07	0.07			0.073	0.082	0.073		0.03	0.009	0.024
					0.088	0.093	0.099			0.086	0.1	0.092		0.04	0.017	0.007
3	2	Well-Designed	0.019	0.022	0.020	0.016	0.014	0.014		0.015	0.013			0.010	0.012	0.015
		Poorly Designed	0.019	0.022	0.020	0.016	0.014	0.014		0.015	0.013			0.010	0.012	0.015
		Heterogeneous	0.096	0.109		0.069	0.07	0.068		0.077	0.076			0.03	0.018	0.045
					0.054	0.063	0.067	0.08		0.067	0.073			0.03	0.018	0.008

Note: Not all combinations of factors were fully explored. The shaded areas of the table indicate where no data is available.

4.3.3 Study 3: Introducing Inconsistency in Item Generation

Results from the third simulation study are shown in Table 11 and

Table 12. In each of these tables, relative bias and variability of estimates for the heterogenous condition are reported in two rows: the first row corresponds to parameter estimates for the templates that were well-designed, and the second row reflects parameter estimates for the third of the items generated from the poorly designed template.

These results suggest that under conditions when the sample size is small, with few templates and few specified design manipulations there is unlikely to be enough information to be able to accurately estimate the desired effects. Under the conditions studied here, when the impact of distinct manipulations are estimated for each item family and item model within each template, the magnitude of the bias in estimates of both fixed and random effects sharply increases. In particular, estimates of within-family variance are consistently under-estimated for those items which were poorly generated; and given the direction of the observed bias (underestimating variances and overestimating select fixed effects), reflects estimates for parameters that are more similar than they should be given the generating parameters. This result is not unexpected given the small sample sizes and is consistent with the findings elsewhere in that literature.

4.3.4 Discussion

The results of this study clearly underscore the challenges associated with estimating models under conditions of small sample sizes, where those sample sizes are determined not simply by the number of items, examinees, or observations but instead by the number of groups at each level of the model. Important within the AIG context, these results temper expectations around the extent to which parameter

estimates might be diagnostic of the quality of the item generation process when sample sizes are small, as was suggested in Chapter 3.

Looking ahead to the empirical analysis, a key take-away from the simulation study is to simplify the analytic model. Having a well-researched generation process and well-specified calibration model aligned to that process does not inoculate against the estimation challenges which arise from having a small number of templates is small and a relatively simple generation process. There may be reason to believe that there are different degrees of variation within and across item families, and understanding the magnitude of that variation may be important. However, variances are not well estimated and increasing model complexity may negatively impact the estimation of fixed effects, with little improvement in the estimation of random effects. A second insight gained from this simulation is to be cautious when interpreting item parameter estimates: the relative impact of generation features may be interpretable, but a strict interpretation of the value of fixed effects may lead to incorrect inferences about the nature of items or the generation process given the consistent underestimation of coefficients.

4.4 Analysis of Items Generated for the Summer Math Challenge Program

4.4.1 Analytic Data File

MetaMetrics provided response data comprised of more than 80,000 observations, which are the responses from more than 1500 rising third through sixth graders who participated in the Summer Math Challenge program. These response data, along with a file containing item characteristics and the original calibration

values, provide a richly contextualized opportunity to examine the properties of algorithmically generated items.

The mechanics of the item generation process implemented through the MIG and the theory guiding its development provide the basis for specifying a pre-calibration model using the conceptual and mathematical framework provided in the previous chapter. Codes were developed based on the characteristics and components of the item generation process discussed above and then applied to response data received from MetaMetrics. Each item was indexed by its generating item form, the corresponding item model, parent item, display format and coded content characteristics representing characteristics of the item operands. Together, the applied codes identify items which target the same educational objectives and also share an evidentiary focus and key content characteristics. The codes should, in theory, identify those items which have similar, if not identical, psychometric properties. Once all of the items were indexed, the following criteria were applied to the response data for inclusion in the analysis: item families were required to have at least two items present in the data, item models were required to have at least two parent items, and at least two item models needed to align with each item form. In addition, in those cases where students encountered the same item more than once (e.g. on both a homework assignment and practice exercise for a particular week), only the first student-item interaction was retained.

Of the approximately 902 items included in the response data received from MetaMetrics, a subset of 335 items was identified for inclusion in the analysis. The resulting analytic data file contained a total of 25,479 observations corresponding to

responses from 767 students. The items selected for analysis represent a concentrated subset of the original items from three distinct templates, designed to assess students' proficiency with addition, subtraction and multiplication as described by the corresponding QSCs (Table 13).

Table 13. Distribution of Items and Observations by QSC

			Items		Observations	
QSC	**Operation**	**Description**	*N*	*Pct*	*N*	*Pct*
3	Addition	Add 2- and 3-digit numbers with and without models for number and word problems that do not require regrouping.	124	37%	9,245	36%
8	Multiplication	Estimate and compute products of whole numbers with multi-digit factors.	95	28%	6,103	24%
11	Subtraction	Subtract 2- and 3-digit numbers with and without models for number and word problems that do not require regrouping.	116	35%	10,131	40%

4.4.2 Analytic Models

Six models are applied to these data. Each the calibration model specifies the predicted log odds of success of person *j* on test item *i*, as a function of that person's ability, and some combination of that particular item's characteristics, as defined by the components of that item's generative process. Each model may provide a different model-data fit to the initial calibration sample being considered, and each model also offers a different approach to item pre-calibration.

Random Person Random Item Model (RPRI). This model treats both persons and items as random, acknowledging that both are selected from larger populations. This model does not include parameters which connect item properties to the generative process, and as such the parameters estimated with this model do not provide any guidance to practitioners who are using these items and hoping to further refine or

modify the item generation process used to create these items. The log odds for the analytic model can be written as,

$$\eta_{(i,p)} = \theta_p - (\pi_{\beta_0} + u_{\beta_i})$$

Although this model may offer decent model-data fit, the concern is that it does't successfully differentiate among subgroups of items. Using this model, all items would be banked on the global mean for all items.

Linear Logistic Test Model with Error (LLTM + e). This model decomposes the mean difficulty of the items into a sum of the parent item characteristics; a residual term is included in the model to capture variance in item difficulties that is unexplained by parent item attributes. This model ignores the multi-level structure that produced the items, and as such it does not provide any information about template or item model characteristics. The model could provide some guidance to item developers by estimating the impacts of form-level characteristics and primary content integration. As estimated, the log odds for the analytic model can be written as,

$$\eta_{(i,p)(t,k)} = \theta_p - (\pi_{\beta_{0t0}} + \pi_{\beta_{001}} + \pi_{\beta_{002}} + \pi_{\beta_{003}} + \pi_{\beta_{004}} + \pi_{\beta_{005}} + u_{\beta_{i(t,k)}})$$

Using this model, items would be banked based on the template mean and also the particular combination of design features activated for that item.

Generative Process Model. This model incorporates cross-classified fixed and random effects within a hierarchical structure that captures as completely as possible the item generation process. This model has three levels, incorporating fixed effects to capture form-level characteristics at level 3 as well as fixed effects to capture primary content integration at level 2. The model also includes random item residuals

to capture unexplained variation in item models which share the same template and form-level characteristics, as well as residuals to capture unexplained variation in item family properties, after taking into account both item structure and primary content features.

For this analysis, four versions of the model were estimated: (1) a model in which the regression parameters were constrained to be equal across templates and item models, with templates treated as fixed effects; (2) a model in which regression parameters were constrained to be equal across groups, but templates were treated as random; (3) a model in which regression parameters were permitted to vary across templates and item models, with templates treated as fixed; and (4) a model in which regression parameters were permitted to vary across templates and item modes, with templates treated as random at level four. In all cases, only a single variance component was estimated at each level.

For the constrained model with templates estimated as fixed effects, the log odds of a correct response, $\eta_{(i,j)p(m,k)(t,c)}$, can be written as,

$$\eta_{(i,j)p(m,k)(t,c)} = \theta_p - (\gamma_{0001} + \gamma_{0002} + \gamma_{0003} + \gamma_{00010}Z_1 + \gamma_{00020}Z_2 + \pi_{01000}X_1 + \pi_{02000}X_2 + e_{i(j_1,j_2)(k_1,k_2)t} + u_{(j_1,j_2)(k_1,k_2)t} + v_{0(k_2,k_1)t})$$

In contrast, for the unconstrained model, a vector of values is estimated for each of the coefficients, and the log odds of a correct response is written as,

$$\eta_{(i,j)p(m,k)(t,c)} = \theta_p - (\gamma_{0001} + \gamma_{0002} + \gamma_{0003} + \gamma_{000}\ \ Z_1 + \gamma_{0002}\ Z_2 + \pi_{01(k_1,k_2)t}X_1 + \pi_{02(k_1,k_2)t}X_2 + e_{i(j_1,j_2)(k_1,k_2)t} + u_{(j_1,j_2)(k_1,k_2)t} + v_{0(k_2,k_1)t})$$

When items are calibrated using the constrained model with templates treated as random, the log odds of a correct response is as follows,

$$\eta_{(i,j)p(m,k)(t,c)} = \theta_p - (\omega_{00000} + \gamma_{00010}Z_1 + \gamma_{00020}Z_2 + \pi_{01000}X_1 + \pi_{02000}X_2$$
$$+ e_{i(j_1,j_2)(k_1,k_2)t} + u_{(j_1,j_2)(k_1,k_2)t} + v_{0(k_2,k_1)t} + w_{000t}$$

The unconstrained model with random effects at Level 4 estimates the log odds of a correct response as,

$$\eta_{(i,j)p(m,k)(t,c)} = \theta_p - (\omega_{0000} + \gamma_{0001}\ Z_1 + \gamma_{0002}\ Z_2 + \pi_{01(k_1,k_2)t}X_1 + \pi_{02(k_1,k_2)t}X_2$$
$$+ e_{i(j_1,j_2)(k_1,k_2)t} + u_{(j_1,j_2)(k_1,k_2)t} + v_{0(k_2,k_1)t} + w_{000t}$$

Using the GPM offers the most flexibility with respect to item banking and pre-calibration, as items could be banked based on estimates for prototypical items, item models, or item forms. It is worth noting that the models which parameterize template means as fixed effects may provide additional information that can be used for pre-calibration, though the simulation results suggestion caution in over-interpreting these estimates.

4.4.5 Estimation

Estimation was performed using RStan to facilitate the estimation of cross-classified fixed and random effects (Stan Development Team, 2015). Each model was estimated using six chains with 10,000 burn-in iterations and 1,000 samples after warm-up. Each chain was initialized with random starting values. A non-centered parameterization was used when estimating the variances for each model. Half-normal priors were specified for each variance parameter that was estimated, with the upper bound of those priors estimated using half-normal *N(*0,1) hyperpriors. In all cases, the ability parameter, θ_m is specified a normal variate, with a mean of 0 and standard deviation equal to 1. Following estimation, trace plots and sampling

parameters for each chain were examined for convergence, along with divergence information and Rhat values for each parameter.

4.4.6 Results

An examination of the distribution of item difficulties across each of the proposed models, as well as the correlation of item difficulties across conditions suggests that in many ways the resulting estimates from each of the models are similar to one another. However, the unconstrained Generative Process Model, with the inclusion of template means as fixed effects appears to offer the best model-data fit , in addition to the resulting parameter estimates providing some insight into the nature of the data generation process.

Table 14. Deviance Information Criteria for Six Analytic Models

		Parameters	Log Likelihood	DIC
Random Person Random Item		4	-6204.618	13500.35
Linear Logistic Test Model		17	-6215.123	13515.97
Generative Process Model	Const., Fixed	17	-6218.293	13582.7
	Const., Random	18	-6217.664	13566.82
	Unconst., Fixed	37	-6213.604	**13480.4**
	Unconst., Random	38	-6213.839	13539.56

Item parameter estimates for the GPM are shown in Table 16, organized to clearly illustrate the varying impacts of feature manipulations depending on the template from which items are generated. Displaying the data in this way underscore the potential utility of ensuring a close alignment between the generation process and the calibration model. Across templates, word problems are consistently more difficult, and this particularly true among multiplication problems. The allowance of three-digit versus two-digit operands in multiplication tables that are formulated as word problems generates more difficult items. The inclusion of operands that are multiples of 10 produces items which are consistently easier.

Table 15. Correlation Between Item Difficulty Estimates Using Different Calibration Models, by Template

						Generative Process Model		
			Original Calibration	RPRI	LLTM	Const., Fixed	Const., Random	Unconst., Fixed
Addition	Random Person Random Item		0.806					
	Linear Logistic Test Model		0.826	0.908				
	Generative Process Model	Constrained, Fixed	0.799	0.910	0.966			
		Constrained, Random	0.800	0.913	0.966	1.000		
		Unconstrained, Fixed	0.775	0.871	0.916	0.984	0.983	
		Unconstrained, Random	0.767	0.868	0.913	0.983	0.982	1.000
Multiplication	Random Person Random Item		0.937					
	Linear Logistic Test Model		0.916	0.978				
	Generative Process Model	Constrained, Fixed	0.887	0.936	0.983			
		Constrained, Random	0.888	0.936	0.982	1.000		
		Unconstrained, Fixed	0.889	0.914	0.948	0.978	0.979	
		Unconstrained, Random	0.887	0.914	0.947	0.978	0.979	1.000
Subtraction	Random Person Random Item		0.622					
	Linear Logistic Test Model		0.759	0.890				
	Generative Process Model	Constrained, Fixed	0.634	0.951	0.942			
		Constrained, Random	0.628	0.953	0.939	1.000		
		Unconstrained, Fixed	0.607	0.933	0.883	0.983	0.983	
		Unconstrained, Random	0.597	0.934	0.878	0.982	0.982	1.000

Table 16. Parameter Estimates for Item Generation Process Components Using the Unconstrained Generative Process Model for Item Calibration

		HDI		
	Mean Est.	*2.50%*	*97.50%*	*N Eff.*
Addition	-2.967	-3.758	-2.088	1015
Formulated as a Word Problem	0.286	-1.247	1.539	463
Horizontal Orientation	0.146	-1.215	1.304	865
Numeric, Horizontal Orientation				
Includes Multiples of 10	-0.552	-1.297	0.131	2159
Includes 3-Digit Integers	0.027	-0.740	0.823	2590
Word Problems				
Includes Multiples of 10	-0.199	-0.847	0.411	2110
Includes 3-Digit Integers	-0.481	-1.107	0.136	1576
Numeric, Vertical Orientation				
Includes Multiples of 10	-0.158	-1.133	0.850	1995
Includes 3-Digit Integers	-0.555	-1.426	0.367	2128
Multiplication	-2.422	-3.358	-1.554	329
Formulated as a Word Problem	1.206	-0.082	2.533	886
Horizontal Orientation	-1.000	-2.475	0.475	568
Numeric, Horizontal Orientation				
Includes Multiples of 10	-0.396	-1.131	0.310	2133
Includes 3-Digit Integers	0.136	-0.585	0.868	656
Word Problems				
Includes Multiples of 10	-0.031	-0.887	0.810	1872
Includes 3-Digit Integers	0.866	-0.132	1.902	2057
Numeric, Vertical Orientation				
Includes Multiples of 10	-0.436	-1.229	0.398	1559
Includes 3-Digit Integers	-0.103	-0.902	0.752	1341
Subtraction	-2.805	-3.674	-1.833	135
Formulated as a Word Problem	0.627	-0.523	1.864	208
Horizontal Orientation	-0.070	-1.264	1.302	350
Numeric, Horizontal Orientation				
Includes Multiples of 10	-0.001	-0.500	0.490	2417
Includes 3-Digit Integers	0.235	-0.315	0.792	1744
Word Problems				
Includes Multiples of 10	0.005	-1.110	1.182	2493
Includes 3-Digit Integers	0.038	-0.926	0.980	1923
Numeric, Vertical Orientation				
Includes Multiples of 10	-0.183	-1.027	0.675	2416
Includes 3-Digit Integers	-0.570	-1.386	0.248	2288
Var. within Families	0.390	0.316	0.389	1459
Resid. Var. within Models	0.149	0.002	0.097	311
Resid. Var. within Templates	0.259	0.002	0.160	56

4.4.7 Discussion

Model Utility and Interpretability. Perhaps the most compelling argument for the utilization of the proposed calibration framework is that the parameters are readily interpretable. In addition, there is something reassuring about the distribution of item difficulties achieved using the Generative Process Model (Figure 13), which suggests a greater separation between items generated from different templates: we might expect that addition, multiplication, and subtraction items are not only qualitatively different from one another but they differ systematically in their average level of difficulty. This separation stands in contrast to the distribution of item difficulties estimated vis a vis the RPRI (Figure 14), and also offers a better match to the distribution of original item calibrations.

In light of the simulation work, however, while the relative location of the template means seems reasonable it is necessary to question how the location of those template means should be interpreted. The estimated means are extremely low, suggesting that, on average, all of the items are very easy. On the one hand, this may not be unreasonable. A finding noted in the original validation study conducted by Simpson and colleagues. As shown in Table 17 below, there very little variation in response patterns for many of the items, which were intended as practice and were frequently too easy given respondents' knowledge and abilities (Simpson, Kosh, Bickel, Elmore, Sanford-Moore, Koons, & Enoch-Marx, 2015). On the other hand, these estimates are consistent with what we observed in the simulation study: consistent under-estimation of fixed effect parameters, including template means when those are estimated directly.

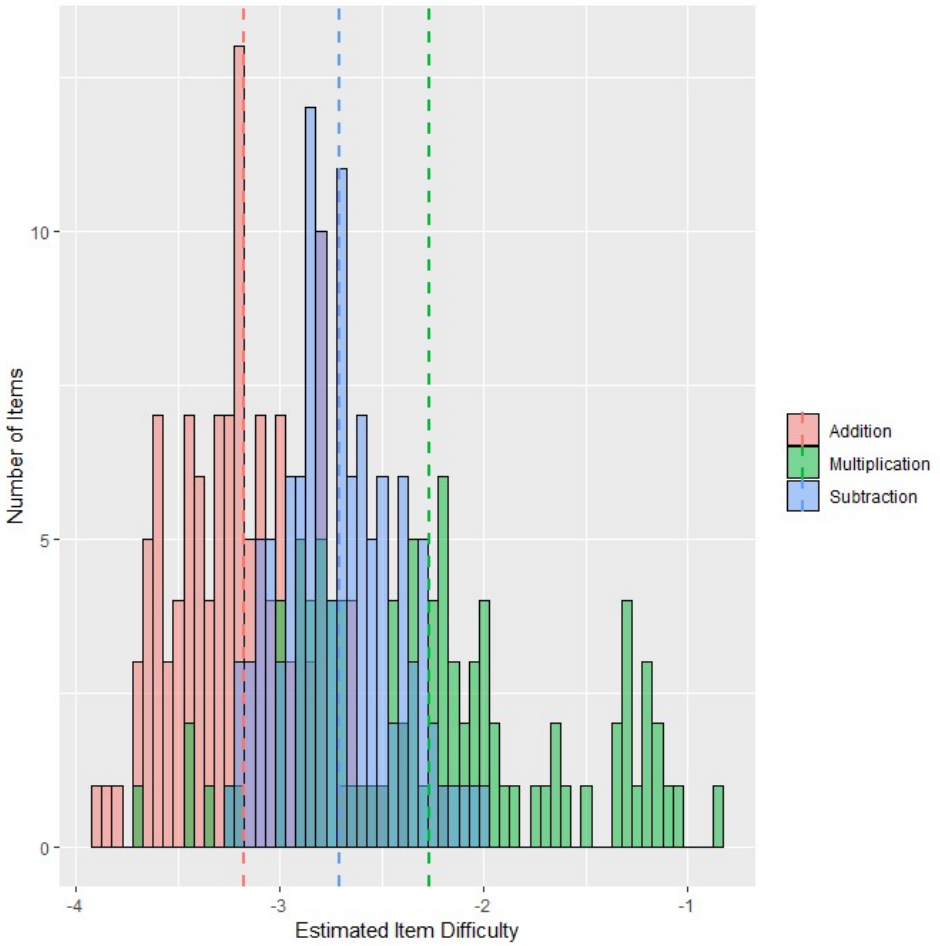

Figure 13. Count Distribution of Estimated Item Difficulties for Addition, Multiplication, and Subtraction Items Using the Unconstrained Generative Process Model with Fixed Template Estimates

Table 17. Response Details by Item Form

Item Form	QSC	Operation	Obs	Items	Percentage Correct		
					Min	*Mean*	*Max*
1	76	Addition	2844	29	0.88	0.94	1.00
2	78	Addition	775	12	0.91	0.96	1.00
3	78	Subtraction	973	13	0.73	0.90	0.96
4	79	Addition	6604	93	0.81	0.94	1.00
5	121	Multiplication	7516	81	0.68	0.89	1.00
8	160	Multiplication	6892	120	0.35	0.70	0.86
10	170	Multiplication	5601	87	0.58	0.84	1.00
11	199	Addition	184	2	0.85	0.90	0.97
12	199	Subtraction	364	6	0.88	0.95	1.00
13	201	Addition	2493	21	0.74	0.89	0.98
14	201	Subtraction	1647	15	0.82	0.90	0.98
15	599	Subtraction	9667	110	0.79	0.90	1.00

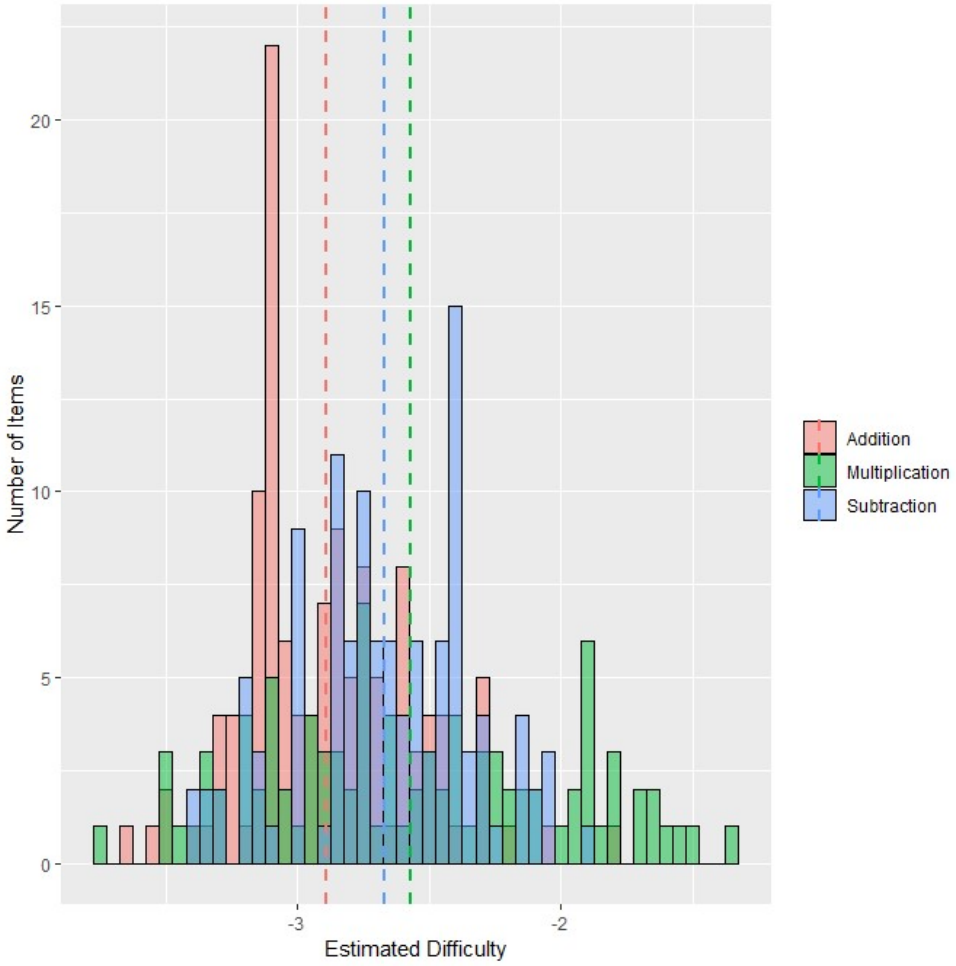

Figure 14. Count Distribution of Estimated Item Difficulties for Addition, Multiplication, and Subtraction Items Using the Random Person Random Item Model

Ability Estimates. While item parameter estimates illustrated the potential utility of the generative process framework, these results did not provide much insight into how the application of different calibration models could impact estimates of student abilities. Across the six models, resulting estimates of student abilities were consistently highly correlated. Figure 15 shows the comparability across models of the distribution of student ability estimates: the distributions overlap with one another almost perfectly.

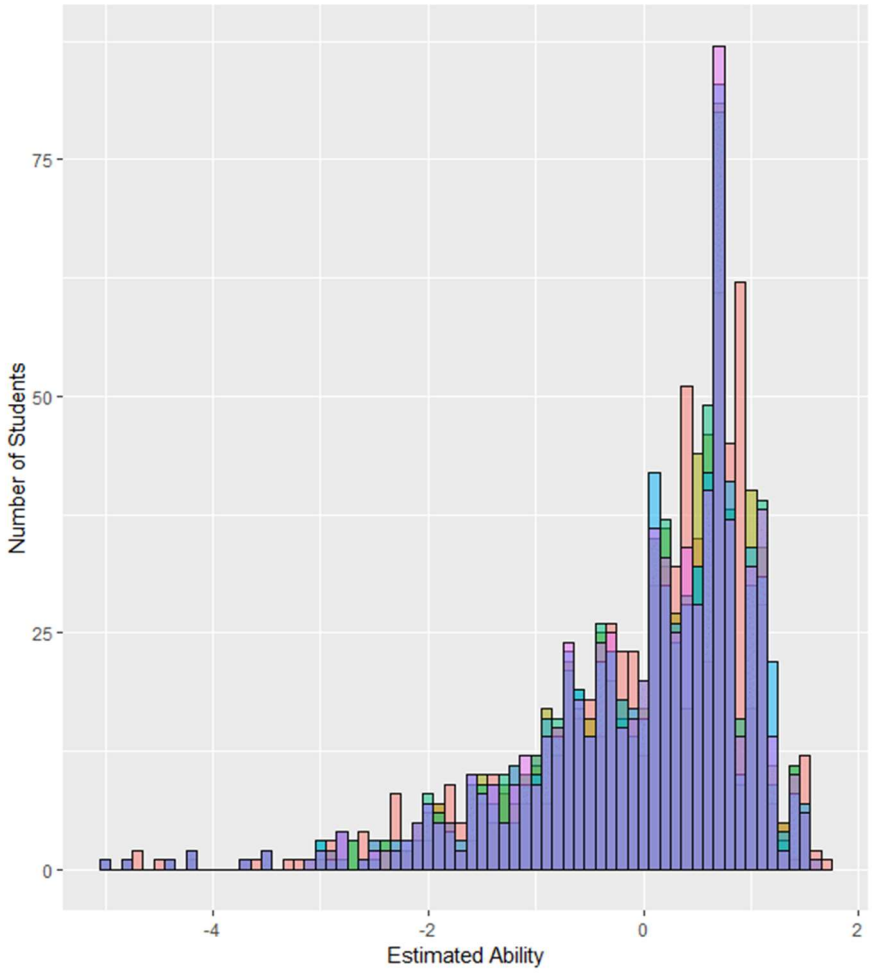

Figure 15. Count Distribution of Estimated Abilities Across Analytic Models

Unfortunately, the extent to which model specification did not impact student ability estimates is likely an artifact of the data used for this analysis. The distribution of estimated abilities is skewed, a pattern that could be explained by the absence of very difficult or even moderately challenging items. A reasonable interpretation of this result is that practice items were able to successfully differentiate between students for whom the items were better matched, but the generated items were not effective in differentiating between students with greater mathematics proficiency.

The impact of model specification on ability estimates is an area which warrants further research.

Chapter 5: A Targeted Exploration of Misspecification

There are no studies within the AIG literature which specifically investigate the impact of model misspecification on resulting parameter estimates, despite the fact that model misspecification is not uncommon in practice. As Gorin (2005) notes, the "applications of the linear logistic test model [LLTM] for specification of cognitive processes are somewhat rare due to the accuracy and completeness required by the list of cognitive processes to achieve model fit" (p. 369). She also identifies as the most significant challenge for implementing AIG "the development and verification of a viable cognitive model and an associated task feature model... [that] contains features that realistically can be manipulated to affect processing in such a way that item difficulty is reliably predicted" (p. 351). Articulating similar challenges in the context of their own research, Sheehan and Mislevy (2001) described their efforts to predict the characteristics of sentence completion items, finding that contrary to expectation, the relation between calibrated item difficulty and the cognitive difficulty of key or stem vocabulary as rated by experts was imprecise. The authors reported that a considerable proportion of item difficulty was left unexplained even after taking into account those features of the items that were expected to have an impact on the response process. Arendasy and Sommer (2007) acknowledged the challenge of specifying a pre-calibration model in their effort to generate and accurately predict the properties of quantitative reasoning items. They noted that, even when cognitive processes are well-researched and items can be successfully generated according to principles derived from cognitive models, predicting the psychometric properties of items based on the application of design decisions can be

problematic. The analytic results in the previous chapter also raised some questions: could some of the bias in parameter estimates be attributed to model misspecification? One possible explanation for the observed bias in estimates is that the generated items were too easy for the students who were answering them, resulting in response vectors with little to no variation, and these data quality issues were compounded by the relatively small number of item forms. It is worth asking, however, if there might be another explanation and to understand the conditions under which and in what ways errors in model specification might propagate, offering AIG researchers a way to diagnose problems with the generation process or with the structure of the model proposed for pre-calibration.

5.1 Simulation Objectives

This chapter describes a targeted simulation study designed to examine the impact of model misspecification on the calibration of design features specified via the Generative Process Model that was introduced and applied in previous chapters. Specifically, the simulation examines how researchers' failure to identify a complete set of design features and the omission of those features from the specification of the GPM impact estimates of higher order features that would later be used for pre-calibration of generated items?

5.2 Simulation Design

5.2.1 Data Generation Approach

Data Generation Model. Given the challenges associated with estimating the parameters of the full Generative Process Model in the first set of simulation studies,

and particularly parameters in the upper levels of the model, the decision was made to use a simpler model in this simulation study.

The data generation model used for this study uses a Rasch model at Level 1, and the difficulty for item *i* can be written as

$$\beta_{i(j_1,j_2,j_3,j_4)t} = \gamma_{000} + \pi_{01t}X_1 + \pi_{02t}X_2 + \pi_{03t}X_3 + \pi_{04t}X_4 + +e_{i(j_1,j_2,j_3,j_4)t} + u_{0j_1t} + u_{0j_2t} + u_{0j_3t} + u_{0j_4t} + \gamma_{000} + v_{00t}.$$

Just as with the full GPM model, generated item difficulties are pre-calibrated based on the initial calibration and estimated mean difficulty of the item family. Each family mean is determined by the item form *t* from which it is generated and the particular combination of four design effects, X_1, X_2, X_3 and X_4. Item forms are treated as random instantiations of tasks that target the same sets of skills. This simplified version of the GPM is very similar to the LICM discussed in Chapter 2, except that Geerlings et al (2011) included a regression at Level 2 and in the GPM design features are included as cross-classified effects. template-level characteristics, the data generation model was simplified to a three-level hierarchical model with cross-classification at Level 2.

Simulation Conditions. In this simulation, three conditions were systematically varied: the number of features included in Level 2 of the analytic model, the within-family variance at Level 1, and the number of people simulated to encounter the items.

Either the analytic model matched the generation model, or one or two covariates were omitted from the analytic model, meaning that the analytic model would either correctly identify and estimate parameters for sixteen item families as

defined by the four features in the generation model, or it would be estimating parameters as if there were eight families defined by three features or four families as defined by only two features. This misspecification was accomplished through the re-indexing of the response data prior to estimation, a process which was confirmed by fitting an unconstrained version of the model to the data and ensuring the number of estimated parameters was appropriate for the proposed (though misspecified) structure. The analytic models considered in this study are shown in Table 18 below.

Table 18. Analytic Models Applied in Simulation Study

Design Features	Item Families Per Item Form	Item Difficulty
4	16	$\gamma_{000} + \pi_{010}X_1 + \pi_{020}X_2 + \pi_{030}X_3 + \pi_{040}X_4 + e_{i(j_1,j_2,j_3,j_4)t} + u_{0jt} + v_{00t}$
3	8	$\gamma_{000} + \pi_{010}X_1 + \pi_{020}X_2 + \pi_{030}X_3 + e_{i(j_1,j_2,j_3)t} + u_{0jt} + v_{00t}$
2	4	$\gamma_{000} + \pi_{010}X_1 + \pi_{020}X_2 + e_{i(j_1,j_2)t} + u_{0jt} + v_{00t}$

Echoing the previous study, the within-family variance was manipulated to be equal to either 0.1 or 0.4 in an effort to replicate a well-designed item generation process which produces items with very similar item properties, or a poorly designed process where items within the same family are not isomorphic. This range of values maps reasonably well onto values within the literature, and although thresholds have not been established for items that are treated as isomorphic versus those that are not, these levels induced sufficient variation in the previous study to impact the quality of parameter estimates and so the same levels were included here.

Finally, in an effort to begin to account for the process of pre-calibration, to include data collection as well as model selection and specification, the number of

simulees was included as a factor and set to either 45, 90, or 180, with responses from these simulees were randomly assigned to items. Using this sampling strategy, which is described in more detail below, the number of observations per person was approximately 45 observations with 45 simulees, 25 observations with 90 simulees, and 15 observations per with 180 simulees. The number of people was included as a factor instead of varying the number of observations per person directly so as not to increase or decrease the amount of observations per item across conditions, which is known to impact the quality of parameter estimates (Leucht, 2013; Sinharay & Johnson, 2008).

Together, these conditions were combined to yield 18 total cells, and 50 replications were completed per cell. These conditions are summarized in Table 19 below.

Table 19. Summary of Simulation Conditions

Factor	Levels	Number of Levels
Number of Specified Design Factors	2, 3, 4	3
Variance Across Items Within Family	0.1, 0.4	2
Number of Simulees	45, 90, 180	3
Number of conditions		18
Number of replications per condition		50

Response Data Generation. Response data were generated for each condition following a similar process to what was described in Chapter 4. People and items are generated separately. First, an ability vector, θ_p, was drawn according to a normal unit distribution for the designated number of *simulees* (45, 90, or 180). Second, a complete matrix of item difficulties was calculated consistent with the simplified Generative Process Model outlined above (Equation 5.1). Important to the AIG context*, items are generated as a population* rather than as a limited sample of items

so that they might be appropriately modeled as random. The total number of items generated for the simulation therefore far exceeds the number that is considered in any single replication or even in the study as a whole: a total of 160,000 unique items were generated. These items were derived from 5,000 simulated item forms, and four design features were manipulated on each of those forms to yield 16 prototypical items (80,000 total families). From each prototypical item, 20 instantiations were generated which differ from one another only in surface features.

The next step is to calculate a complete *response probability matrix* for every person-item combination, where the log odds of a correct response by person p to item i can be written as

$$\eta_{ip(j_1,j_2,j_3,j_4)t} = \theta_p - (\gamma_{000} + \pi_{010}X_1 + \pi_{020}X_2 + \pi_{030}X_3 + \pi_{040}X_4 + e_{i(j_1,j_2,j_3,j_4)t} + u_{0j_10} + u_{0j_20} + u_{0j_30} + u_{0j_40} + v_{000}).$$

For the purposes of this study, $\theta_p \sim N(0,1)$. Item forms have difficulties which are normally distributed with a mean of $\gamma_{000} = -1$ and a variance σ_v^2=1. Each of the design features are all binary, with the value of each coefficient defined following Dardick & Harring (2008) so that 80% of the variation at Level 2 is explained by the linear combination of those features,

$$\pi_{010} = \pi_{020} = \pi_{030} = \pi_{040} = 0.8.$$

As noted in the previous section, the within-family variation, is a simulation condition, and so σ_e^2 is either equal to .1 or .4.

Following the calculation of the log odds of a correct response for each person-item combination, the inverse logistic function is used to transform that probability matrix into a matrix of *dichotomous response data*, which each row

represents a complete response vector for each person. Response data are generated in this way once per condition to ensure consistency of response data within each condition (so that if a person "encounters" the same item more than once, the response vector is not inconsistent by chance), and random seeds were specified within the generation code to facilitate comparisons across conditions by minimizing sources of sampling variability.

Response data for each replication within each condition was assembled by drawing two samples: the first from among the available item pool, and the second from available simulees. For each replication, 30 templates were selected at random and without replacement from the 5,000 available. For every item derived from each of those templates, the responses from two simulees were selected at random from the response matrix. Each person could only encounter each item once, though no restrictions were placed on how many or which items each person might see that were derived from the same item form and/or shared common design features.

The response data used for each replication was therefore comprised of 1200 rows, with each row containing a unique identifier denoting the person, an identifier for the item, and a dichotomous score variable. Each response was also indexed by relevant features of the generation process: an identifier for item form and either a "1" or a "0" denoting the presence or absence of each of four design features. As noted above, a total of fifty replications were completed for each condition.

5.2.2 Estimation

Estimation was performed using RStan to facilitate the estimation of cross-classified fixed and random effects (Stan Development Team, 2015). Each model was

estimated using six chains with 5,000 burn-in iterations and 1,000 samples after warm-up. Each chain was initialized with random starting values. A non-centered parameterization was used when estimating the variances for each model. Half-normal priors were specified for each variance parameter that was estimated, with the upper bound of those priors estimated using half-normal *N(*0,.5) hyperpriors. In all cases, the ability parameter, θ_p is specified a normal variate, with a mean of 0 and standard deviation equal to 1. Following estimation, trace plots and sampling parameters for each chain were examined for convergence, in addition to monitoring both effective sample sizes and Rhat values for each parameter.

5.3 Results

5.3.1 Sampling Parameters

Conventional wisdom would suggest that models will have difficulty converging when the model is misspecified, but in this simulation study the model seemed to have the most difficulty under what were arguably the "best" conditions for item generation and pre-calibration, namely that the within-family variation between items was lowest. It was rare that chains reached a maximum treedepth during sampling, though this occurred in between two and four percent of transitions when the within-family variance was equal to 0.4 and there were 180 simulees interacting with only 15 items on average. There was no difference in the total time needed for estimation across the various conditions in this simulation study.

Table 20. Summary Sample Statistics by Condition

Condition	No. Covariates	σ_e	Students	Replications with Divergent Transitions		Max Treedepth Reached (%)	Total Run Time (s)
				%	Max		
1	4	0.1	45	74	267	0	1387
2	4	0.1	90	88	555	0	1450
3	4	0.1	180	88	443	0	1384
4	4	0.4	45	0	0	0	1347
5	4	0.4	90	0	0	0	1577
6	4	0.4	180	0	0	4	1859
7	3	0.1	45	78	158	0	1309
8	3	0.1	90	86	482	0	1388
9	3	0.1	180	82	218	0	1455
10	3	0.4	45	0	0	0	1373
11	3	0.4	90	0	0	0	1342
12	3	0.4	180	0	0	2	1600
13	2	0.1	45	74	160	0	1357
14	2	0.1	90	90	270	0	1418
15	2	0.1	180	88	346	0	1422
16	2	0.4	45	0	0	0	1219
17	2	0.4	90	0	0	0	1397
18	2	0.4	180	0	0	2	1384

The pattern of results in Table 21 tells a similar story: given the estimation parameters of this simulation study, the only parameter which had difficulty converging was the within-family variance. Interestingly, evidence of non-convergence is not diagnostic, in this case, of model misspecification. Instead, results suggest that this parameter was most difficult to estimate when the data generation process was well-designed, and when there were fewer simulees in the calibration sample, meaning that they were more likely to encounter items generated from the same templates or incorporating the same design features as compared to the other conditions. Difficulties estimating within-family variance when the templates were well-designed (and the integration of secondary content was simulated to induce minimal variation)

are also visible in the magnitude of the bias in the variance parameter estimates (Table 23).

5.3.2 Item Parameter Estimates

Table 23 and Table 24 show a pattern in estimate quality similar to that seen in the previous simulation work, parameter estimates are biased and show minimal variability across replications. As expected, the omission of design effects negatively impacts parameter estimates by inducing significant negative bias in the upper level mean estimates ($\widehat{\gamma_{000}}$). The omission of design features at Level 2 does not propagate upward or downward to negatively impact within-family variance estimates. The impact of fixed effect misspecification appears to be in the significant inflation of variance estimates also at Level 2.

An unexpected result is that the results suggest an interaction between the number of items each person responds to and the quality of pre-calibration estimates. Specficially, when the number of simulated examinees is lower, meaning those same people are interacting with more items which are derived from a shared template and share common design features, the negative bias in the parameter estimates is less pronounced. This may have implications for designing pre-calibration samples.

Table 21. Parameter Non-Convergence as a Percentage of Replications Per Condition

				Fixed Effects					Random Effects		
Condition	No. Covariates	σ_e	Students	$\widehat{\gamma_{000}}$	$\widehat{\pi_{01}}$	$\widehat{\pi_{02}}$	$\widehat{\pi_{03}}$	$\widehat{\pi_{04}}$	$\widehat{\sigma_e}$	$\widehat{\sigma_u}$	$\widehat{\sigma_v}$
1	4	0.1	45	0	0	0	0	0	12	0	0
2	4	0.1	90	0	0	0	0	0	2	0	0
3	4	0.1	180	0	0	0	0	0	0	0	0
4	4	0.4	45	0	0	0	0	0	0	0	0
5	4	0.4	90	0	0	0	0	0	0	0	0
6	4	0.4	180	0	0	0	0	0	0	0	0
7	3	0.1	45	0	0	0	0		10	0	0
8	3	0.1	90	0	0	0	0		6	0	0
9	3	0.1	180	0	0	0	0		0	0	0
10	3	0.4	45	0	0	0	0		0	0	0
11	3	0.4	90	0	0	0	0		0	0	0
12	3	0.4	180	0	0	0	0		0	0	0
13	2	0.1	45	0	0	0			0	0	0
14	2	0.1	90	0	0	0			2	0	0
15	2	0.1	180	0	0	0			2	0	0
16	2	0.4	45	0	0	0			0	0	0
17	2	0.4	90	0	0	0			0	0	0
18	2	0.4	180	0	0	0			0	0	0

Note: The shaded areas of the table indicate where no data is available.

Table 22. EAP Parameter Estimates by Condition, Median Values Across Replications

				Fixed Effects					Random Effects		
Condition	No. Covariates	σ_e	Students	$\widehat{\gamma_{000}}$	$\widehat{\pi_{01}}$	$\widehat{\pi_{02}}$	$\widehat{\pi_{03}}$	$\widehat{\pi_{04}}$	$\widehat{\sigma_e}$	$\widehat{\sigma_u}$	$\widehat{\sigma_v}$
1	4	0.1	45	-1.384	0.847	0.864	0.786	0.864	0.224	0.413	0.851
2	4	0.1	90	-1.195	0.82	0.847	0.783	0.852	0.192	0.403	0.833
3	4	0.1	180	-1.105	0.834	0.854	0.767	0.85	0.186	0.408	0.831
4	4	0.4	45	-1.382	0.839	0.893	0.802	0.859	0.632	0.414	0.848
5	4	0.4	90	-1.198	0.808	0.876	0.792	0.846	0.588	0.41	0.836
6	4	0.4	180	-1.115	0.831	0.871	0.78	0.854	0.618	0.41	0.833
7	3	0.1	45	-0.951	0.847	0.864	0.79		0.224	0.613	0.838
8	3	0.1	90	-0.772	0.818	0.845	0.785		0.181	0.606	0.823
9	3	0.1	180	-0.68	0.831	0.852	0.768		0.18	0.606	0.82
10	3	0.4	45	-0.957	0.839	0.891	0.803		0.63	0.607	0.834
11	3	0.4	90	-0.782	0.807	0.871	0.791		0.588	0.6	0.823
12	3	0.4	180	-0.69	0.828	0.871	0.783		0.614	0.601	0.822
13	2	0.1	45	-0.554	0.846	0.863			0.228	0.735	0.83
14	2	0.1	90	-0.382	0.817	0.846			0.173	0.722	0.816
15	2	0.1	180	-0.297	0.829	0.851			0.184	0.727	0.813
16	2	0.4	45	-0.557	0.838	0.888			0.627	0.736	0.825
17	2	0.4	90	-0.398	0.808	0.872			0.585	0.727	0.816
18	2	0.4	180	-0.301	0.828	0.87			0.61	0.726	0.817

Note: The shaded areas of the table indicate where no data is available.

Table 23. Median Relative Bias Across Replications by Simulation Condition

				Fixed Effects					Random Effects		
Condition	No. Covariates	σ_e	Students	$\widehat{\gamma_{000}}$	$\widehat{\pi_{01}}$	$\widehat{\pi_{02}}$	$\widehat{\pi_{03}}$	$\widehat{\pi_{04}}$	$\widehat{\sigma_e}$	$\widehat{\sigma_u}$	$\widehat{\sigma_v}$
1	4	0.1	45	38.1	-8.1	-6.9	-14.8	-6.3	-22.1	78.8	-15.1
2	4	0.1	90	20.4	-11.1	-8.3	-15.6	-8	-32.7	75.5	-16.5
3	4	0.1	180	11	-10.5	-7.5	-16.3	-8	-33.4	77	-16.9
4	4	0.4	45	39.1	-9.1	-4.1	-13.4	-6.7	-0.3	78.7	-15.5
5	4	0.4	90	20.8	-12.4	-5.9	-15.1	-8.7	-5.9	77.8	-16.7
6	4	0.4	180	12.1	-11.1	-5.4	-15.5	-8	-3	77.6	-16.8
7	3	0.1	45	-4.8	-8.2	-7	-14.6		-22.1	165.4	-16.1
8	3	0.1	90	-22.5	-11.2	-8.4	-15.5		-34.2	160.5	-17.6
9	3	0.1	180	-31.6	-10.6	-7.7	-16.3		-34.2	161.3	-17.8
10	3	0.4	45	-4	-9.2	-4.3	-13.4		-0.6	164.3	-16.4
11	3	0.4	90	-21.4	-12.4	-6.1	-15.1		-6	160.9	-17.7
12	3	0.4	180	-30.5	-11.2	-5.6	-15.4		-3.3	161.9	-17.8
13	2	0.1	45	-44.6	-8.2	-7.1			-24.1	218.9	-16.9
14	2	0.1	90	-61.4	-11.3	-8.5			-35.1	214	-18.4
15	2	0.1	180	-70.4	-10.6	-7.7			-34.7	213.8	-18.6
16	2	0.4	45	-44.1	-9.2	-4.4			-0.9	219.7	-17.3
17	2	0.4	90	-60.7	-12.5	-6.2			-6.3	214.9	-18.6
18	2	0.4	180	-69.6	-11.3	-5.7			-3.6	215.2	-18.6

Note: The shaded areas of the table indicate where no data is available.
Note: Relative bias was calculated for all parameters for each replication using the generating parameters and EAP estimates of those parameters. The median values within each condition are shown in this table.

Table 24. Empirical Variability Across Replications, by Simulation Condition

				Fixed Effects					Random Effects		
Condition	No. Covariates	σ_e	Students	$\widehat{\gamma_{000}}$	$\widehat{\pi_{01}}$	$\widehat{\pi_{02}}$	$\widehat{\pi_{03}}$	$\widehat{\pi_{04}}$	$\widehat{\sigma_e}$	$\widehat{\sigma_u}$	$\widehat{\sigma_v}$
1	4	0.1	45	0.046	0.038	0.037	0.04	0.034	0.099	0.022	0.02
2	4	0.1	90	0.042	0.035	0.038	0.037	0.043	0.095	0.024	0.023
3	4	0.1	180	0.049	0.042	0.037	0.037	0.039	0.094	0.026	0.022
4	4	0.4	45	0.053	0.035	0.042	0.034	0.039	0.068	0.026	0.025
5	4	0.4	90	0.06	0.037	0.045	0.047	0.047	0.086	0.027	0.031
6	4	0.4	180	0.047	0.044	0.034	0.038	0.034	0.078	0.026	0.023
7	3	0.1	45	0.043	0.038	0.037	0.039		0.098	0.02	0.019
8	3	0.1	90	0.033	0.035	0.039	0.038		0.094	0.026	0.023
9	3	0.1	180	0.042	0.042	0.037	0.038		0.096	0.025	0.022
10	3	0.4	45	0.039	0.036	0.043	0.034		0.069	0.024	0.026
11	3	0.4	90	0.043	0.037	0.045	0.047		0.085	0.03	0.029
12	3	0.4	180	0.044	0.044	0.035	0.038		0.079	0.024	0.024
13	2	0.1	45	0.037	0.039	0.038			0.098	0.022	0.02
14	2	0.1	90	0.031	0.036	0.039			0.092	0.024	0.024
15	2	0.1	180	0.034	0.042	0.038			0.09	0.025	0.022
16	2	0.4	45	0.033	0.036	0.043			0.067	0.024	0.025
17	2	0.4	90	0.036	0.037	0.045			0.087	0.031	0.03
18	2	0.4	180	0.038	0.045	0.035			0.079	0.024	0.024

Note: The shaded areas of the table indicate where no data is available.
Note: The empirical variability of the EAP estimates was calculated within each condition across r=50 replications.

Chapter 6: Discussion

6.1 Brief Summary

Increased processor speed and greater programming flexibility have made it possible to not only imagine but implement automated item generation systems. Realizing the promise of automatic item generation lies in successful engineering and careful design, not merely improved automation. Ideally, automatic item generation (AIG) processes engineer a precise alignment between elements represented in features of the cognitive task models, the structural and variable elements of item templates, and the structure of the corresponding psychometric models used in calibration. That alignment is critical to the *a priori* prediction of item properties.

Unfortunately, while full automation may be technologically feasible, the necessary alignment of cognitive, generative, and psychometric models continues to prove difficult to achieve in practice (e.g., Luecht, 2013). Careful engineering has not eliminated challenges faced by item writers in traditional development contexts, and improvements in technology have not resolved the challenges inherent in defining the set and range of template elements so that they have well-understood impacts on item properties (e.g., Irvine & Kyllonen, 2002; Luecht, 2013). The persistent difficulty in consistently predicting properties of generated items highlights the need for research into both appropriate model specification and the development of procedures for evaluating item quality.

This paper presents a new conceptual framework to facilitate the alignment of generative and psychometric models for the pre-calibration of automatically generated items. Using this multi-level framework with its combination of crossed

fixed and random effects to capture key components of the generative process, an analysis of response data collected from the online administration of automatically generated items yielded readily interpretable parameter estimates. Simulation results suggest, however, that while this model has the potential to support the direct assessment of the quality of the item generation process, more work has to be done in order to understand the conditions under which these models will yield parameter estimates that are sufficiently accurate to be diagnostic of the quality of the generation process.

Realizing the promise of pre-calibration is not something that will be achieved through more rapid computation or improved engineering. Perhaps the most important lesson learned through this work is that if we are serious about moving forward within the AIG arena, we're going to need to be proactive in our efforts to close conceptual gaps and synthesize existing research. We are going to need to investigate strategies to improve model estimation, particularly given the inherent and likely unavoidable tension between the desire for more flexible models and more elegantly engineered processes. We are going to need to interrogate our assumptions, and recognize that although some assumptions improve model estimability it may limit our ability to assess model suitability for use in applied contexts. And finally, we make fewer assumptions about what we believe success should look like. It is this result, out of everything presented here, that stands out to me most clearly: throughout the AIG literature there are a number of assertions as to what a well-designed item generation process looks like, it was within the well-designed

conditions that parameter estimates were the most bias and model performance was weakest.

6.2 Out of Scope but on the Horizon

Perhaps it is true of every ambitious project that it only begins to scratch the surface of what is possible. During the course of this research, I uncovered several open questions that could be pursued within the context of AIG and pre-calibration that were beyond the scope of this effort.

6.3.1 Extending the Generative Process Model

One extension of this work would be to investigate the model performance using a two-parameter logistic model as opposed to a Rasch model for calibration. When fixed effects models are being used for item calibration, the literature suggests that incomplete model specification, the presence of multiple populations, or examinees' utilization of alternative response strategies may result in poor model-data fit and failure to find significant effects of particular design manipulations. But where misspecifications might lead to systematic *variation* among instances generated from the same templates, this presents a very different problem than that of *covariation* among those instances. The simulation work presented in this paper suggests that variation may present only minimal problems for item calibration and ability estimation, consistent with previous simulation studies (e.g. Sinharay, Johnson, & Williamson, 2005). In contrast, covariation among item parameters that would result from misspecification in a 2PL context would present a fundamentally different problem for calibration that should not be ignored (Leucht, 2013).

In a small simulation study Luecht (2013) clearly highlighted, within a limited range of conditions, the potential impact of unmodeled residual covariances between item parameters within families using a 2PL model. Luecht did not specify the origin of this covariance between parameters otherwise unaccounted for in the model, but in a limited simulation study in which he varied test length (10 versus 40 items), estimation error of item parameters resulting from family-level calibration (none using the generating parameters, low, moderate, and high), and conditional covariance between discrimination and difficulty parameters (low, moderate, high), he demonstrated increased error and bias in ability estimates. Although increasing test length was shown to ameliorate the effects of the loss of efficiency resulting from family-level calibration in the absence of residual (level-1) covariance, simply increasing test length failed to address the bias that resulted from the presence of even low residual covariances when there was a high degree of within-family variability (Luecht, 2013).

With the exception of Luecht's (2013) limited investigation, simulation studies designed to demonstrate the promise of hierarchical models for use within an AIG context, with few exceptions (notably Geerlings, 2012; Geerlings et al., 2011) routinely set residual covariances equal to zero, with minimal variances on the diagonals that govern the variability of instances within item families. This is consistent with assumptions routinely made about homoscedasticity and the independence of residuals in the presentation or application of models proposed for use in the pre-calibration of AIG items. Many authors examining the performance of hierarchical models for use in an AIG context note that while the proposed models

can arguably accommodate non-zero covariances between parameters at level one (Embretson & Daniel, 2008) or at level two (Geerlings et al., 2011), they routinely make simplifying assumptions about covariances between item parameters within and across families. The challenge of estimability of some of these models, and the limitations of computation are real. That said, little attention has been paid to the modeling or accurate estimation of covariances between item parameters, which presents an exciting opportunity for continued research.

6.3.2 Examining Strategy Usage and Its Implications

One of the reasons that pre-calibration is so elusive might be because items don't function the same way for everyone in the population. Persistent (and frustrating) lack of model-data fit might be due to multiple strategy usage or the presence of differential item functioning among items within a family. Unfortunately, within the AIG literature, the cognitive models that are at the core of AIG system design, describe single-strategy solution processes; requiring system constraints to be implemented to ensure that respondents use the dominant strategy (Arendasy, 2006; 2007; Embretson, 1999).

Gorin's work on reading comprehension items illustrates the possible pitfalls of multiple strategy usage within an AIG context. In discussing the results of her analysis of response data from reading comprehension items Gorin (2005) noted that several factors that were theoretically relevant were found not to be significant. Experimental manipulations of reading comprehension items did not result in changes in psychometric properties of items as predicted. Non-results were attributed to the possible under-representation of study constructs, where the experimental

manipulations were not sufficient to produce the desired changes in response processes. She also noted, however, that the results might reflect "subgroup processing differences," in line with research suggesting that processing models for reading comprehension developed for specific populations may not generalize to individuals from other populations (Gorin, 2005, p. 368). She explained that "items generated with predictable properties based on psychological principals may not be equally valid for all examinees. *In such cases, multiple algorithms for item generation might be necessary to fit the various individual processing models*" (emphasis added, p. 368).

As is typical within the AIG literature, some researchers advocate for engineering a solution to any problem. When confronted with the possibility of multiple strategy usage, Arendasy and his colleagues (2006, 2007) underscored the importance of implementing functional constraints as part of any AIG framework. These constraints, or rules governing the selection of radicals and incidentals and permissible combinations of these are above and beyond the radicals and incidentals that are part of the cognitive model. Specifically, constraints need to be developed in order to ensure that "alternate solution strategies leading to differential item functioning are not supported by test material" (Arendasy et al., 2006, p. 3). But like most challenges in the realm of AIG, it is probably reasonable to be skeptical of any solution that simply consists of designing our way out of a problem.

Unfortunately, it might not be possible to eliminate the possibility of multiple-strategy items, even through careful engineering. Mislevy and Sheehan (2001) argued this point, saying that "GRE sentence-completion pools contain a fair number of

items that are solvable through two or more alternative solution strategies... both the current analyses and psycholinguistic literature support the notion that multiple-strategy items are likely to appear in verbal-reasoning item pools" (Mislevy & Sheehan, 2001, p. 29). They go on to say that developing items that permit the usage of multiple strategies is not inappropriate, but that "what is inappropriate is expecting the intended strategy for a given item to function as the only solution strategy, and creating models of student proficiency that completely ignore the phenomenon of multiple-strategy items" (Mislevy & Sheehan, 2001, p. 30). This assertion then begs the question: how to we create new and different models that do a better job of parameterizing the behaviors we care about? This is the question that sparked my interest in the very beginning.

Appendix A: Generation Code

```
#############################################################
##### Generation Code for Item Difficulties,
##### Cross-classification at Levels 2 and 3, Random Effects Only
##### Grand Mean of 0
##### Edited: 09/16/18
#############################################################

#install.packages("data.table")
#install.packages("reshape")
#install.packages("MASS")
#install.packages("plyr")
#install.packages("psych")
#install.packages("rje")
#install.packages("reshape2")

#############################################################
### LIBRARIES
#############################################################

library(data.table)
library(reshape)
library(MASS)
library(plyr)
library(psych)
library(rje)
library(reshape2)


set.seed(12345)

#############################################################
### INITIALIZATION
#############################################################

### Calculation of Coefficient Values
### Dardick & Harring 2012

coef_calc = function(ncov,r_sq,totvar,covvar){
  coef_val = sqrt((r_sq*totvar)/(covvar*ncov))
  return(coef_val)
}

### Definition of Simulation Parameters
### Number of Replications, Number of Simulees, Number of Templates,
Features, and Items

NR = 100                                    ### Number of Replications
NP = 250                                        ### Number of Simulees
NO = 75                                  ### Number of Observations per
Item (Equal to Average Per Item in Empirical Data)

NTP = 1000                               ### Number of Templates in the
population, Used to Generate Item Difficulties
```

```
NT = 3         ### Number of Groups at Level 4, Selected Randomly
Each Replication

NC = c(2,2)                              ### Number of Binary Covariates
at (Level 2, Level 3) Describing Feature Manipulations
NK = 2**NC[2]   ### Number of Item Models per Template at Level 3,
NK*NT = Total Number of Item Models at Level 3
NJ = 2**NC[1]### Number of Families per Item Model, NJ*NK*NT = Total
Number of Families at Level 2
NI = 10                                  ### Number of Items per Family,
NI*NJ*NK*NT = Total Number of Items

### Definition of Random Effects
### Variance Explained and Unexplained, and Coefficients

vec = c(1,0)




                                                ### Levels of Feature
Manipulation per Feature/Covariate
lst_2 = lapply(numeric(NC[1]),function(x) vec)        ### Function to
Describe Feature Combinations at Level 2
lst_3 = lapply(numeric(NC[2]),function(x) vec)  ### Function to
Describe Feature Combinations at Level 3

CV = .25                                    ### Variance of each Binary
Covariate (Specified as Independent, no Covariance)
R2 = c(.75,.75) ### Variance Explained at (Level 2, Level 3) by
Manipulated Features

var_total = c(.1,.6,.8,1)




                                              ### Total Variance TO BE
EXPLAINED at each level of the model
e_sigma = sqrt(var_total[1])



                                           ### Within-family variation,
Sigma defined for rnorm() function
u_sigma = sqrt((var_total[2]-R2*var_total[2])/NC[1])  ### Unexplained
Variance at Level 2, Distributed Equally Across Features, Sigma
Defined for rnorm() function
v_sigma = sqrt((var_total[3]-R2*var_total[3])/NC[2])  ### Unexplained
Variance at Level 3, Distributed Equally Across Features, Sigma
Defined for rnorm() function
```

```
w_sigma = sqrt(var_total[4])



                                        ### Random Effects at Level 4

c_2 = as.matrix(expand.grid(lst_2))



                                                ### Matrix of Feature
Combinations for Level 2
c_3 = as.matrix(expand.grid(lst_3))



                                                ### Matrix of Feature
Combinations for Level 3

b_2 = coef_calc(NC[1],R2[1],var_total[2],CV)                    ###
Coefficients at Level 2

b_3 = coef_calc(NC[2],R2[2],var_total[3],CV)                    ###
Coefficients at Level 3
b_3 = -1*b_3




                                      ### Coefficients at Level 3 are
negative, positive at level 2

############################################################
### ITEM GENERATION FUNCTION
### Random Effects Only, Grand Mean = 0
############################################################

### In the 'itemgen' function, difficulties are being generated
### for the full population
### Items observed from specific templates will be selected at a
later stage

itemgen =
function(sige,sigu,sigv,sigw,nc,nt,nk,nj,ni,cc2,cc3,b2,b3){

  ### Storage Vectors

  g0 = rep(0,nt)
  ww = rep(0,nt)
  p0 = rep(0,nk)
  f3 = matrix(0,nrow=nk,ncol=nc[2])### fixed effects at level 3
  vv = matrix(0,nrow=nk,ncol=nc[2])### random effects at level 3
  m0 = rep(0,nj)
```

```
  f2 = matrix(0,nrow=nj,ncol=nc[1])### fixed effects at level 2
  uu = matrix(0,nrow=nj,ncol=nc[1])### random effects at level 2
  ee = rep(0,ni)
  betai = rep(0,ni)

  tempdiff = matrix(0,nrow=ni*nj*nk*nt,ncol=1)    ### Storage for
Template Difficulties
  gendiff = matrix(0,nrow=ni*nj*nk*nt,ncol=1)    ### Storage for Item
Difficulties
  cxmat2 = matrix(0,nrow=ni*nj*nk*nt,ncol=nc[1])      ### Storage for
Level 2 Features
  cxmat3 = matrix(0,nrow=ni*nj*nk*nt,ncol=nc[2])   ### Storage for
Level 3 Features

  ### Grand Mean

  g00 = 0

  ### Templates
  for (t in 1:nt){
    ww[t] = rnorm(1,0,sigw)
    g0[t] = g00 + ww[t]

    ### Item Models
    for (k in 1:nk){

      ### Across Features Defining Item Models within Templates
      ### Matrix of random effects
      ### Matrix of fixed effects, product of covariate and 0/1
indicator
      for (n in 1:nc[2]){
        vv[k,n] = rnorm(1,0,sigv)
        f3[k,n] = b3*cc3[k,n]
      }

      p0[k] = g0[t] + sum(f3[k,]) + sum(vv[k,])

      ### Families
      for (j in 1:nj){

        ### Across Features Defining Families within Item Models
        ### Matrix of random effects
        ### Matrix of fixed effects, product of covariate and 0/1
indicator
        for (m in 1:nc[1]){
          uu[j,m] = rnorm(1,0,sigu)
          f2[j,m] = b2*cc2[j,m]
        }

        m0[j] = p0[k] + sum(f2[j,]) + sum(uu[j,])

        ### Items
        for (i in 1:ni){
          ee[i] = rnorm(1,0,sige)
          betai[i] = m0[j] + ee[i]
        }
```

```
        ### Saving features and generated difficulties for each
family of items

        is = (t-1)*nk*nj*ni + (k-1)*nj*ni + (j-1)*ni + 1
        ie = (t-1)*nk*nj*ni + (k-1)*nj*ni + j*ni

        gendiff[is:ie,1] = betai
        tempdiff[is:ie,1] = ww[t]

        for (xx in is:ie){
          cxmat2[xx,] = cc2[j,]
          cxmat3[xx,] = cc3[k,]
        }
      }
    }
  }

  ### Indices for each level of the model

  uid = seq(1:(ni*nj*nk*nt))


                                        ### Unique Identifier for Items
  tid = sort(rep(seq(1:nt),ni*nj*nk))

                                          ### Identifier for Templates
(Level 4)
  mid = rep(sort(rep(seq(1:nk),ni*nj)),nt)    ### Identifier for Item
Models w/in Template  (Level 3)
  fid = rep(sort(rep(seq(1:nj),ni)),nk*nt)    ### Identifier for Item
Families w/in Item Model(Level 2)
  iid = rep(seq(1:ni),nj*nk*nt)

                                         ### Identifier for Items w/in
Families        (Level 1)

  mydat=cbind(uid,tid,tempdiff,mid,cxmat3,fid,cxmat2,iid,gendiff)
  return(mydat)
}

############################################################
### ITEM PARAMETER ESTIMATION (FUNCTION CALL)
############################################################

set.seed(0911)

### Function Call

itemdiff =
itemgen(e_sigma,u_sigma,v_sigma,w_sigma,NC,NTP,NK,NJ,NI,c_2,c_3,b_2,
b_3)
itemdiff = as.data.frame(itemdiff)

### Generating Names for Data Frame, Given Number of Covariates at
Levels 2 and 3
```

```
jnum = seq(1:NC[1])
jname = rep(0,NC[1])
for (m in 1:NC[1]){
  jname[m] = paste('c',jnum[m],'2',sep="")
}

knum = seq(1:NC[2])
kname = rep(0,NC[2])
for (n in 1:NC[2]){
  kname[n] = paste('c',knum[n],'3',sep="")
}

names(itemdiff) =
c("uid","tid","t_mu","mid",kname,"fid",jname,"iid","diff")

############################################################
### ITEM GENERATION CHECK
############################################################

### The grand mean is set to 0.
### At level 3, each fixed effect is equal to 0.8660254
### At level 2, each fixed effect is equal to 0.8660254

### At level 4, the variance is equal to .4
### At level 3 (with four covariates, as written), there are four
variance components, each is equal to 1/16 = .0625.
### At level 2 (with four covariates, as written), there are four
variance components, each is equal to 1/16 = .0625.
### At level 1, the variance is equal to .1.

#summary(lmer(diff ~ 1 + c12 + c22 + c32 + c42 + c13 + c23 + c33 +
c43 + (1|tid) + (1|mid:tid) +(1|fid:(mid:tid)), data=itemdiff))

############################################################
### PERSON DATA
############################################################

### Generate a vector of person abilities indexed by a person id

set.seed(0912)

peeps = seq(1:NP)
thetaj = rnorm(NP,0,1)

############################################################
### SCORE DATA
############################################################

### Generate a complete matrix of response probabilities for all
person-item combinations
### Generate a complete matrix of scores for all person-item
combinations

probmat = matrix(0,nrow=(NT*NK*NJ*NI),ncol=NP)
scoremat = matrix(0,nrow=(NT*NK*NJ*NI),ncol=NP)
```

```
### loop over items

for (i in 1:(NT*NK*NJ*NI)){

  ### loop over people

  for (j in 1:NP){

    probmat[i,j] = expit(thetaj[j] - itemdiff$diff[i])
    scoremat[i,j] = as.numeric(rbinom(1,1,probmat[i,j]))

  }
}

### Transform the matrix of scores into a 3-column data set
### With scores indexed by the item number (unique id across all
items, uid)
### and the person (peeps)

scoredat = reshape2::melt(scoremat)
names(scoredat) = c('newuid','peeps','score')

#########################################################
#### SCORE DATA FOR ANALYSIS
#########################################################

#datafolder =
"C:\\Users\\SW\\Dropbox\\EmpiricalDiss\\Ch4_Sim\\Condition1\\Respons
eData\\"
#datafolder =
"C:\\Users\\SW\\Dropbox\\EmpiricalDiss\\Ch4_Sim\\LargeSample\\"
#datafolder  = "~/ResponseData/"

datafolder =
"G:\\Dropbox\\EmpiricalDiss\\Results\\ResponseData_Mod1_Cond1\\"

set.seed(0916)

### Loop over Replications

for (r in 1:NR){

  ### Sample Simulees Responding to Each Item
  ### Generate Observation Index for Items
  ### Sample Drawn Across All Items, No Constraints by
Template/Model/Family

  for (i in 1:(NI*NJ*NK*NT)){
    curr_item = i
    item_sample = sample(peeps,NO)

    if (curr_item==1){
      sample_vector = item_sample
      obs_index = rep(curr_item,NO)
```

```
    }

    else {
      sample_vector<-
append(sample_vector,item_sample,after=(NO*(curr_item-1)))
      obs_index<- append(obs_index,rep(curr_item,times =
NO),after=(NO*(curr_item-1)))
    }

    sample_vector
    obs_index
  }


  ### Create Sample Matrix as a Data Frame
  ### Simulees for Each Item Indexed by the Item Number

  sample_data = as.data.frame(cbind(obs_index,sample_vector))
  names(sample_data) = c('newuid','peeps')

  ### Select NT Templates per Replication

  temp_sample = sample(seq(1:NTP),NT)
  sample_itemdiff = itemdiff[itemdiff$tid %in% temp_sample,]
  sample_itemdiff = sample_itemdiff[order(sample_itemdiff$tid),]
  sample_itemdiff$newuid = seq(1:(NI*NJ*NK*NT))

  ### Merge Sample Matrix with Sampled Item Parameter Matrix

  mydat<-merge(sample_itemdiff,sample_data,by="newuid",all=TRUE)

  ### Remove Additional Objects in Large Sample Cases
  ###rm(itemgen)
  ###rm(probmat)
  ###rm(scoremat)
  ###rm(sample_data)
  ###rm(sample_itemdiff)
  ###rm(sample_vector)

  mydat = mydat[with(mydat,order(mydat$peeps)),]

  ### For Large Sample Demonstrations, Need to Reduce Size of Files
for Merge

  #datafolder =
"G:\\Dropbox\\EmpiricalDiss\\Ch4_Sim\\LargeSample\\ScoreFiles_1\\"

  #for(tt in 1:NT){

  #                 newdat = mydat[mydat$tid == tt,]

  #                   min_uid = min(newdat$newuid)
  #                   max_uid = max(newdat$newuid)

  #                   uid_vec = c(min_uid:max_uid)
  #        newscore = scoredat[scoredat$newuid %in% uid_vec,]
```

```
  #                              alldat <-
merge(newdat,newscore,by=c("newuid","peeps"),all.x=TRUE)

  #  myfile = paste0(datafolder,"cond1_tempnum_",tt,"_respfile.csv")
  #
               write.csv(alldat,myfile,row.names=FALSE)

  #                             rm(newdat)
  #                            rm(newscore)
  #                             rm(alldat)

  #}

  #file.list = list.files("~/ResponseData/")

  #fulldata = do.call("rbind",lapply(file.list,FUN = function(file){
  #               read.table(file,header=TRUE,sep=",")
  #                                 }))

  #saveRDS(fulldata,"LargeSampleFile_WellGen.rds")

  ### Merge Sample Matrix with Score Matrix

  mydat = mydat[with(mydat,order(mydat$peeps)),]
                                        ### First sorted by Peeps, then
by NEWUID

  mydat <- merge(mydat,scoredat,by=c("newuid","peeps"),all.x=TRUE)

  ### Save Files

  myfile = paste0(datafolder,"sim_cond1_",r,"_respfile.csv")
  write.csv(mydat,myfile,row.names=FALSE)

}
```

Appendix B: Stan Code for Model Estimation

```
##################################################
## Parameter Recovery study -
## Heterogeneous Means
## Heterogeneous Variances
## Code to loop through subset of replicates, save to RDS
##################################################

########################################
### LOAD LIBRARY, SET OPTIONS
########################################

library('rstan')

rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())

set.seed(12345)

########################################
### STAN MODEL INITIATION
########################################

mymodel<- "

data {
int<lower=1> N;            // number of observations

int<lower=1> J;            // number of students
int<lower=1> K;            // number of items
int<lower=1> P;            // number of families
int<lower=1> M;           // number of item models
int<lower=1> Q;                              // number of templates

int<lower=1> X;            // length of multten array
int<lower=1> Y;            // length of threedig array

int<lower=1> A;            // length of vv array
int<lower=1> B;            // length of hh array

int peeps[N];                  // student giving response n
int<lower=1,upper=K> item[N];  // item for response n
int<lower=0,upper=1> score[N];   // correctness for response n

int<lower=0,upper=J> pid[J];   // Person ID number
int<lower=0,upper=K> iid[K];   // Item ID number
int<lower=0,upper=P> fid[P];   // family ID number
int<lower=0,upper=M> mid[M];   // item model ID number
int<lower=0,upper=Q> tid[Q];   // template ID number

int<lower=1,upper=P> parent[K];           //indexes items to families
int<lower=1,upper=M> mm[P];                //indexes families to item
models
```

```
int<lower=1,upper=Q> tt[M];        //indexes item models to
templates

int multten[X];                  // Array of indices for families -
numbers are some multiple of ten
int threedig[Y];                 // Array of indices for families -
numbers are maximum of three digits

int vv[A];//Array of indices for imodels - display format is verbal
int hh[B]; //Array of indices for imodels - display format is
horizontal

}

parameters {

vector[J] uj;

vector <lower=0> [P] sigma_item;
vector <lower=0> [M] fam_resid;
vector <lower=0> [Q] mod_resid;

vector [K] betai_offset;
vector [P] fammu_offset;
vector [M] modmu_offset;

vector[Q] template_mu;
vector[Q] disp_horiz;
vector[Q] disp_verb;
vector[M] char_m10;                    //fixed effects of content
characteristics
vector[M] char_d3;                     //fixed effects of content
characteristics

}

transformed parameters{
vector[N] eta;
vector[K] betai;
vector[P] family_mu;
vector[M] model_mu;

// varying item family difficulty across item families within models
// decomposition of family means into a model mean and fixed effects

model_mu = template_mu[tt] + modmu_offset[tt] .* mod_resid[tt];
model_mu[vv] = model_mu[vv] + disp_verb[tt[hh]];
model_mu[hh] = model_mu[hh] + disp_horiz[tt[vv]];

// varying item family diffculty across item families within models
// decomposition of family means into a model mean and fixed effects

family_mu = model_mu[mm] + fammu_offset[mm] .* fam_resid[mm];
family_mu[multten] = family_mu[multten] + char_m10[mm[multten]];
family_mu[threedig] = family_mu[threedig] + char_d3[mm[threedig]];
```

```
// item difficulties parameterized as random, with parent-specific
means
betai = family_mu[parent] + betai_offset[parent] .*
sigma_item[parent];

//log odds of a correct probability
eta = uj[peeps]-betai[item];

}

model {

//hyperprior
betai_offset ~ normal(0,1);
fammu_offset ~ normal(0,1);
modmu_offset ~ normal(0,1);

//prior on random variable theta to scale the item difficulty
parameters
uj ~ normal(0,1);

//weakly informative prior on item variance
sigma_item ~ normal(0,1);
fam_resid ~ normal(0,1);
mod_resid ~ normal(0,1);

//likelihood function
score ~ bernoulli_logit(eta);

}
"

my_stan_code <- stanc(model_code=mymodel)
my_compiled_model <- stan_model(stanc_ret =
my_stan_code,verbose=FALSE)

########################################
### LOOP THROUGH FILES, ESTIMATE, SAVE RDS
########################################

ResultsFolder <- "~/Results/"
DataFolder <- "~/ResponseData/"

mystan <- sapply(1:1, FUN = function(r) {

  DataFile<-paste0(DataFolder,"pr_homfe_hetre_",r,"_respfile.csv")
  #DataFile<-
"C:\\Users\\SW\\Dropbox\\EmpiricalDiss\\Ch4_PR\\HomFE_HetRE\\Respons
eData\\pr_hetfe_hetre_1_respfile.csv"
  dd<-read.csv(DataFile,header=TRUE)

############################
### Transform Data into a List
############################
```

```
  #names(dd) <-
c("obs","peeps","item","family","imodel","template","verbal","horiz"
,"m10","d3","score")

N <- length(unique(dd$obs))
K <- length(unique(dd$item))
J <- length(unique(dd$peeps))
P <- length(unique(dd$family))
M <- length(unique(dd$imodel))
Q <- length(unique(dd$template))

peeps <- dd$peeps
item <- dd$item
score <- dd$score

### People ID - Length J
pid <- sort(unique(dd[c("peeps")])[,"peeps"])
### Item ID - Length K
iid <- sort(unique(dd[c("item")])[,"item"])
### Family ID - Length P
fid <- sort(unique(dd[c("family")])[,"family"])
### Item Model ID - Length M
mid <- sort(unique(dd[c("imodel")])[,"imodel"])
### Template ID - Length T
tid <- sort(unique(dd[c("template")])[,"template"])

### Item - Family Index - Length K, P Unique Values
### For each row in parent[], it returns the family ID number
dd<-dd[with(dd,order(item)),]
parent <- unique(dd[,c("item","family")])[,"family"]
### Family - Item Model Index - Length P, M Unique Values
### For each row in mm[], it returns the model ID number
dd<-dd[with(dd,order(family)),]
mm <- unique(dd[c("family","imodel")])[,"imodel"]
### Item Model - Template Index - Length M, T Unique Values
dd<-dd[with(dd,order(imodel)),]
tt <- unique(dd[c("imodel","template")])[,"template"]

### Content Characteristics
### For each row denoting a family, are the numbers multiples of 10
or are they three digits
### Array of indices

dd<-dd[with(dd,order(family)),]

multten<-unique(dd[dd$m10==1,c("family")])
threedig<-unique(dd[dd$d3==1,c("family")])

X<-length(multten)
Y<-length(threedig)

### Form Characteristics
### For each row denoting a template, is the form a verbal
representation or is it arranged horizontally
### Base category is comprised of items displayed in numeric format,
numbers arranged
```

```
### array of indices

dd<-dd[with(dd,order(imodel)),]

vv<-unique(dd[dd$verbal==1,c("imodel")])
hh<-unique(dd[dd$horiz==1,c("imodel")])

A<-length(vv)
B<-length(hh)

#########################
### Free Space
#########################

rm(dd)

#########################
### Estimate Model
#########################

myfit <-sampling(my_compiled_model,
              data = c("N","J","K","P","M","Q",
                       "A","B","X","Y",
                       "peeps","item","pid",
                       "iid","fid","mid","tid",
                       "parent","score","mm","tt",
                       "multten","threedig",
                       "vv","hh"),
              pars =
c("uj","sigma_item","fam_resid","mod_resid","template_mu","char_m10"
,"char_d3","disp_verb","disp_horiz"),
              control = list(stepsize=0.1, adapt_delta=0.9,
max_treedepth=15),
              iter = 5000, warmup=2500, chains = 4, thin=1,
save_warmup=TRUE,
              verbose = T)


myresultsfile =
paste0(ResultsFolder,"HomFE_HetRE_woffset_nprior_pt9_indexed_Rep",r,
".rds")
saveRDS(myfit, myresultsfile)

#########################
### Free Space
#########################

rm("N","J","K","P","M","Q",
     "A","B","X","Y",
     "peeps","item","pid",
     "iid","fid","mid","tid",
     "parent","score","mm","tt",
     "multten","threedig",
     "vv","hh")

rm(myfit)
```

```
#########################
### Bookkeeping
#########################

mytime <- Sys.time()
logentry <- (paste("Replication",r,"completed at",mytime))

print(logentry)

gc()
gc()

})
```

References

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

Arendasy, M., Sommer, M., Gittler, G., & Hergovich, A. (2006). Automatic generation of quantitative reasoning items: A pilot study. Journal of Individual Differences, 27(1), 2-14.

Asparouhov, T. & Muthen, B. (2012). General random effect latent variable modeling: Random subjects, items, contexts, and parameters. Technical report.

Bejar, I. I., Lawless, R. R., Morley, M. E., Wagner, M. E., Bennett, R. E., & Revuelta, J. (2002). A feasibility study of on-the-fly item generation in adaptive testing. Journal of Technology, Learning, and Assessment, 2(3).

Bradlow, E. T., Wainer, H., & Wang, X. (1999). A Bayesian random effects model for testlets. Psychometrika, 64(2), 153-168.

Brown, J. C., Frishkoff, G. A., & Eskenazi, M. (2005). Automatic question generation for vocabulary assessment. Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05.

Chen, C. & Wang, W. (2007). Effects of ignoring item interaction on item parameter estimation and detection of interacting items. Applied Psychological Measurement, 31(5), 388-411.

Cho, S.J., De Boeck, P., Embretson, S., & Rabe-Hesketh, S. (2013). Additive multilevel item structure models with random residuals: Item modeling for explanation and item generation. Pschometrika, 79, 84-104.

Cohen, J. Chan, Z., Jiang, T., & Seburn, M. (2008). Consistent estimation of Rasch item parameters and their standard errors under complex sample designs. Applied Psychological Measurement, 32, 289-310.

De Boeck, P. & Leuven, K.U. (2008). Random Item IRT models. Psychometrika, 73(4), 533-559.

de Leeuw, J., & Krefl, I.G.G. (1986). Random coefficient models for multilevel analysis. Journal of Educational and Behavioral Statistics, 11, 57-86.

Deane, P., Graf, E.A., Higgins, D., Futagi, Y., Lawless, R. (2006). Model Analysis and Model Creation: Capturing the Task-Model Structure of Quantitative Item Domains (ETS Research Report No. RR-06-11). Princeton, NJ: Educational Testing Service.

Embretson, S. E. & Daniel, R. C. (2008). Understanding and quantifying cognitive complexity level in mathematical problem solving items. Psychological Science Quarterly, 50(3), 328-344.

Embretson, S. E. & Reise, S. P. (2000). Item response theory for psychologists. Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

Embretson, S. E. (1998). A cognitive design system approach to generating valid tests: Application to abstract reasoning. Psychological Methods, 3, 300-396.

Embretson, S. E. (1999). Generating items during testing: Psychometric issues and models. Psychometrika, 64, 407-433.

Embretson, S.E. & Daniel, R. C. (2008). Understanding and quantifying cognitive complexity level in mathematical problem solving items. Psychological Science Quarterly, 50(3), 328-344.

Enright, K., Morely, M., & Sheehan, K. (2002). Items by design: The impact of systematic feature variation on item statistical characteristics. Applied Measurement in Education, 15(1), 49-74.

Frederickx, S., Tuerlinkckx, F., De Boeck, P., & Magis, D. (2010). RIM: A random item mixture model to detect differential item functioning. Journal of Educational Measurement, 47(4), 432-457.

Geerlings, H., Glas, C.A.W., & van der Linden, W.J. (2011). Modeling rule-based item generation. Psychometrika, 76(2), 337-359.

Gelman, A., & Hill, J. (2007). Data analysis using regression and multilevel/hierarchical models. Cambridge: Cambridge University Press.

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

Gierl, M. J., & Lai, H. (2012). Using item models for automatic item generation. International Journal of Testing, 12, 273-298.

Gierl, M. J., & Lai, H. (2013). Evaluating the quality of medical multiple-choice items created with automated generation processes. Medical Education, 47, 726-733.

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

Gierl, M. J., Lai, H., & Turner, S. R. (2012). Using automatic item generation to create multiple-choice test items. Medical Education, 46(8), 757-765.

Gitomer, D. H. & Bennett, R. E. (2002). Unmasking constructs through technology, measurement theory, and cognitive science (ETS Research Memorandum No. RM-02-01). Princeton, NJ: Educational Testing Service.

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

Goldstein, H. and Browne, W.J. (2005). Multilevel factor analysis models for continuous and discrete data. In Maydeu-Olivares, A and McArdle, J.J. (Eds.), Contemporary psychometrics: A festschrift for Roderick P. McDonald (pp. 453-475). Lawrence Erlbaum, New Jersey. (TI 11, GS 25).

Goldstein, H., & Goldstein, H. (1995). Multilevel statistical models (2nd ed.). London: E. Arnold.

Gorin, J. S. (2005). Manipulation of processing difficulty on reading comprehension test questions: The feasibility of verbal item generation. Journal of Educational Measurement, 42, 351-376.

Gorin, J.S. (2006). Test design with cognition in mind. Educational Measurement: Issues and Practice,25(4), 21-35.

Graf, E. A., Peterson, S., Steffen, M., & Lawless, R. (2005). Psychometric and cognitive analysis as a basis for the design and revision of quantitative item models (ETS Research Report No. RR-05-25). Princeton, NJ: Educational Testing Service.

Hagopian, J. (2014). More than a score: The new uprising against high-stakes testing. Chicago, IL: Haymarket Books.

Huff, K., C.B. Alves, J. Pelligrino & P. Kiliski (2013). Using evidence-centered design task models in automatic item generation. In M.J. Gierl & T. Haladyna (Eds.) Automatic item generation: Theory and practice. New York: Routledge.

Ip, E. H. (2002). Locally dependent latent trait model and the Dutch identity revisited. Psychometrika, 67, 367-386.

Ip, E.H., Smits, D.J.M., & De Boeck, P. (2009). Locally dependent linear logistic test model with person covariates. Applied Psychological Measurement, 33, 555-569.

Irvine, S.H., & Kyllonen, P.C. (Eds.) (2002). Item generation for test development. New York: Routledge.

Janssen, R., Tuerlinckx, F., Meulders, M., & De Boeck, P. (2000). A hierarchical IRT model for criterion- referenced measurement. Journal of Educational and Behavioral Statistics, 25, 285–306.

Jiao, H., Kamata, A., & Binici, S. (2010a, July).The effects of ignoring item and person clustering on ability estimation and proficiency classification across years. Paper presented at the conference of the Psychometric Society, Athens, GA

Jiao, H., Kamata, A., Wang, S., & Jin, Y. (2012). A multilevel testlet model for dual dependence. Journal of Educational Measurement, 49(1), 82-100.

Jiao, H., Kamata, A., Wang, S., & Jin, Y. (2010b, April). Simultaneous modeling of item and person dependence using a multilevel Rasch measurement model. Paper presented at the meeting of the American Educational Research Association, Denver, CO.

Jiao, H., Mislevy, R., & Zhang, Y. (2011, April). A general framework for clustering effects in IRT modeling. Paper presented at the meeting of the American Educational Research Association, New Orleans, LA.

Jiao, H., Wang, S., & Kamata, A. (2005). Modeling local item dependence with the hierarchical generalized linear model. Journal of Applied Measurement, 6(3), 311-321.

Kellogg, M., Rauch, S., Leather, R., Simpson, M.A., Lines, D., & Bickel L. (2015, April). Construction of a Dynamic Item Generator for K-12 Mathematics. Paper presented at the Annual Meeting of the National Council on Measurement in Education, Chicago, IL.

Kuhn, T. S. (1962). The structure of scientific revolutions. Chicago: University of Chicago Press.

Li, Y., Bolt, D. M., & Fu, J. (2006). A comparison of alternative models for testlets. Applied Psychological Measurement, 30(1), 3-21.

Liu, C., Wang, C., & Gao, Z. (2005). Using lexical constraints to enhance the quality of computer-generated multiple-choice cloze items. Computational Linguistics and Chinese Language Processing,10(3), 303-328.

Liu, C., Wang, C., & Gao, Z. (2005). Using lexical constraints to enhance the quality of computer-generated multiple-choice cloze items. Computational Linguistics and Chinese Language Processing,10(3), 303-328.

Luecht, R.M. (2013). An introduction to assessment engineering for automatic item generation. In M.J. Gierl & T.M. Haladyna (Eds.), Automatic item generation: Theory and practice (pp. 59-76). New York: Routledge.

McArdle, J. J. (1994). Structural factor analysis experiments with incomplete data. Multivariate Behavioral Research, 29(4), 409-454.

Mislevy, R.J., Levy, R., Kroopnick, M. & Rutstein, D. (2008). Evidentiary foundations of mixture item response theory models. In G.R. Hancock & K.M. Samuelsen (Eds.) Advances in latent variable mixture models (pp. 149-175). Charlotte: Information Age Publishing, Inc.

Mislevy, R., & Riconscente, M. (2005). Evidence-centered assessment design: Layers, structures, and terminology (PADI Technical Report 9). Menlo Park, CA: SRI International.

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

Mislevy, R. J. & Verhelst, N. (1990). Modeling item responses when different subjects employ different solution strategies. Psychometrika, 55(2), 195-215.

Mislevy, R.J., & Zwick, R. (2012). Scaling, linking, and reporting in a periodic assessment system. Journal of Educational Measurement, 49(2), 148-166.

Muthén, B. & Asparouhov, T. (2013). Item response modeling in Mplus: A multi-dimensional, multi- level, and multi-timepoint example. In Linden & Hambleton (Eds.) Handbook of item response theory: models, statistical tools, and applications (Forthcoming).

Muthén, B. (1989). Latent variable modeling in heterogeneous populations, Psychometrika, 54, 557-585.

Muthén, B. (1990). Mean and covariance structure analysis of hierarchical data. Paper presented at the Psychometric Society meeting in Princeton, NJ. (UCLA Statistics Series #62, August 1990.)

Rasbash, J., & Browne, W. J. (2008). Non-hierarchical multilevel models. In J. De Leeuw & E. E Meijer (Eds.) Handbook of multilevel analysis (pp. 1-38). New York: Springer.

Rasbash, J., & Goldstein, H. (1994). Efficient analysis of mixed hierarchical and cross-classified random structures using a multilevel model. Journal of Educational and Behavioral Statistics, 19(4), 337-350.

Reise, S.P., Ventura, J., Nuechterlein, K. H., & Kim, K. H. (2005). An illustration of multilevel factor analysis. Journal of Personality Assessment, 84(2), 126-136.

Rijmen, F. (2010). Formal relations and an empirical comparison among the bi-factor, the testlet, and a second-order multidimensional IRT model. Journal of Educational Measurement, 47(3), 361-372.

Rijmen, F., Tuerlinckx, F. De Boeck, P., & Kuppens, P. (2003). A nonlinear mixed model framework for item response theory. Psychological Methods, 8(2), 185-205.

Rost, J. (1990). Rasch models in latent classes: an integration of two approaches to item analysis. Applied Psychological Measurement, 14(3), 271-282.

Rupp, A. A. & Mislevy, R.J. (2006). Cognitive foundations of structured item response models estimation of the parameters in an item-cloning model for adaptive testing. In J.P. Leighton & M.J. Gierl (Eds.), Cognitive diagnostic assessment: Theories and applications. Cambridge: Cambridge University Press.

Rupp, A. A., diCerbo, K. E., Levy, R., Benson, M., Sweet, S., Crawford, A., Fay, D., Kunze, K. L., Calico, T., & Behrens, J. (2012). Putting ECD into practice: The interplay of theory and data in evidence models within a digital learning environment. Journal of Educational Data Mining, 4, 49-110.

Sheenhan, K. M. & Mislevy, R. J. (2001). An inquiry into the nature of the sentence completion task: implications for item generation (GRE Board Report No. 95-17bP). Princeton, NJ: Educational Testing Service.

Simpson, M.A., Elmore, J., Bickel, L., & Price, R. (2015, April). Initial validation of theory of task difficulty and creation of item families. Paper presented at the Annual Meeting of the National Council on Measurement in Education, Chicago, IL.

Simpson, M.A., Kosh, A., Bickel, L., Elmore, J., Sandford-Moore, E., Koons, H., & Enoch-Marx, M. (2015, April). Theory-based item families and the effect of varying grain size on the exchangeability of item isomorphs. Paper presented at the Annual Meeting of the National Council on Measurement in Education, Chicago, IL.

Sinharay, S., & Johnson, M.S. (2008). Use of Item Models in a Large-Scale Admissions Test: A Case Study, International Journal of Testing, 8, 209-236.

Sinharay, S., & Johnson, M.S. (2013). Statistical Modeling of Automatically Generated Items. In M.J. Gierl & T. Haladyna (Eds.) Automatic item generation: Theory and practice. New York: Routledge.

Sinharay, S., Johnson, M.S., & Williamson, D.M. (2003). Calibrating item families and summarizing the results using family expected response functions. Journal of Educational and Behavioral Statistics, 28, 295–313.

Stenner, A.J., Simpson, M.A., Fisher Jr., W.P., & Burdick, D.S. (2015, April). A unified theory of task difficulty in K-12 mathematics. Paper resented at the Annual Meeting of the National Council on Measurement in Education.

Tibaldi, F. S., Verbeke, G., Molengerghs, G., Renard, D., den Noortgate, V. V., & de Boeck, P. (2007). Conditional mixed models with crossed random effects. British Journal of Mathematical and Statistical Psychology, 60, 351–365.

Tuerlinckx, F. & De Boeck, P. (2001a). The effects of ignoring item interactions on the estimated discrimination parameters in item response theory. Psychological Methods, 6, 181-195.

Tuerlinckx, F. & De Boeck, P. (2001b). Non-modeled item interactions lead to distorted discrimination parameters: A case study. Methods of Psychological Research Online, 6(2).

Van den Noorgate, W., De Boeck, P., & Meulders, M. (2003). Cross-Classification multilevel logistic models in psychometrics. Journal of Educational and Behavioral Statistics, 28(4), 369-386.

von Davier, M. (2010). Hierarchical mixtures of diagnostic models. Psychological Test and Assessment Modeling, 52(1), 8-28.

Wang, W. & Wilson, M. (2005) The Rasch testlet model. Applied Psychological Measurement, 29(2), 126-145.

Wang, X., Bradlow, E. T., & Wainer, H. (2002). A general Bayesian model for testlets: theory and applications. Applied Psychological Measurement, 26(1), 109-128.

Weiss, J. (2011, March 31). The Innovation Mismatch: “Smart Capital” and Education Innovation. Harvard Business Review. Retrieved from https://hbr.org/2011/03/the-innovation-mismatch-smart/#.