Automatic Item Generation Methodology in Theory and Practice\textsuperscript{1,2}

Mary J. Pitoniak

University of Massachusetts Amherst

January 24, 2002

\textsuperscript{1}Center for Educational Assessment Research Report No. 444. Amherst, MA: University of Massachusetts, School of Education.

\textsuperscript{2}This study was prepared for the Medical Council of Canada. The author is grateful to Ronald K. Hambleton for his thoughtful comments and suggestions on earlier versions of the paper. The author also wishes to thank Isaac Bejar and his colleagues at Educational Testing Service for the provision of much helpful information on the topic of automatic item generation.
Abstract

Test items are the building blocks of educational and psychological assessments, including achievement and credentialing tests. The first efforts to more systematically generate items and tie them directly to domains of content grew out of the criterion-referenced testing movement in the 1970s, and several types of methods developed out of this focus. However, due in part to the renewed connection between psychometrics and cognitive psychology principles, more theoretically-based approaches to automatic item generation have also evolved. Automatic item generation methodologies can now be classified along a continuum ranging from functional techniques at one end, through model-based methods in the middle, and on to grammatical approaches at the other end. In this paper, automatic item generation approaches representing different points along this continuum are described. In addition, information about current operational implementations of automatic item generation methodologies is presented.
Automatic Item Generation Methodology in Theory and Practice

Mary J. Pitoniak
University of Massachusetts Amherst

The test item is obviously a crucial building block of any assessment. As Wesman (1971) noted, “each item in a test yields a unit of information regarding the person who takes the test. The test as a whole is no better than the sum of its parts; a good test is one composed of well-written items” (p. 81). Over the years, scrutiny of the quality of test items has increased. Today, for example, the search for biased items is as routine as an item analysis. In addition, the need has grown for even greater numbers of test items due to the wide-spread use of computer-based testing, with the provision of flexibility in candidate scheduling of exam times. Security of these exams can only be maintained with large banks of test items. The confluence of these two factors has stimulated even greater interest in automatic item generation procedures, in which the computer is used as an item creation adjunct in order to produce items more efficiently—and eventually, after initial cost and effort outlays, more economically.

Reducing the time and cost needed to produce items is of key importance when one considers the amount of items needed to support a test. For example, estimates have been made that a bank needs to contain 10 to 20 times more items than the number that appear on any test form. Thus, for a 100-item test (perhaps on the short side for many exams in the medical field), a 2000-item bank may be required. If an item costs $1,000 to create (a figure ETS used recently in a lawsuit), this results in the item bank costing $2,000,000 to produce! Although these are only estimates
appear to be on the high side), they serve to illustrate the keen interest out in the field about techniques that will reduce time and expense needed to produce items.

In this paper, a review of literature relating to automatic item generation will be presented. In addition, available information related to its implementation in the field will be summarized. Several types of applications will be addressed, ranging from educational assessments through to aptitude tests. However, the focus will be automatic item generation’s potential for multiple-choice items in credentialing applications, specifically for the Medical Council of Canada (MCC).

In the first section of the paper, early approaches to item generation that grew out of the criterion-referenced testing movement's initial attempts to more clearly define domains of content will be described. In the subsequent section, model-based techniques informed by cognitive psychology principles will be reviewed. In the final section, examples of operational uses of automatic item generation will be described. First, however, a description will be presented of a continuum along which item generation techniques may be classified in order to facilitate the organization and description of subsequent material.

A Continuum of Item Generation Techniques

Bejar (in press) has visualized automatic item generation methodologies as being classifiable along a continuum. Figure 1 illustrates this continuum, as well as those proposed by other authors (Bennett, in press; Dennis, Handley, Bradon, Evans, & Newstead, in press). At Bejar’s functional level of generativity, the focus is more on the domain of content, and not on the psychological construct being measured; thus, a
thorough modeling of examinee responses is not undertaken. As we will see in the next section, methodologies at this level were among the first to be developed, and were for the most part focused on accurately mapping domains of instructional content. An example of a functional approach is to take an existing item and turn it into a shell into which variations on key words can be substituted. Although research about these methods has been limited in recent years, functional approaches continue be very beneficial to testing agencies as an aid to item production, and they may prove useful to the MCC.

<table>
<thead>
<tr>
<th>Dennis et al. (in press)</th>
<th>Approach 1</th>
<th>Approach 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bennett (in press)</td>
<td>Weak</td>
<td>Strong</td>
</tr>
<tr>
<td>Bejar (in press)</td>
<td>Functional</td>
<td>Model-Based</td>
</tr>
</tbody>
</table>

**Figure 1.** A continuum of item generation techniques, as visualized by different authors.

At a higher level of generativity lie approaches that are model-based. Bejar (in press) described these approaches are being guided by models of performance, such as those resulting from a detailed cognitive analysis of the domain that items are developed to measure. Item generation activities at the model-based level are thus informed by principles of cognitive psychology, unlike those at the functional level. Dennis et al. (in press) also go on to distinguish between two types of approaches at the model-based
level: those that start with an established item type and work deductively, using cognitive theory, to clarify the item structure and which features can predict item characteristics (Approach 1), and those that can be seen as “starting from scratch” by using cognitive principles to build an item type from the ground up and more fully use item characteristics to predict functioning (Approach 2). Bennett (in press) also makes a distinction between “weak” and “strong” theories that resembles Dennis et al’s taxonomy. More will be said about the Dennis et al. and Bennett classifications later in the paper.

At the other end of Bejar (in press)’s continuum is the grammatical level of generativity. In these types of approaches item generation activities and psychometric modeling are completely intertwined. As a result, using the appropriate grammar one can “parse” a given item in order to define its psychometric properties. Bejar observed, however, that for the most part implementation of this level of generativity is not yet a realistic goal. We agree, but the work is described here because it represents the next generation of automated item development.

Early Approaches to Systematic Item Writing Methodology

Systematic approaches for writing test items were refined in part as interest in criterion-referenced testing grew during the late 1960s and 1970s. In the 1971 edition of Educational Measurement, Wesman’s chapter on item writing was still basically geared towards norm-referenced testing: “the chief purpose of most tests of educational achievement is to rank examinees as accurately as possible in order of their attainments. Only those items that are answered correctly by the more knowledgeable examinees and
missed by those who are less knowledgeable contribute to the effectiveness of such tests” (p. 87). Although Wesman noted that those statements do not apply to diagnostic or mastery tests, it is clear that those criterion-referenced applications received less attention in his description of item-writing guidelines. Instead, the aim of much item-writing was to develop discriminating items, with less concern given to detailed coverage of a domain of knowledge.

As Hively (1974) observed, “the distinctive thing about Norm-Referenced Tests is that the way the sample problems were generated is not clearly specified. Therefore, it is impossible to call up an indefinite number of tests . . . by systematically sampling from among the different types of problems” (p. 7). In contrast, in what he termed “domain-referenced testing,” the goal is to create a comprehensive pool of items that can be seen to embody the important characteristics of the original universe of knowledge.

As a result of efforts to create items that can be seen to more accurately represent a domain of content, several different systematic item-writing methodologies were developed. Most took advantage of the evolving powers of computing technology. And all sought to transform item writing from an art (e.g., see Wesman, 1971, p. 81) to more of a science. As Roid and Haladyna (1982) observed, these early systematic approaches can be grouped into three main types: (1) linguistic transformations, (2) item forms, and (3) facet design. Each will be outlined in turn; subsections are presented that describe each method, highlight relevant research, review the extent to which the method was computerized, and summarize the current state of the method.

First it should be noted, however, that in this review a number of studies are described that are almost 30 years old. This research is presented for several reasons,
including the painting of an accurate historical pictures of the etiology of item generation.

In addition, however, published research about specific methodologies has waned since that time, and thus some methods must be illustrated with less-than-timely studies. Nonetheless, early efforts to systematize item writing techniques, and at times automate them, are valuable for what they can teach us about what was viable versus what appeared to be a good idea at the time but did not pan out.

**Linguistic Transformations**

**Description of the method.** Bormuth (1970) presented what Cronbach (1970) termed “the first modern proposal for rationalizing item development” (p. 511). Cronbach observed that before Bormuth’s work appeared, texts on item writing tended to summarize the experience of item writers rather than provide scholarly analysis. Specifically, Bormuth presented an outline of item writing technology that illustrated how to construct items from written instruction based upon its syntactical structure. There are two steps to this process. In the first step, a syntactic structure is assigned to the instruction, which involves partitioning the content into discrete chunks and then expressing those chunks in a sentence. After the syntax is assigned, the second component of the process is that “a set of operations [is] performed on that syntax which transforms to relevant segments on the instruction into test items. In other words, the instruction itself is transformed into the questions and the responses through the manipulations referred to by the operations which constitute the definition” (Bormuth, 1970, p. 9) Cronbach (1970) observed that this approach appeared to constrain its application to content that could be put into words, and in fact the book’s main focus in its examples is on assessing written discourse comprehension.
Linguistic transformations may be made at several levels. Bormuth (1970) described two approaches. The first involved transforming a single sentence into a test item, and the second related to anaphora and intersentence syntax. As will be described below, extensions of this work have been undertaken. However, as an introduction to the method, Roid and Haladyna’s (1982) outline of general steps to be taken for the simplest approach—single sentence transformation—will be presented, followed by a description of research in that area.

First, the text should be screened for instructionally relevant key sentences, since much prose contains sentences irrelevant to that purpose that should be discarded. Second, the key sentences should be selected; approaches include having subject-matter experts choose sentences from the passage, having those experts write their own sentences summarizing key points, or doing a keyword or word frequency analysis search. Third, after the important sentences are chosen, they must be transformed. References to other sentences, such as pronouns, should be clarified, and sentences should be simplified so that they do not contain more than one idea. Then, the keyword noun or noun phrase should be removed. For example, in the sentence “Chromium oxide is used as a polishing agent for stainless steel,” (Roid & Haladyna, 1982, p. 103), “chromium oxide” is one of the key noun phrases.

Next the sentence is rewritten in a question format, with the appropriate “wh-word” (i.e., who, what, which, when, where, or why) substituted for the noun. For example, the sentence given above can be transformed into the question “What is used as a polishing agent for stainless steel?” (Roid & Haladyna, p. 104). In the fourth step, distractors are created. Options include using field-test responses, selecting words from
the original list of keywords, or constructing a new list of nouns and categorizing them in ways to facilitate proper selection.

**Research on the method.** Roid and Haladyna (1982) summarized research conducted to evaluate the effectiveness of sentence-based techniques, and the characteristics of the items produced. They and their colleagues (Roid & Finn, 1978; Roid, Haladyna, Shaughnessy, & Finn, 1979, 1980; all as cited in Roid & Haladyna, 1982) investigated how item-generation methodology and the use of different item writers affected item difficulty. They found that if in step 3, the noun removed was a very frequently used one, and was removed from the sentence in which it first appeared, the items were too easy. In contrast, rare singleton nouns, which appeared only once in the passage, were more effective. While one study found that different item writers were able to produce items of similar difficulty using sentence-based techniques, another found that the method of constructing distractors could have a demonstrable effect on the difficulty of the items.

As mentioned above, the linguistic transformation approach can also be extended beyond the two instantiations Bormuth (1970) described relating to single sentence and intersentence syntax. For example, Finn (1975) built upon the work of Bormuth, but instead of using surface structure syntactical analysis, he outlined an approach that used deep structure analysis. He created an algorithm for creating items from text passages, and found that separate item writers were able to create items with a high degree of overlap.

Anderson (1972) argued that comprehension cannot be adequately measured by items that are created using verbatim transformations from text, in methods such as
Bormuth’s (1970) and Finn’s (1975). Instead, he advocated development of items in which instructional text was paraphrased or in which superordinate terms are substituted for terms. He noted that “in order to answer a question based on a paraphrase, a person has to have comprehended the original sentence, since a paraphrase is related to the original sentence with respect to meaning but unrelated with respect to the shape or the sound of the words” (p. 150). In contrast, verbatim transformations may only test rote learning, not comprehension.

Conoley and O’Neil (1979) also advocated extending the Bormuth technique beyond verbatim transformations, and tied usage of different types of items to different levels of Bloom’s (1956) taxonomy. They echoed Anderson’s (1972) argument that comprehension is best assessed through paraphrase questions, while verbatim questions merely test knowledge. In addition, they agreed that the substitution of superordinate terms for specific terms, or vice versa, is useful, and maintained that these types of items can be used to measure the ability of students to engage in application-level activities. Conoley & O’Neil also suggested that through the identification of mediating structure in instruction, questions could also be developed at the analysis level.

Research investigating expansions of Bormuth’s (1970) approach appears to have been rather limited. Roid and Haladyna (1978), in a study comparing a modified-Bormuth item writing approach with a more traditional objectives-based approach, had as one of its aims testing Anderson’s (1972) assertion (later endorsed by Conoley & O’Neil, 1979) that paraphrasing the words taken from the instructional content better assesses comprehension than taking those words verbatim. The linguistic transformation technique used in the study was one modified from Finn’s (1975) 82-step process to be
easier to implement, but it still used rules to generate items by linguistically transforming instructional sentences. Mixed support for their hypotheses was obtained. Contrary to Anderson’s (1972) predictions, verbatim and paraphrased items were not significantly different in difficulty, though a non-statistically-significant difference favoring the paraphrased items was observed. Overall, subjectivity was still present in the item writing process even when rules were employed, as one of the item writers consistently produced easier items than the other. Rule-generated items were slightly less variable in difficulty than objectives-based items, but were not instructionally more sensitive as measured by a pretest/posttest design. The authors concluded that these findings suggest that linguistic transformation-based item creation may not be effectively undertaken in a casual manner, and that perhaps Finn’s (1975) 82-rule approach is necessary.

**Computerization of the method.** Linguistic transformations were not initially computerized, but became so in later years. Finn (1975) used a computer program to compile a list of target words and the sentences in which they appeared, though the actual item writing appeared to have been done by the students who served as subjects in the study. More ambitiously, Wolfe (1976) used a computer-based algorithmic approach to discourse analysis by automating the transformation of text into test questions. As Millman (1984) noted, however, Wolfe’s approach remained at the syntactic, rather than the semantic level. That is, the meaning of the sentence was not considered, only its form.

**Summary of the method.** Linguistic applications have become quite prevalent in information processing approaches such as automated scoring, and work such as Anderson’s (1972) has laid the foundation for advances in the assessment of reading
comprehension (see, e.g., Royer, Carlo, Dufresne, & Mestre, 1996). However, a review of recent literature did not reveal any uses of the Bormuth technique for automatic item generation in research or operationally. Millman’s (1984) observation on Wolfe’s work most likely captures the reasons why the general approach has not had widespread application: “Although the majority of questions made sense, many did not, and Wolfe has abandoned this line of item writing. Also, the kinds of questions generated in this way are quite prosaic and easily formed by unskilled item writers” (p. 181). Despite the apparent fall from favor of Bormuth’s technique, it is clear that development of his theory and its systematic approach to linking items directly to instructional text provided valuable impetus to the development of other principled item-generation methodologies.

Item Forms

**Description of the method.** Osburn (1968) described an item form as follows:

“1) it generates items with a fixed syntactical structure; (2) it contains one or more variable elements; and (3) it defines a class of item sentences by specifying the replacement sets for the variable elements. An item form may be very general and abstract or quite specific and particular” (p. 97). Roid and Haldyna (1982) noted that item forms may be simple in nature or more formalized. A simple item form can be visualized as taking an item and reducing it to its skeleton, leaving blank spaces where words or numbers can be filled in. The following example from Roid and Haladyna illustrates this approach.
**Existing Item**

A random sample of 100 trucks weighed at a highway checkpoint had an average weight of 40,250 lbs. with a standard deviation of 2500 lbs. Find a 95% confidence interval for the true average gross weight of trucks passing this checkpoint accurate to at least one decimal place.

**Sample Item Form**

Given a random sample of \((N)\) (objects) with an average (dimension or feature) of \((M)\), with a standard deviation of \((SD)\). Find a \((95\%, 99\%)\) confidence interval for the true average (dimension or feature) of the (objects) accurate to at least one decimal place. (p. 118)

Specifications for the replacement sets, or characteristics of the permissible numbers, objects, dimensions, and features that may be used in this item form, should also be outlined.

Hively and his colleagues (Hively, 1974; Hively, Patterson, & Page, 1968; Hively, Maxwell, Rabehl, Sension, & Lundin, 1973, as cited in Roid & Haladyna, 1982) made the item form approach more formal in nature. Hively (1974) described item forms as follows: “a list of rules for generating a set of related items is called an ‘item form’ . . . . The ‘shell’ tells the person presenting the item what to say. The blank in the ‘script’ portion of the shell may be filled with any of the replacements shown in the ‘cell matrix’ that follows. . . . The shell and the replacement rules together make up the item form. The item form generates a clearly defined set of items within each of the specified cells” (p. 11). Then, a domain, or clearly delineated group of items, can be generated by
examining what components of the item can be varied in order to generate additional items that measured the same general ability.

**Research on the method.** Hively et al. (1968) outlined use of the item forms approach in the construction of a mathematics achievement test. Each constructed-response item was created through the use of an item form, which are essentially rules for generating test items, and are quite specific as to the features of the item. For example, in an item form from the subtraction universe, even a simple “borrow across 0” item such as “403 – 108” has six rules associated with it (see their Table 2, p. 281). Hively et al. conducted research on tests comprised of these items, and found that the tests in each family (i.e., subtraction, addition, etc.) did satisfy the classical assumptions for parallel tests, such as equality of means, variances, and intercorrelations, as well as independence of true score and error. However, analyses of different components of variance did not support as strongly as they had hoped the utility of item forms to serve as an underpinning for diagnosis and remediation. That is, the item forms did not necessarily represent categories delineating homogenous classes of behavior, as evidenced by the fact that variance within item forms was very similar to variance between item forms.

Other studies investigating the properties of items generated with the item form approach include Macready and Merwin (1973) and Macready (1983). The first study investigated the degree to which items generated by a given form are (1) of equal difficulty and (2) homogenous, meaning that the candidate would answer correctly all items of lesser difficulty than an item he or she was administered and got correct, but would answer incorrectly those items of greater difficulty. Macready and Merwin argued that for tests to be useful diagnostically, these characteristics must be present within an
item form. They reanalyzed data from Hively et al. (1968), and found that “in most cases item forms which generate items of moderate difficulty can be used to obtain relatively homogenous sets of items of equivalent difficulty for a defined population of subjects” (p. 359). However, this did not hold for items with extreme $p$-values. In a subsequent study in which Macready (1983) investigated the characteristics of items generated by a later version of Hively et al.’s (1968) software, an attempt was made to analyze domains whose items were of average difficulty, thus avoiding very easy or difficult items. Homogeneity was then observed for most of the domains.

Popham (1980), in a review of the history of his work with test specifications, described his work with item forms, which he had hoped would work well with efforts to fully explicate domains of content through development of detailed test specifications. Although they were able to compose detailed and constraining item forms, they did not prove workable. Too many item forms resulted, and their features were such that item writers did not tend to use them. Popham and his colleagues concluded that they had ended up creating item forms for “miniscule” classes of behavior, and that “we were overwhelming ourselves with hyperspecificity” (p. 20).

Popham (1980) also found, as had Hively et al. (1968), that item homogeneity did not necessarily result from the use of item forms: “Once upon a time, when I was younger and foolisher, I thought we could create test specifications so constraining that the test items produced as a consequence of their use would be functionally homogenous, that is essentially interchangeable. . . . About the only way we can ever attain functional homogeneity is to keep pruning the nature of the nature of the measured behavior so that
we’re assessing ever more trifling sorts of behavior. That would be inane” (Popham, 1980, p. 26).

Computerization of the method. The item forms approach obviously lends itself to computerization. In their pioneering work with mathematics items in 1968, Hively et al. noted that the items were written by hand, though it would have been possible to program a computer to generate them. Fremer and Anastasio (1969) developed one of the earliest well-publicized application of rule-based item generation techniques via computer. In that study, they developed and programmed error-generation rules in order to create misspelled items for use by writers as they created items (thus, complete items were not generated by the computer program in this exploratory study). The misspelled words generated were deemed by experienced items writers to be an acceptable pool from which to draw material. Another early example of computer-based item generation is word by Vickers (1973) in creating items measuring knowledge of a programming language. The author noted that “even though this system is limited to the study of the Fortran language, it is felt that the same approach should be applicable to many other subjects which obey a set of quantifiable rules” (p. 44).

Millman (1980) and Millman and Outlaw (1978) also developed a computer program to generate items algorithmically. The program that they devised is described by Roid (1979) as follows: “The item program specifies a structure for each question. Some of the wording of the item can be fixed, and elements in the item can be variables that are replaced to form unique questions. Variable elements can be words of other strings of alphabetic characters, random numbers, or quantities computed from mathematical functions” (p. 73). Roid observed that in this approach, actual items need
not be stored on the computer, only the rules required to generate them. In 1989, Millman and Westman advocated the development of an even more ambitious “computer-supplied prototype items approach,” in which the “the author and computer would interact to write the text for the item. Working through sets of hierarchically arranged menus, item writers would choose the mental operation they wish their item to measure, and, in response, the computer would display an item prototype” (pp. 178–179). However, Halaydyna reported in 1999 that, as regarded the particular type of approach endorsed by Millman, “no progress has been reported in advancing this technology and no research has been reported on its use” (p. 126). This is unfortunate, because the approach seems promising, at least in some content domains, and may have some relevance to the MCC and other testing agencies.

**Summary of the method.** Item forms are clearly a step forward from item-writing techniques that paid no heed to systematically addressing domain coverage. The approach is on Bejar’s (in press) functional level, in that it does not call on the principles of cognitive psychology. As Roid (1984) noted, item forms “are an example of a procedural method not requiring a theoretical or research base for the design or selection of items” (p. 52). However, despite the fact that using item forms does not necessarily produce items that were functionally homogeneous (Popham, 1980), the approach does live on in some useful operational applications. Those adaptations will be described later in the paper.

**Facet Design**

**Description of the method.** Guttmann and Schlesinger (1967) described the facet design approach as “a way of devising questions systematically which leaves little room
for arbitrary decision . . . and leads to a construction of items on the basis of a priori definition of test content” (p. 570). Roid and Haladyna (1982) noted that facet design thus requires that the construct under consideration be capable of being described in terms of its components. The structure and boundaries of the domain are defined “by specifying summary statements called mapping sentences that are similar to the item forms” (p. 128).

A mapping sentence is created by a process called “structioning,” in which fixed and variable parts are determined. The fixed part of the sentence is analogous to an item form shell, while the variable parts—termed facets—resemble the replacement set in an item form. Roid and Haladyna (1982) presented the following example of a mapping sentence for measuring a skill:

Given a car that gets 30 mpg and two towns, one of which is a destination, the other a starting point \{Given two towns (Town 1, . . ., Town 10)\}, the student will identify the travel route that is the \{1. shortest, most direct
\begin{itemize}
  \item 2. most scenic
\end{itemize}\} route between the two towns and will identify places to stop for meals and gas and at least one

\begin{itemize}
  \item 1. roadside rest.
  \item 2. park between the two towns.
\end{itemize}

If the distance is great enough to initiate an overnight stay, the student must identify the appropriate town in which to stay (p. 140).

In many ways, the example above appears similar to an item form. However, Roid (1984) observed that “the major difference in the two approaches is that mapping sentence methods were intended to be verified by empirical research (typically using some form of cluster analysis or smallest-space analysis) that documents the relationship
between the theoretical structure of the items and actual data” (p. 55). However, he noted that in practice, this formal confirmation is not usually done.

Research on the method. In a 1984 review of the mapping sentence approach, Roid described some applications in attitude scale construction, but noted that “one must make the intuitive leap from the attitude or personality realm into the achievement domain, in order to see how the mapping sentence method, and facet theory, can be applied to criterion-referenced tests” (p. 56). A review of more recent research did not reveal that any such leaps have been made in practice. Perhaps this is due in part to factors such as those mentioned by Berk (1978): “while the notion of a finite item domain is conceivable, circumscribing of a content domain to the extent that the facets are exhaustive is rarely achieved in practice. The specification of the facet elements, in particular, at a level whereby they supply all essential information for generating an item domain is often impossible or impractical” (pp. 80–71).

Summary of the method. As suggested in the previous subsection, facet theory has not appeared to have borne great fruit in the area of item generation for criterion-referenced testing (that is, achievement testing and credentialing exams). Research is lacking, as is any evidence for its implementation operationally. Shye, Elizur, and Hoffman (1994), in a book devoted to facet theory, noted that it is helpful “for conceptualizing a study, for choosing or creating the study’s variables in accordance with its purposes, and for formulating hypotheses” (p. 3). In addition, they observed that “constructing a mapping sentence is extremely useful for sharpening researchers’ conceptions of the investigated concept universe” (p. 93). They appeared to be
recommending the use of facet theory more for broadly conceptualizing behaviorally-oriented research rather than for writing individual items for large-scale tests.

Summary of Early Approaches

In a 1979 summary of the state of the art of multiple-choice testing, Wood declared “I do not think that the computer has any place in item writing and it is salutary to note that the various attempts in the late 1960’s to program computers to write items appear to have fizzled out” (p. 230). This statement was clearly premature, but perhaps understandable given the difficulties that researchers encountered when trying to use operationally-defined approaches such as those described above, all of which would fall into Bejar’s (in press) functional level. As we will see in a subsequent section, though, procedural or functional techniques can be of great benefit to testing agencies feeling the pressure to create many items quickly. Also, the characteristic of homogeneity which many early researchers strived for with item forms is not an essential characteristic if there is a commitment to field-testing items to determine item statistics. Before turning to the operational uses section, however, model-based methods tied to cognitive principles will be described.

Item Generation Approaches Informed by Cognitive Psychology

Over the past two decades, the potential benefits of combining cognitive psychology principles and psychometrics in test development have been recognized (e.g., see Glaser, 1981). While the earliest psychological tests did reflect ties to cognitive science, this connection became weaker as psychometric developments evolved along less theoretically-based lines (Snow & Lohman, 1989). In recent years, however,
increasing attention has been paid to the ways in which information being gained from research in cognitive psychology can inform measurement in the educational as well as psychological realm.

Embretson (in press) noted that cognitive theory can facilitate the process of item generation in several ways. First, it can guide the determination of which elements in an item can be varied, and how these variations can affect how the item is solved. This will facilitate the development of greater numbers of items, and decrease the degree to which an examinee could “learn” from an item type in which few elements are altered from one instantiation to the next. Second, a cognitive theory that plausibly delineates the problem-solving process will aid in the prediction of difficulty for generated items, since the effects of variation in elements can be more accurately anticipated. Third, and perhaps more important than the capacity to predict item properties, is the capability to explain them. Construct validity is enhanced by the decomposition of item into its cognitive processes. A focus on construct validity at the item level can aid in the assembly of a test that emphasizes specific processing competencies.

The path along which item development must proceed if it is to be guided by cognitive psychology is, as Embretson (in press) observed, not a familiar one to many test developers. She outlined the following steps for ensuring that an integrated and valid explanatory model of task performance is formulated, in what she termed the “cognitive design system approach” (see also Embretson & Gorin, 2001, for another description of this approach). In the first step, the goals of measurement are specified. Embretson acknowledged that this step is also a part of the traditional test development process, but noted that in the cognitive design system approach, two separate types of goals must be
made explicit—those of construct representation and nomothetic span. As described in more detail in Embretson (1983), the former relates more to meaning, and the latter to significance. That is, construct representation is concerned with the types of processes, strategies, and knowledge structures that the examinee uses when answering an item. Nomothetic span refers to the relationship between a score on the test comprised of those items and other measures of that construct. In the cognitive design system approach, both of these types of measurement goals should be stipulated.

The next step in cognitively-informed item development is to identify design features in the task domain. Some components of this step are the same as in traditional item development, such as specifying the mode of administration and format of the item. However, in contrast to the traditional focus of content coverage, the target in the cognitive design system approach is the extent to which item features relate to the cognitive processes, strategies, and knowledge structures that are utilized by the examinee in responding to the item. As Embretson (in press) observed, this step obviously requires familiarity with the principles of cognitive psychology.

The same must be said of the next step in the cognitive design system approach, that of developing the cognitive model. Unlike previous steps, Embretson (in press) argued that this component of item development has no analogue in traditional item development. The cognitive model is informed by research into the cognitive processes, strategies, and knowledge structures theorized in the previous step as being relevant to the item type. Both theory and empirical results must be integrated as a part of this review process. Next, these findings are used to inform the operationalization of these processes in the stimulus features of the item. The empirical studies revealed during the
literature search will most likely describe relevant item features, which must be then quantified on the items under development in a manner that yields item characteristics that are not only manipulable but scorable.

Another key component of the development of the cognitive model is the impact of these item features on the psychometric characteristics of the items. Research should be conducted using existing items in order to determine the relative impact of given features on item difficulty and discrimination; such studies will assist in the evaluation of the potential of different cognitive models. As Irvine (in press) noted, item characteristics can be viewed as either radicals, which are structural elements tied to the construct under consideration and will affect item difficulty, or incidentals, which are construct-irrelevant surface characteristics that will not impact item functioning.

In the next step, items are generated in accordance with the cognitive model. Information gleaned from previous development activities will inform the specification of item structures and substitution rules for stimulus features. Then, the success of the cognitive model is evaluated through empirical tryout of the generated items. Embretson (in press) noted that in contrast to traditional test development, in the cognitive design system approach the ability of the model to predict item difficulty is appraised. This is done by looking at the relationship between item structure and item stimulus features, which can be viewed as the independent variables, and item difficulty and response times, representing the dependent variables. In addition, the psychometric model postulated to describe the construct must be evaluated; for example, the dimensionality of scores should be assessed to see if the assumption of unidimensionality is warranted in a given case.
If the results of model evaluation are positive, and items can be generated with characteristics predicted by the model, those items can then be banked. In addition to traditional item characteristics such as difficulty and discrimination, item features relating to their cognitive complexity are also designated. The preceding steps have laid the foundation for the establishment of construct validity. Nomothetic span should also be investigated by comparing test scores to external measures of the same construct. Embretson (in press) observed that these studies, often also part of traditional test development, are better informed in the cognitive design system approach by specific predictions made as a result of the deeper knowledge of how item features relate to the construct.

Embretson (in press) goes on to illustrate the implementation of these steps in the cognitive design system approach in the development of abstract reasoning items. Following consideration of the measurement goals and task design features, four studies investigated (1) the cognitive model in relation to existing items; (2) the generation of items using this model; (3) the evaluation of psychometric properties in terms of the model; and (4) the nomothetic span of the scores via comparison to another test. The construct of abstract reasoning, or the ability to make inferences independently of specific knowledge, was operationalized through the use of matrix completion items. Embretson described in detail the cognitive model that was built up from an existing theory (Carpenter, Just, & Shell, as cited in Embretson, in press), augmented by mathematical models developed in the study.

In the first study described in Embretson (in press), existing items were characterized by features specified by two cognitive models, and these models’ ability to
explain item difficulty was investigated. Both models were seen to predict difficulty equally well, and thus structurally equivalent items could be developed from either theory. In the second study, new matrix completion items were generated using the rules refined during the first study. The types and combinations of rules were determined, and a formal structural notation was developed to describe the objects and attributes relevant to the items. Then, items were generated, it appears by computer (though this is not made explicit).

In the third study reported by Embretson (in press), an empirical tryout was conducted to evaluate whether items were of acceptable psychometric quality, whether structurally similar items were also similar in difficulty, and whether construct representation, exemplified by the effectiveness of mathematical modeling of item difficulty and response time, appeared adequate. The study provided support for all three criteria, and again, both models investigated in the second study also appeared plausible when evaluated in view of the results of the third. In the fourth and final study reported, nomothetic span of the matrix completion items was explored by analyzing the factor structures of scores and comparing them to those of another measure, the Raven’s APM (Raven, Court, & Raven, 1992, as cited in Embretson, in press). Scores from tests comprised of the generated items correlated highly with those from the other measure, and were also seen to have greater internal consistency.

In summarizing the results of these four studies, Embretson (in press) observed that the results are promising in that they demonstrate the feasibility of automatically generating items for high-level abilities, in contrast to the low-level or specific abilities often seen as the most likely candidates for item generation. Also of note is the varied
appearance of the items that were produced by the generating system, despite their structural similarity. Similarly encouraging is the extent to which the item structures and the cognitive model were able to predict item difficulties, which enhances construct validity. However, Embretson noted that these successes would not have been achieved without the development of a plausible cognitive model, which may be difficult in some test development contexts.

The cognitive design system approach proposed by Embretson (in press) and exemplified by the four studies conducted with matrix completion items is a comprehensive method for utilizing cognitive psychology principles and a deeper understanding of the construct to aid item generation. However, work described Dennis et al. (in press) illustrates that there can be a range of approaches informed by cognitive principles. They described two sets of studies, one of which represented what they termed Approach 1; the other, Approach 2.

In Approach 1, items were generated for existing tests. Dennis et al. (in press) observed that the potential for generativity existed for the tests described because (1) the capability existed to specify an algorithm with which items may be created from a universe of possible items, (2) item features could be identified that may be manipulated to affect item difficulty, and (3) features could also be targeted that do not affect item difficulty and can be altered to change the appearance of the item without changing its functioning. After item features were specified, a computer algorithm was designed and implemented to generate items that vary along these features. However, it should be noted that distractors were not generated according to features or rules. Because of what they described as the potential for a given distractor to interact with other distractors or
features of the stem, they did not use distractors to manipulate item difficulty, and instead fixed them for all items of a given type. Dennis et al. described two studies in which the ability to predict item difficulties from item features generated using Approach 1 was investigated. The first study involved a multiple-choice spatial test for officer selection, and the second, literacy and numeracy tests for candidates for naval enlistment. In both cases, difficulty was able to be predicted with sufficient accuracy for the purposes specified.

Approach 2, as described by Dennis et al. (in press), can be contrasted with Approach 1 as being more closely tied to the cognitive principles underlying the construct and the items generated. In addition, in Approach 1, the item universe is specified by combining certain predefined item features, while in Approach 2 an algorithm is designed, and the number and types of items generated is not known a priori. In Approach 2 the models are more complex, and take into account more factors; also, distractors may also be a function of the rules incorporated in the generative algorithm.

Newstead, Bradon, Handley, Evans, and Dennis (in press) described the work conducted to develop the cognitive model underpinning the analytical reasoning items created using Approach 2, while Dennis et al. (in press) summarized the generation and empirical evaluation of the items. The items generated were of the analytical reasoning type found on the Graduate Record Exam (Educational Testing Service, 1996, as cited in Newstead et al.). Steps involved in the formulation of the cognitive model in Approach 2 echo those described by Embretson (in press).

In Phase 1, Newstead et al. (in press) worked with content experts and conducted think-aloud protocols in order to develop a conceptual framework for the problems, and
then created a computer program in which the structure of the items could be encoded. In Phase 2, available sources for information were studied that could be used to formulate hypotheses about factors underlying difficulty; these resources included both experimental and psychometric data already existent in the literature and compiled by ETS through previous work on the GRE. Next, in Phase 3, they conducted experiments to investigate factors that impacted the difficulty of different tasks, moving from basic rules on to problems that resembled actual analytical reasoning items. Finally, in Phase 4, a performance model and underlying theory were designed and refined. Dennis et al. then described how the performance model was evaluated through the generation of items and an examination of their performance. They found that they were able to predict item difficulty at an acceptable level from the structure and features of the items. It should be noted that distractors were varied in this Approach 2, as compared to in Approach 1.

Goeters and Lorenz (in press) reviewed research undertaken to develop tasks for aptitude tests given to candidates for aviation positions such as pilot, air traffic controller, and astronaut. The authors implemented what may be termed by Dennis et al. (in press) as an Approach 2 study. As Goeters and Lorenz themselves noted, in this case items were created and tests were composed under the guidance of a validated information processing manual, in contrast to other studies where what they termed “experimental prototyping” is employed (a la Approach 1). Tests previously administered via paper-and-pencil were adapted for a computerized adaptive testing platform, and generation techniques were used to produce items. Theory and past research undergirded their attempt to determine which item properties would affect difficulty, and which effects of item variables were moderated by individual differences. Mixed results were obtained in
relation to various hypotheses too extensive to describe in this paper; the reader is referred to the study for an example of a thorough and detailed explication of construct delineation and investigation.

Hornke (in press) described a series of studies conducted over the past decade to investigate the degree to which item difficulty could be predicted by item features. The types of items developed included: mental rotation, pattern matrices, verbal analogies, number problems, visual analysis, visual memory, serial learning, and management/work behavior and attitude scales. The bulk of what Hornke described were the results of empirical investigations, and thus the nature of the process undertaken to specify item design rules and features was not given much emphasis. He stated only that for most of the item types, a review of the literature informed the design of the items, except for the last type, in which practical considerations were heeded (however, the reader is referred to Hornke & Habon, 1986, for a more detailed description of an implementation of item generation with matrix-type items). Of note in Horne (in press) was the fact that the items were created by hand, not by computer. For the most part, item design rules were able to predict item difficulty (using the one-parameter logistic model). However, there was some variation in the level of success achieved. Analogies were predicted least well, and visual memory, the best. Hornke observed, as have others, that as item content is more restricted, item design rules are able to predict item difficulty more effectively.

Space does not permit the summary of all research in item generation for aptitude tests. Irvine (in press), however, provided a helpful overview of item generation research in three areas: (1) R-models, in which responses are scored right or wrong; (2) L-models, which measure response time; and (3) D-models, which involve repeated measurements.
of individuals. The ability of researchers to predict $p$-values in each of these item types is documented at length; the reader is referred to Irvine’s chapter for this informative review. Bejar (1993) also provided a useful summary of aptitude-related item generation studies done through that time. The reader is also referred to specific studies by Bartram (in press), Bejar and Yocom (1991), Embretson (1998, 1999), and Irvine, Dann, and Anderson (1990).

Model-based research on item generation has also been done at Educational Testing Service. Because much of it is tied to their proprietary software, a review of this work will be presented in the next part of the paper.

**Operational Use of Item Generation Techniques**

Within this section, operational uses of item generation techniques used by large-scale testing companies in the United States will be described. Some of these methodologies are less “automatic” (i.e., computerized) than others, but due to the limited availability of information in this area, a variety of applications along the automaticity continuum will be described. Note will also be made of the placement of each approach on the functional/model-based/grammatical continuum (Bejar, in press) described earlier.

First, two similar sets of systematic, though not technically automated, item generation techniques used for medical licensing tests will be described that will be of interest to the MCC. Next, a study done at American College Testing Program with their proprietary software will also be reported. Then, work done at Educational Testing Service, operationalized in their proprietary item generation software, will be described. Research related to work presented in this section will be provided as available.
Item Shells

Haladyna and Shindoll (1989) outlined an item writing technique that echoes item forms, though that connection is not made explicit. In their approach, an “item shell” is used, which is a “‘hollow’ item that contains the syntactic structure and context of an item without specific content” (p. 99). Item shells are developed from existing items; they may be rigidly structured, or may be “open shells” that include prompts for information. Haladyna (1999) summarized the steps in creating an item shell as follows: (1) identify the stem of a successfully performing item; (2) underline key words of phrases representing the content of the item; (3) identify variations for each key word or phrase; (4) write the correct answer; and (5) write the required number of distractors (pp. 129–130).

As an example of an item based on the general premise of a patient coming to the hospital in need of attention, Haladyna and Shindoll (1989) presented the following original stem and an open shell derived from it:

ORIGINAL STEM

A 6-year old child is brought to the hospital with contusions over the abdomen and chest as a result of an automobile accident. Initial treatment should consist of:

DERIVED OPEN ITEM SHELL

A (age, preexisting condition, gender of patient) is brought to the hospital (complaining of, with injuries showing symptoms of . . .) (as a result of). Initial (treatment, intervention, diagnosis, laboratory studies, antibiotic therapy) (should consist of, is, are, include): (p. 101)
Haladyna and Shindoll (1989) described the results of a tryout of the item shell with health care professionals, implemented in order to evaluate whether the technique improved the quality of the items. They noted that these item writers often feel ill-equipped for their tasks, and posited that item shells would make their job easier, while not being too constraining: “the item shell complements the more traditional method of item writing without restricting item writers to any single form for an item. To briefly reiterate, item shells provide syntactic or contextual alternatives that traditional item-writing methods do not. The task for the item writer, then, shifts from primary construction to specification of content for certain items or ‘fleshing out’ more complex items. Because the item writer, ideally, toils less with expression, item writing may be a faster, more efficient process and more likely to be regarded with enthusiasm” (p. 104).

The study provided limited, yet encouraging, support for this hypothesis. Of the 50 items that were prepared in content areas in which item shells had been provided to the item writers, only 20 (40%) were constructed using the item shells. The majority of those items were, Haladyna & Shindoll (1989) noted, editorially sound, with fewer grammatical and punctuation errors than items prepared by the same item writer without using the item shells. However, they concluded that overall performance, presumably reflected in all items created by the writers, were not improved by merely having the item shells as a guide. In addition, no item writers used the open shells (as per the example given above), thus limiting their use to lower taxonomic levels where more rigid shells were employed. It is important to note that this technique did not appear to be computer-assisted in any manner, even as an aid to composition. However, it certainly appears
feasible to store the item shell and replacement sets electronically, and to design a relatively simple program to generate items for review.

The item shell approach can also be used for item writing in other fields. For example, R. Hambleton (personal communication, January, 2002) has worked with item writers in the security industry to take items of good quality and use them to generate additional items. He estimated that, for example, a single item may easily be used to produce 10 to 20 additional items. An example of an item stem from which key variables have been removed, and for which replacement sets have been created, is presented in Figure 2. For this sample item, approximately 120,000 different item stems could be yielded from different combinations of values from the replacement sets! This is clearly an upper limit and not reflective of the way that item forms would most likely be used in practice, but the example clearly illustrates the power of this technique.

The item forms approach is clearly functional (Bejar, in press) in nature, as it is not based on cognitive principles. As such, the potential for knowing a priori the characteristics of the items so generated is small. As Bejar (1985) observed, “in practice, items differ with respect to a number of characteristics, and a useful generation scheme must have control over those characteristics. For example, a useful generation scheme should be able to generate easy items or hard items at will. I suspect that to build such systems it is first necessary to have an idea of what makes an item easy or hard” (p. 284). However, the inability of the item form approach to generate items with known characteristics does not render it as lacking in usefulness. More will be said about this point in the Discussion section of this paper.
**Shell for Stem**

XYZ Dealer purchases [__1__] bond through [__2__] sale. XYZ Dealer serves as a [__3__], and is [__4__] on this issue. By [__5__] after the [__6__], the [__7__] must deliver a copy of the [__8__] to which of the following entities?

<table>
<thead>
<tr>
<th>Replacement Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Advance Refunding</td>
</tr>
<tr>
<td>Taxable</td>
</tr>
<tr>
<td>Tax Exempt</td>
</tr>
<tr>
<td>GO</td>
</tr>
<tr>
<td>Revenue</td>
</tr>
<tr>
<td>2. Negotiated</td>
</tr>
<tr>
<td>Competitive</td>
</tr>
<tr>
<td>3. FA</td>
</tr>
<tr>
<td>Senior Manager</td>
</tr>
<tr>
<td>Co-Manager</td>
</tr>
<tr>
<td>Syndicate Member</td>
</tr>
<tr>
<td>Selling Group Member</td>
</tr>
<tr>
<td>Placement Agent</td>
</tr>
<tr>
<td>4. Working with consultant</td>
</tr>
<tr>
<td>Serves as FA</td>
</tr>
<tr>
<td>Made political contributions</td>
</tr>
<tr>
<td>Has a control relationship with issuer</td>
</tr>
<tr>
<td>5. 1 business day</td>
</tr>
<tr>
<td>2 business days</td>
</tr>
<tr>
<td>7 business days</td>
</tr>
<tr>
<td>10 business days</td>
</tr>
<tr>
<td>6. Hiring date</td>
</tr>
<tr>
<td>Sale date</td>
</tr>
<tr>
<td>Bond settlement</td>
</tr>
<tr>
<td>Award date</td>
</tr>
<tr>
<td>Account settlement date</td>
</tr>
<tr>
<td>7. Issuer</td>
</tr>
<tr>
<td>FA</td>
</tr>
<tr>
<td>Senior manager</td>
</tr>
<tr>
<td>Co-manager</td>
</tr>
<tr>
<td>Syndicate manager</td>
</tr>
<tr>
<td>8. POS</td>
</tr>
<tr>
<td>OS</td>
</tr>
<tr>
<td>ADRD</td>
</tr>
<tr>
<td>FA Disclosure Letter</td>
</tr>
<tr>
<td>Consultant Notification</td>
</tr>
</tbody>
</table>

*Figure 2.* Sample item shell and replacement sets for the stem of an item from an examination in the securities industry.
Item Modeling

LaDuca, Staples, Templeton, and Holzman (1986) described an approach they termed “item modeling,” which they proposed as a systematic method that would assist in the preparation of functionally equivalent items. In this approach, a source item is identified, from which specifications are delineated that describe the content of the stem and options for responses. For example, the stem content in the sample source item presented in their article has four types of information: (1) a description of the patient and his or her medical history, (2) the patient’s symptoms, and their duration, (3) findings from a physical examination, and (4) outcomes of diagnostic tests. Response options include the correct response and distractors.

In a study conducted by LaDuca et al. (1986), the participants—all doctors—listed alternatives for each of the predetermined stem content categories, and generated answer options, both correct and incorrect, for items built from those categories: “using the source item as a model, doctor experts identify significant alternatives under each category of stem content, and stipulate the nature of the incorrect options (distractors). Completing the process leads to formulation of a set of item specifications for a family of test item [sic] addressing the same the same or closely related evaluative objective, and, in the extreme, exhaustive of the source item’s major variants” (p. 53).

A four-member task force was then formed, and its members were able to produce specifications for more than 100 items when starting with ten source items. A non-doctor item writer then wrote items based on those specifications, and the items were deemed acceptable by the task force. Writing the specifications may be as time-consuming as writing items, but the fact that actual items aren’t written until committees review the
specifications was seen as beneficial. Committee reviews of specifications may replace reviews of items themselves. Although items were deemed by expert judgment to be equivalent construct-wise, empirical data was still needed; “further research on item modelling will incorporate systematic data-gathering and analysis to assess functional equivalence of items prepared using this method” (LaDuca et al., 1986, p. 56).

In Shea, Poniatowski, Day, Langdon, LaDuca, & Norcini (1992), an extension of the approach outlined in LaDuca et al. (1986) was described. As before, “an existing item stem is divided into information sets that might be changed to create additional items in the same domain” (p. 20). In this study, the 1986 work was extended in two ways: first, the modelers were instructed to modify the stem in such a way that one of the distractors became the correct response. Second, the task included in the stem was systematically varied. Data suggested that item modeling was four times more efficient than the existing process in terms of meeting time alone. In this study the items were actually written by the modelers (as opposed to previous study, in which only specifications were written by them).

Haladyna (1999) observed that although item forms and item modeling appear similar, the latter is more closely tied to a fuller exploration of the domain under consideration. As such, he noted that while item forms are operationally defined, item modeling is construct defined. In fact, he outlined an approach to item modeling that divided parts of the items into what he termed facets (though without reference to facet theory, a la Guttman & Schesinger, 1967); facet one being the setting, facet two, physician tasks; and facet three, case cluster. In the latest description of the method by the earlier researchers, however, LaDuca, Downing, and Henzel (1995) presented steps
for preparing modeling specifications that do not include that terminology. While item modeling may indeed be more closely tied to the construct, the technique as described above would seem only to be at the weak end of Bennett’s (in press) continuum in that it does not allow for cognitive modeling of examinee responses. (A more cognitively-based extension of item modeling is presented later in the paper, in the context of work done at ETS by Bejar et al., 2001.)

Item modeling as described within this section has been used intermittently at the National Board of Medical Examiners (NBME, T. LaDuca, personal communication, December, 2001) and at the American Board of Internal Medicine (J. Norcini, personal communication, January, 2002). These approaches have not, however, been automated via computer. They appear to serve more as facilitative resources for human item writers. For example, the item writing guide published by the NBME (Case & Swanson, 2001) describes “item templates,” notes that many items can be placed within the context of a patient vignette, and provides numerous examples in which the different components of the vignettes are varied. This relatively unstructured form of item modeling would most certainly not produce items with identical psychometric characteristics (or “isomorphs,” Bejar, in press), and no research has been done at NBME to investigate the functioning of items produced when more structured item modeling has taken place. Nonetheless, the use of item models is of value in facilitating the work of human item writers.

American College Testing Program’s Math Item Creator

Meisner, Luecht, and Reckase (1993) described results of research done on the performance of mathematics items generated algorithmically for the ACT Assessment Program Mathematics Test. Items were created using proprietary software entitled Math
Item Creator, using algorithms covering content ranging from pre-algebra to intermediate algebra, and cognitive skills from basic skills on up through higher-order analysis.

Examples of the low and high ends of the continuum are, respectively, algorithm 1, “solving a quadratic equation by factoring,” and algorithm 16, “calculating a coefficient that would give a system of linear equations in 2 variables an infinite number of solutions.” They provided the following example as an illustration of how their algorithm functions:

An algorithm like the following could be used for generating stems:

For all $x$, $(ax + b)(c) = ?$

where $a$ was a randomly generated integer chosen from the set \{2,\ldots,9\} and $b$ and $c$ are randomly generated integers chosen from the set \{2,\ldots,9\}.

The key and four alternatives reflecting plausible solution errors could then be derived from the values of $a$, $b$, and $c$ in the following way:

A. $acx + b$
B. $ax + b + c$
C. $ax + bc$
D. $acx + bc$
E. $abcx$ (p. 8)

Meisner et al. (1993) reviewed the statistical characteristics of the items, which were pretested on operational forms, and concluded that most of the algorithms produced items that were homogenous. For those algorithms that did not, inspection of differences among the items sometimes yielded information that might prove helpful in subsequent work. A replication study was mentioned, but a review of research reports from ACT did not reveal same (perhaps due in part to the fact that the second and third authors left ACT
during that general time period, and that research may not have been pursued or
summarized as the first study had been). In addition, several efforts to obtain information
from test development staff at the company about continued use of the item generation
software were not successful.

**Educational Testing Service—Test Creation Assistant**

Perhaps the most sophisticated automatic item generation system that has been
used operationally and about which information is available is the Test Creation Assistant
(TCA) developed at Educational Testing Service. As Singley & Bennett (in press)
described, the development of the TCA involved the application of schema theory to
automatic item generation. They noted that although problems may appear on the surface
to be quite different, they may actually be instances of the same schema, or underlying
problem structure. However, for any given problem based on that schema, not all of the
underlying equations are necessarily required to solve it. In this sense, schemas have
psychological content; as such, they can be differentiated from problem descriptions that
are merely at the structural level.

Schema theory is instantiated in the Mathematics version of the Test Creation
Assistant (1998) in relation to the systems of equations that underlie word problems and
that tie the entities of the problem together. Singley & Bennett (in press) presented an
example of the round-trip schema, in which a system of equations can be seen to underlie
the problem. Taken together, the variables and equations define an entire class of
problems. An illustration of this type of problem will be presented below, along with a
sample screen from the Mathematics TCA.
However, it should be noted that although the original implementation of the Mathematics TCA was from a strong, model-based theory (Bejar et al., 2001), subsequent research demonstrated that fully developing schemas for each problem class within the domain of quantitative reason would be extremely time-consuming. As a result, the software as presently used by item developers is used more at the functional level. However, efforts are being made to bring cognitive schema back into the picture (I. Bejar and R. Bennett, personal communications, January, 2002).

Descriptions of the software are presented in Bennett (1999) and Singley and Bennett (in press), and the reader is referred to those sources for a more detailed review of its features. However, the following is a brief description of the version of the software used for quantitative items—the Mathematics TCA.

A sample screen from the Mathematics TCA is presented in Figure 3. As described by Singley and Bennett (in press, in which this figure also appears), this screen depicts the “Model Workshop,” in which the creation and editing of the item model takes place (as can be seen in the top right section of the screen, the other two tabs for which screens are available are “Family Overview” and “Generate Variants”). The multiple-choice item type shown here is one of three templates, the other two being quantitative comparison and data sufficiency (however, the authors noted that the multiple-choice model may also be used to create constructed-response items merely by leaving the distractor fields blank).

On the left side of the screen in Figure 3, the item stem is presented. The variable names, shown in bold, take the place of literal strings of text and/or numbers in the original item. All non-bolded elements remain the same in each instantiation of the item
Figure 3. Sample screen from Educational Testing Services’ Mathematics Test Creation Assistant computer software program.

(termed a variant, Singley & Bennett, in press). Variables that contain numeric extensions, such as \textbf{SVar.1}, make up an \textit{ntuple}; that is, each variable’s value changes in synchrony with the other variables. This allows for the situational context to vary from one variant to the next, with the assurance that the values taken by the variables will also change in a coordinated manner. In the top-most window of the right-hand pane, labeled “Variables,” variables are defined, their type (string, real, integer, untyped) specified, and acceptable values enumerated. Values can either be specified as a range, as shown for the variable “Rd” (“1 to 5 by 1”), or as an equation in which the value of the current variable is derived from the values of other variables.

In the next window, “Variation Constraints,” equations that specify the relationships between the variables in the schema are given. In the “Distractor Constraints” window, equations defining the values that distractors will take on are presented; in this case, each distractor is one variable (Du, the length of the first leg of the trip), plus or minus a constant. Shown at the bottom of the screen is general information about the model. “GRE” indicates that the program with which the model is associated; “Family” indicates that the model comes from the DRT, distance-round-trip family; in “Attributes,” SMC stands for standard multiple choice; “nongeneric” is a memorability classification; and “Near” indicates that the mode will produce near variants that are different only in their surface features.

Though the example shown here creates a “near variant,” or an item whose surface features are different, the model itself may be modified to produce variants that are less similar in appearance to the parent item. As Singley and Bennett (in press) noted, in the round trip example one could define a new model that gives different information
in the stem and thus requests different information in the answer. Thus, “givens” and “goals” can be modified to create problems that are built within the same general problem family but look substantially different.

Once the item developer has finished constructing the model, he or she can press the “Test All” button along the right side of the screen, which allows the system to test whether the model is sufficiently constrained. If it is, the item developer can proceed to create variants by clicking on the “Generate Variants” tab at the top right-hand corner of the screen. An example of a variant created from the model shown in Figure 3 can be found in Bennett and Singley’s (in press) Figure 14.2, not reproduced here. In addition, as shown in their Figure 14.3 (also not presented here), attributes affecting item difficulty are presented, and a predicted item difficulty is calculated (though this function is not currently used to any great extent; M. Morley, personal communication, July, 2001).

The Mathematics TCA is used to produce items that are then entered into the ETS Test Creation System. The items are banked, reviewed, pretested, and used operationally along with other items drafted by hand. Tests on which these items have been used include the SAT I, GRE General Test, and the Graduate Management Admission Test (Singley and Bennett, in press).

A version of the TCA is also under development for analytical reasoning items. It is still in the research stage, however, and no items have been produced for operational use. The system is now only capable of producing a “raw form” of the item, one without “clothing,” or the English language to wrap around the values of the variables; research is being done, however, on how to automatically generate text for these items (Fairon, 2001). It is also important to note that current plans are for the analytical reasoning
version of the software to be at the grammatical level, making it possible to not just
create, but also analyze, new items (I. Bejar, personal communication, January, 2002). A
third version of the TCA, for verbal reasoning, is also just beginning to be developed
(P. Brittingham, personal communication, July, 2001).

The TCA can also be customized for other content areas. For example, the
Chauncey Group, an ETS subsidiary, has worked with the American Institute for
Certified Public Accountants (AICPA) to create item variants using a modified version of
the TCA (Greenberg, 2001). As in the mathematics example described above, an existing
item—in this case for the CPA Examination—can be used to create an item model.
Variables are defined, and constraints are imposed to clarify the values that variables can
take on and how the variables can relate to each other. As with the Mathematics TCA,
the model can be changed in order to produce items that are less or more similar in
surface features. And as with that version, the TCA used for the Uniform CPA
Examination also tracks generation activities in order to prevent the creation by the
system of identical items. Greenberg reported that the TCA has been used to produce
approximately 1,800 items from about 60 item models. Items created using the TCA
have been pretested and used operationally; however, no formal analyses have yet been
done to establish whether items perform similarly to their “parent” items (A. Goldman,
personal communication, January, 2002). These items have been in a financial area, in
which numerical constraints can be built in, as in the Mathematics version of the TCA
(R. DeVore, personal communication, January, 2002).

Research has been done at ETS into several aspects of automatic item generation,
though not necessarily directly linked to the TCA. For example, Enright, Morley, and
Sheehan (1999) conducted a study that might best be classified as falling into Dennis et al.’s (in press) Approach 1, or Bennett’s (in press) weak model. Enright et al. investigated the statistical characteristics of systematically designed mathematical word problems—items developed with a knowledge of the cognitive attributes required to solve them (however, the items were written by hand, and were not created with the TCA described above; M. Enright, personal communication, January, 2002).

Enright et al. (1999) noted that systematic creation of items is not a new idea, and has been long been done informally by item writers. However, that practice brings with it the risks of inadvertent overlap among items or narrowing of the construct. A more structured approach would allow the creation of item variants through the use of guiding principles based on a thorough understanding of the construct. Such an understanding would be reflected in the systematic manipulation of construct-relevant item features to create items of varying difficulty, and construct-irrelevant, incidental features to create items of similar difficulty. However, Enright et al. (1999) observed that “the constructs tapped by most existing tests are not articulated in enough detail to allow the development of construct-driven item design frameworks” (p. 1).

An alternative approach—the one they utilized—was to work from observed correlations between item characteristics and those items’ statistical functioning to determine features that could be manipulated. Hypotheses were thus generated that could be tested through analysis of those items’ subsequent functioning. Enright et al.’s (1999) work thus serves as an example of construct-driven item generation, which “requires a description of items that can be related to the processes, skills, and strategies used in item solving. The benefits of such an approach are that, if item variants can be created
systematically through an understanding of critical item features, tests can be designed to cover important aspects of a domain, overlap can be controlled, and pretesting requirements can be reduced” (p. 29).

Results from the Enright et al. (1999) study were promising in that manipulated features accounted for much of the variance. Two types of items were investigated: rate and probability. Rate items spanned a wide difficulty range, and manipulated features accounted for 90% of the variance in difficulty. Item discrimination and guessing were also similarly associated with item features. Probability items were more difficult, and their difficulty range more limited. Compared to rate items, manipulated item features for probability items accounted for less of the difficulty variance, and little of the discrimination or guessing variance. The authors discussed how collateral knowledge of how item features affect their functioning could be used with statistical procedures designed to reduce pretest sample sizes (see, e.g., Mislevy, Sheehan, & Wingersky, 1993). Extensions of this work, including refinements of the item classification model, are presented in Enright and Sheehan (in press).

Two recent studies at ETS focused on how to deal with the fact that items that are created to be isomorphs—identical in psychometric characteristics (Bejar, in press)—often do not perform identically. Dresher and Hombo (2001) and Hombo and Dresher (2001) cited several studies previously done at ETS that demonstrated such findings (Lewis, 1985, 2001; Mislevy, Wingersky, & Sheehan, 1994), and sought to establish to what degree slight variation in item parameters resulting from lack of isomorphism would affect estimation of examinee ability. In both studies, Dresher and Hombo found that ability estimation was robust, in that using parameters with a small degree of error in
them did not significantly affect the estimation of examinee ability. They concluded that automatic item generation “appears to be a feasible and importance alternative to individual item construction” (p. 17), while cautioning that further research is necessary.

At the cutting edge of work in automatic item generation is a study by Bejar et al. (2001), in which the testing of a pilot version of an on-the-fly adaptive testing system was described. This work was part of a coordinated ETS research effort that also includes studies described above. As noted in the description of the Mathematics TCA, the item model is a key component of the generative process. However, Bejar et al. noted that their use of the term item model is clearly an extension of the manner in which LaDuca et al. (1986) used it. It is useful to present the following description of the item model to illustrate its theoretical foundation in the applications described in the 2001 study:

An item model (Bejar, 1996) can be thought of as a procedure for instantiating isomorphic items—items that are exchangeable psychometrically and content-wise. We view item modeling as construct-driven because it entails an understanding of the goals of the assessment and the application of the pertinent psychological research. That is, item models set the expectation for the behavior of the instances produced by a given model and those expectations can be verified upon administration of the isomorphs, thus providing an opportunity to refine our understanding of the construct and the supporting psychological principles (Bejar et al., 2001, p. 1).

As noted earlier, the Mathematics TCA was first designed to operate from strong theory, which would allow item models to produce isomorphs. However, at this time the software is less schema-based, and thus many items created by what is still termed an
item model are viewed instead as variants, or instances of an item model that are not necessarily identical in a psychometric sense (Bejar, in press). This point is emphasized only to prevent confusion due to the somewhat different meanings that the term item model has been given in different research efforts described within this paper.

In the Bejar et al. (2001) study, existing items from a GRE CAT pool were converted into item models. First, a simulation study was conducted to assess score precision and bias when item models were used. Results indicated that while bias was minimal, score precision was lower, particularly in the middle range of the true score scale. The authors observed that this drop in measurement precision could be offset operationally by slightly lengthening the test form.

The second component of Bejar et al. (2001) was a field study in which an experimental CAT composed of items generated on-the-fly was administered to examinees, in addition to a linear test form. Item parameters used in the selection algorithm were those from the original items, attenuated to conservatively approximate a lack of isomorphicity. Previous operational GRE scores for the candidates were also obtained, and these were shown to correlate .87 with the scores obtained with the experimental CAT (though mean performance on the CAT was lower, which could be due both to regression toward the mean due to a restricted sample and possible lower student motivation). The authors noted that the correlation of .87 is on a par with the test-retest correlation that characterizes operational conditions. The functioning of items on the linear form was also examined to assess isomorphicity across items within a given item model, and the results were very positive. The conclusion was drawn that “the results of this study provide initial evidence that an item model-based approach to
adaptive measurement of quantitative reasoning is feasible and could provide a more efficient and economical approach to adaptive testing” (p. 42).

Other Information on Operational Implementations

The title of this subsection is misleading, in that despite numerous attempts to do so, no further information could be gleaned about operational uses of automatic item generation techniques. Neither a posting to the American Educational Research Association Division D forum nor inquiries to individual companies proved fruitful. This is most likely due to two factors: (1) those companies that are using proprietary item generation software are not interested in releasing information publicly, and (2) many large-scale testing companies are not in fact using these methods.

Support for the second hypotheses, that in some cases totally automated item generation is not seen as a worthy candidate for the expense of resources (both time and money), comes from comments made by a psychometrician at a company providing test development services to the internet technology sector. David Foster (personal communication, September, 2001) noted that while their BenchMark test development software (Galton Technologies, 2001) does allow for what are termed variants, this is a software function whereby the item writer can make a copy of the item and then change different aspects by hand. The system does facilitate tracking of similar items, and prevents variants from appearing on the same test form, but the item creation itself is not in any sense automatic.

Foster observed that a lack of technology does not limit the implementation of totally automated item generation methodologies in the internet technology field, but that instead a lack of need does. He pointed out that for tests in which only a few items (for
example, from 4 to 15) are needed for any given objective, the creation of a system for automatically generating dozens or hundreds of variations does not seem warranted. It certainly seems plausible that this scenario is also played out in other large-scale testing situations, thus partly explaining the lack of documented techniques.

It should be noted, however, that progress is being made in the area of simulations for tests in the licensing and certification arena. Due to this paper’s focus on multiple-choice items, innovative work in the areas of medical licensing and architecture, to name two examples, will not be reviewed. One example that may be briefly mentioned, however, is a component of Step 3 of the United States Medical Licensing Examination (Federation of State Medical Boards of the United States, Inc., and the National Board of Medical Examiners, 1999). This test contains simulations in which examinees are presented with a clinical situation which they must manage, and they determine what diagnostic information to obtain and how to subsequently treat the patient. Based on the examinee’s actions, the computer presents information on how the patient’s condition changes over time, which can influence subsequent steps taken by the examinee.

The reader is also referred to Sumner, Haben, & Rovinelli (1999), Sumner, Truszczynski, & Marek (1995, 1998), and Rivonelli, Sumner, Marek, & Truszczynski (U.S. Patent No. 6,246,975; 2001; first author’s last name is reproduced as it was misspelled in patent documentation) for descriptions of a formal model of family medicine developed for the American Board of Family Practice. This model underlies a computer-based simulation approach that is expected to be pilot-tested in 2002, with operational use beginning in 2003 (W. Sumner, personal communication, December, 2001). For descriptions of the development of an architect licensing exam, an operational
program which utilizes simulations with a generative base, the reader is referred to Bejar (1991, in press) and Bejar and Braun (1999).

Discussion

Automatic item generation methodologies have clearly evolved over the past thirty years. Initial operational approaches whose roots lay in efforts to better define domains of content have been joined by methods based on cognitive principles. Continua proposed by Bejar (in press) and others, as depicted in Figure 1, are useful in illustrating the full range of item generation procedures that are available. They also provide a framework within which to discuss issues that need to be kept in mind during a consideration of the future of item generation.

Turning first to functional techniques, we see that attention to two of the three methods described in the early approaches section—linguistic transformations and facet theory—appears to have waned, both in research venues and in operational implementations. However, the remaining approach, that of item forms, does appear to live on in operational settings under the guises of item shells and item modeling. The former is more strictly an operational procedure in which existing items are turned into shells, into which variations in key words or phrases may be substituted. The latter is somewhat more construct-driven, though still based more in an analysis of content than in cognitive processes.

The item forms-based approaches of item shells and item modeling, as used intermittently at some medical licensing testing agencies, serve as tools to help human item writers structure their activities. They are not automated processes, though use of
computer in these approaches certainly seems viable. As such, these techniques are not strictly algorithmic; constraints are not built into a system that can generate items given certain variables. Due to the unstructured use of replacement sets and the lack of cognitive modeling in these approaches, item characteristics cannot be predicted.

Although some maintain that not reducing the need for item review and pretesting render automatic item generation activities less valuable (e.g., Wainer, in press), it can certainly be argued that functional approaches can be of great benefit to testing agencies that have limited resources. For example, credentialing agencies often have licensed or certified professionals write items. Providing guidelines for how to turn one good item into several new items through the use of replacement sets can be quite helpful in facilitating the item development process. It relieves the item writers of the burden of creating the item from scratch and provides them with a framework within which to begin their work.

Moving now to model-based approaches, we see that strides have been made in utilizing cognitive psychology principles to automatically generate items in a manner informed by thorough understanding of the construct being measured. Implementation of a systematic approach such as Embretson’s (in press) lays the foundation for the generation of items whose characteristics can be predicted. Efforts such as these are invaluable for helping us better understand the construct under consideration and to ensure that entire assessment system—including item generation and scoring—will be designed to facilitate its measurement (see also Mislevy, Steinberg, & Almond, in press). However, much of the work done in this area is related to aptitude tests, not achievement tests. It remains to be seen how much can be extended to large-scale achievement and
credentialing tests, and whether a theory of performance for these domains can be developed (Bejar & Yocom, 1991).

Comments made by Bejar in 1993 still seem relevant now, as regards the potential for using automatic item generation for applications in achievement testing (and by extension, credentialing exams):

A generative approach to achievement testing remains to be developed. Part of the challenge no doubt is due to the elusive nature of the concept of achievement. . . . A generative approach that is consonant with current thinking on the nature of learning . . . is likely to be different from the approaches we have discussed for the assessment of generic abilities because ranking individuals would not be the focus of measurement. . . . The selection of questions would not be based on difficulty, but rather on the degree of information that an answer to a question would provide in updating the several hypotheses under consideration to account for a student state of knowledge. (p. 343)

When fully automated item generation is able to be implemented, questions still remain. For example, the possibility exists of narrowing the construct being measured by only using items that we know how to generate (Enright & Sheehan, in press). Similarly, item generation could, if not used appropriately, just be used to create large quantities of poor items. Bennett (1999), in his review of item generation as an example of using technology to improve assessment, made the following observation:
The major issue is ensuring thoughtful generation and use of item variants. If used thoughtfully, this technology can facilitate learning transfer and its assessment, because we can efficiently generate questions that vary from one another in cognitively principled ways. If used thoughtlessly, however, it will do nothing more than allow us to create bad tests rapidly by generating many variants of questions that don’t test anything important. We can avoid this fate best by closely tying item generation to principled test design—that is, by working carefully to define what we want to test, why we want to test it, and how we plan to do it. (p. 7)

Even when we are able to generate high-quality items effectively, the problem of how to score automatically generated items remains, though research by Bejar et al. (2001), Dresher and Hombo (2001), Hombo and Dresher (2001), and Wright (in press) offers promise in this regard. Security may also be of concern if isomorphs (items identical in psychometric functioning; Bejar, in press) become well-enough known to examinees that their functioning changes as a result of this pre-knowledge (Bejar et al., 2001). The question of “yield,” or how many ultimately usable items will be generated, is also an issue when considering whether to undertake labor-intensive automatic item generation procedures (Lewis, in press).

Implications for the Medical Council of Canada

Systematic approaches such as item forms and item modeling appear to have facilitated item writing for some licensing tests in the medical field. Though these techniques have not necessarily been automated, it seems plausible to do so without
extensive efforts. These techniques seem eminently suitable for the MCC. Another option would be for the MCC to work with the Chauncey Group or other groups that have developed software to produce items. Of course, there is the possibility too of generating new software that could be tailored to the item writing needs of the MCC.
References


Goeters, K.-M., & Lorenz, B. On the implementation of item generation principles in the design of aptitude testing in aviation. In S. H. Irvine & P. Kyllonen (Eds.), *Item generation for test development*. Mahwah, NJ: Erlbaum.

ability and achievement test items. Educational and Psychological Measurement,
27, 569–580.

Mahwah, NJ: Erlbaum.

multiple-choice test items. Evaluation and the Health Professions, 12, 97–106.

Domain-referenced testing (pp. 5–15). Englewood Cliffs, NJ: Educational
Technology Publications.

Hively, W., II, Patterson, H. L., & Page, S. H. (1968). A “universe-defined” system of

automatic item generation under NAEP-like data conditions. Paper presented at
the annual meeting of the National Council on Measurement in Education,
Seattle, WA.

Hornke, L. F. (in press). Item generation models for higher order cognitive functions. In
S. H. Irvine & P. Kyllonen (Eds.), Item generation for test development.
Mahwah, NJ: Erlbaum.

Hornke, L. F., & Habon, M. W. (1986). Rule-based item bank construction and
evaluation within the linear logistic framework. Applied Psychological


