A Testlet Assembly Design for the Uniform CPA Examination

Richard Luecht

Terry Brumfield

University of North Carolina at Greensboro

Krista Breithaupt

American Institute of Certified Public Accountants

April 2002

Paper presented at Annual Meeting of the National Council on Measurement in Education, New Orleans, April 2002.

Introduction

This paper describes some practical test development considerations related to the design and implementation of a multistage, computer-adaptive testlet version of the Uniform CPA Examination, slated for implementation in 2003. The paper further discusses the use of automated test assembly (ATA) procedures in an operational context to produce large numbers of adaptive testlets over time, as well as issues related to the maintenance of item banks and the inclusion of accounting simulations on the new examination.

The American Institute of Certified Public Accountants (AICPA) prepares the Uniform CPA Examination. It is the licensing examination used by 54 licensing jurisdictions to grant entry into the CPA profession. The AICPA has spent the past several years preparing to computerize the Uniform CPA Examination. This has included an extensive research agenda. This paper focuses on one aspect of that agenda.

The Uniform CPA Examination is currently administered in paper-and-pencil format and has four sections: (1) audit, (2) accounting & reporting, (3) financial accounting & reporting, and (4) legal & professional responsibilities. Examinees typically take the four paper-and-pencil sections over a two-day period (15.5 hours). The computerized examination will have some perceptible changes in format and content, as well as a reduction in the total testing time.

Based on an extensive practice analysis of the CPA profession (Norris, Russell, Goodwin, and Jesse, 2001) and related research conducted by the

AICPA, the new computerized examination is anticipated to differ from its predecessor in three ways. First, the content blueprint for the examination is being reorganized and revised to emphasize four new areas: (a) audit and attestation; (b) financial accounting and reporting; (c) taxation and government regulations; and (d) business environment. Second, the skills measured by the examination have been expanded to include written communications, integrated financial and accounting analysis tasks, and research. The addition of performance-based accounting simulations that incorporate features such as word processing, spreadsheet functionality, and on-line capabilities to search the authoritative accounting literature will make measurement of these additional skills possible (Devore, 2002). Third, the new examination will use adaptive testing technology to allow the examination to be shortened by several hours.

These changes in the Uniform CPA Examination have some obvious implications for test design and development. Four key implications and issues addressed in this paper are: (1) the anticipated need to use multistage, adaptive testlets to improve the efficiency of the examination; (2) the need to use automated test assembly procedures to produce large quantities of high quality testlets that meet the Uniform CPA Examination specifications; (3) present uncertainties related to the expected state of the item banks at implementation in 2003; (4) and test development issues related by the incorporation of performance-based accounting simulations into the new examinations.

The Multistage, Adaptive Testlet Delivery Model

An extensive research effort over the past few years, funded by the AICPA, has explored various test delivery models for the computerized Uniform CPA Examination. This research considered a wide range of operational, financial, and psychometric criteria. The test delivery models under consideration ranged from fixed test forms to content-balanced, computer adaptive tests. For a variety of operational, security, and psychometric reasons, one of the more promising configurations that emerged was a three-stage, adaptive testlet delivery model. Luecht (2000) has termed this type of delivery model a "1-3-3 module" CAST configuration. The CAST configuration is depicted in Figure 1.

The seven **testlets**[1] shown on each of the gray rectangles in Figure 1 jointly represent a three-stage test, where the examinee only gets one of the available testlets per stage. Reading from left to right, there is one testlet assigned to the first stage (1M), three testlets assigned to the second stage (2E, 2M, and 2H), and three testlets assigned to the third stage (3E, 3M, and 3H); thus the label, "1-3-3 CAST configuration". The letters "E", "M", and "H" denote the average difficulty of the testlet (E=easy, M=moderate, H=hard).

The prescribed routes through the seven testlets are indicated by the solid and dashed lines. Solid lines are used for the primary pathways (i.e., those

[1] Luecht & Nungester (1998) called the collection of items at each stage a "module" to avoid other connotations of the term "testlet". In this paper, the terms testlet and module are considered to be synonymous with each other.

routes most likely to be taken by examinees who perform as expected). Dashed lines denote the less-common secondary pathways. For example, the centermost pathway would include the following sequence of testlets: 1M→2M→3M. Note that some pathways are precluded (e.g., an examinee would not be able to move from testlet 2E to 3H). Extreme changes in ability are unlikely by chance and such unexpected performance that would probably be flagged as "aberrant". This configuration simply implements a policy to eliminate that possibility.

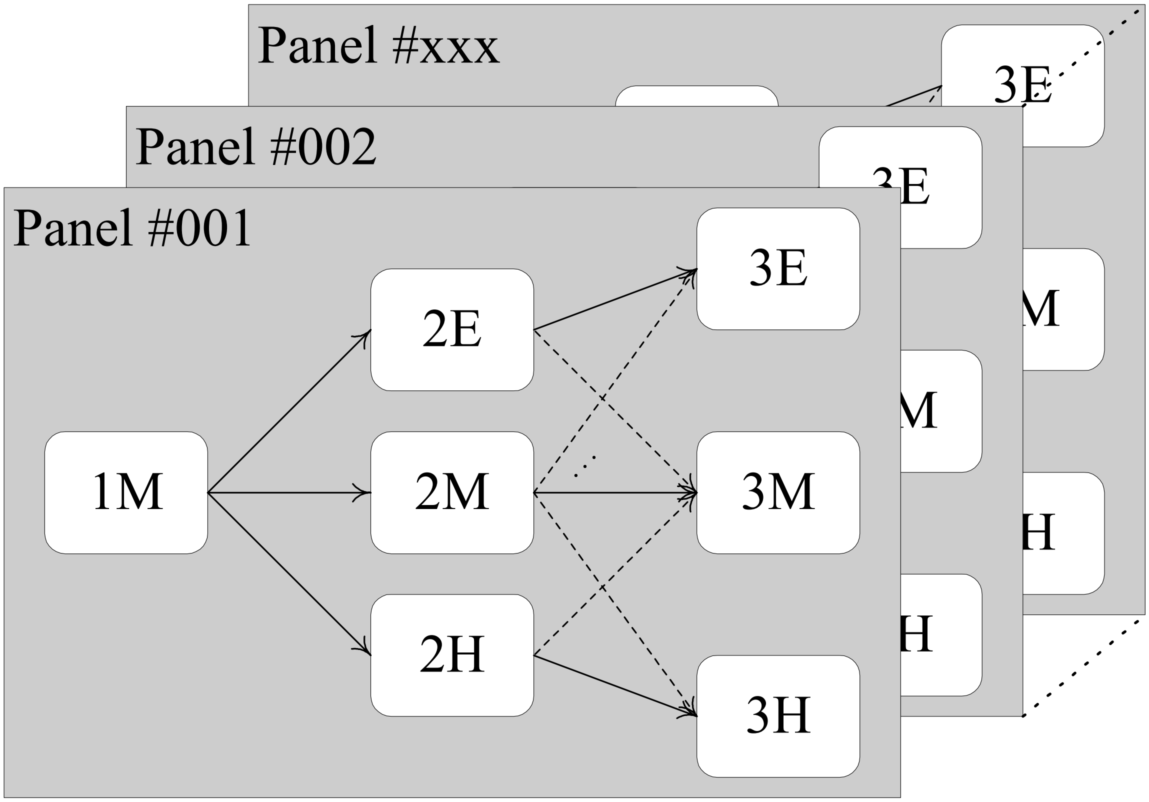

Figure 1. Designing a 1-3-3 CAST Configuration with Multiple Panels

The seven testlets and the associated routing rules are packaged together units called **panels** (Luecht and Nungester, 1998). A panel is a formal database *object* (i.e., an identifiable test unit that can be directly accessed by software for any purpose). Figure 1 depicts multiple panels ("Panel #001", "Panel #002",…, "Panel #xxx"). These multiple panels can be used just like multiple test forms to discourage cheating.

Each panel is uniquely defined and has its own identification number in the database. Every panel also contains a different collage of seven testlets, with each testlet assigned exactly to one and only one of the seven positions within the panel. A specific set of routing rules also needs to be determined, based upon the particular statistical characteristics of the seven testlets that comprise the panel (see Scoring Panels). Using standard jargon from modern object-oriented design (OOD), each panel is fully *encapsulated*; that is, each panel has all of the information needed to administer and score itself. Therefore, panels can be randomly assigned to examinees just like randomly assigned fixed test forms. The difference is that panels know how to adaptively administer themselves.

There are numerous factors that affect the measurement quality of the testlets within each panel and the scores that result from this type of multistage, adaptive testlet design. For example, the size and characteristics of the item bank (e.g., the distribution of item difficulty and discrimination by content) has a direct impact on the possible breadth of test information that can be spread across the ability scale. Test information cannot be magically generated within various regions of a proficiency score scale. A complete inventory of test items having reasonable discriminating power and appropriate difficulty required to achieve a specified level of reliability at any point on the proficiency scale.

The number of items per stage (testlet size) can directly affects the capability of the panel to adapt to examinees having more extreme abilities. Moreover, the testlet size per stage has a direct impact on the number of items needed per panel. This, of course, affects overall exposure risks by possibly limiting the number of panels that can be constructed from a finite item bank.

Table 1 shows ten different specifications for a three-stage panel with the testlets ranging in size from 15 to 30 items. Each set of specifications assumes seven testlets and a total test length of 60 items (i.e., Stage 1 + Stage 2 + Stage 3 = 60 items). The leftmost three columns show the actual testlet size per stage. The next three columns show the total number of items required per stage for the 1-3-3 CAST configuration. For example, referring to the first data row in the table, the testlet sizes are 15, 15, and 30. The total numbers of items per stage are: (a) 15 × 1 = 15 at Stage 1; (b) 15 × 3 =45 at Stage 2; and (c) 30 × 3 =90 at Stage 3. The rightmost column shows the total panel size (i.e., the total number of items needed to build the panel). For the first row, the total is 15 + 45 + 90 = 150 items. Only multiples of five are shown for the testlet sizes (15, 20, 25, and 30 items).

It should be apparent from Table 1 that putting fewer items in the later stages has some obvious benefits in terms of minimizing the total number of items required per panel. Luecht & Nungester (1998) discussed how using smaller testlets in later stages also tended to allow test developers to better target the information provided by some of the latter-stage testlets toward the extremes of the ability distribution (subject, of course, to the availability of items in the item bank).

Table 1. Panel Size as a Function of Testlet Size (1-3-3 CAST Configuration)

Testlet Size Per Stage			Item Counts per Stage			Total Panel
Stage 1	Stage 2	Stage 3	Stage 1	Stage 2	Stage 3	Item Count
15	15	30	15	45	90	150
20	15	25	20	45	75	140
25	15	20	25	45	60	130
30	15	15	30	45	45	120
15	20	25	15	60	75	150
20	20	20	20	60	60	140
25	20	15	25	60	45	130
15	25	20	15	75	60	150
20	25	15	20	75	45	140
15	30	15	15	90	45	150

One implication of reducing the total number of items needed per panel is to reduce item exposure risks. That is, given an item bank of fixed size, more panels can be created if fewer items are needed, overall, per panel. Increasing the number of available panels, in turn, reduces certain exposure risks due to concerted efforts by examinees to memorize test materials (Luecht, 1998b). For example, using the counts in Table 1, an item bank of 1,000 items could produce six to eight unique panels. Ten to twelve panels could be built from a 1,500-item bank and thirteen to sixteen panels could be created from an item bank containing 2,000 items. Of course, allowing systematic [i.e., controlled] item overlap among different panels or even within panels[2] would further increase the viable number of panels.

Scoring Panels

One advantage of packaging the testlets and routing rules together as a panel is that scoring can be simplified in terms of needed data and computational scoring functionality required by the test delivery computer software. There are three types of scoring needed for this type of multistage, adaptive test: (1) scoring the individual items and accounting simulations; (2) cumulatively scoring the testlets for purposes of adaptively selecting the testlets in the subsequent stage; and (3) final scoring for purposes of reporting a score and associated pass/fail decision. The latter type of scoring is not a concern in terms of what happens at the test center, provided that accurate data are captured and stored. That is, real-time final scoring, or any type of immediate reporting of Uniform CPA Examination results at the test center, are simply not options for the AICPA, the National Association of State Boards of Accounting, or any of the jurisdictional boards that issue a CPA license. There are serious legal and ethical risks associated with releasing unverified scores for any high-stakes examination. The record of the examination session and all of the response data need to undergo numerous quality assurance steps to verify their integrity, before releasing scores or other results.

[2] A single examinee only sees one testlet per stage. Therefore, testlets assigned to the same stage can, for all practical purposes, share items as needed.

Nonetheless, real-time scoring by the test delivery software is required to allow a panel to adapt its testlets to an examinee's proficiency. Although IRT scoring (maximum likelihood or Bayes estimation) is certainly possible, Luecht & Nungester (1998) empirically demonstrated that number-correct scoring is probably sufficiently accurate for purposes of selecting testlets. Number-correct scoring certainly simplifies the amount of data needed at the test center and complexity of scoring and testlet selection routines that need to be supported by the test delivery software.

The basic implementation of number-correct scoring in this type of multistage, adaptive test is to incrementally compute the upper and lower bounds for the number-correct scores associated with various combinations of testlets that reflect a particular routing decision to the next stage. For example, in adaptively transitioning from Stage 1 to Stage 2 in Panel #001 (Figure 1), we would need to know three *pairs* of values, corresponding to the upper and lower number-correct scores on **Testlet 1M**: $\{X_{L(1)}, X_{U(1)}\}$ to move to Testlet 2E; $\{X_{L(2)}, X_{U(2)}\}$ to move to Testlet 2M, and $\{X_{L(3)}, X_{U(3)}\}$ to move to 2H. Since no body can score below zero or above the maximum possible points, we can set the lower bound for the easy route at $X_{L(1)}=0$ and the upper bound for the hard route at $X_{U(3)}=n_j$, where n_j is the size (or maximum possible points) for the testlet. Using the lower and upper boundary pairs of values merely generalizes the selection routine.

Table 2 shows a sample "routing" data table for two panels, P000001 and P0000002. This type of table could be stored in a database and would include all of the score routing information for all active panels. Optionally, separate tables could be created for each panel.

Table 2. A Sample "Routing" Table for Two Panels

Panel ID	Current Stage	Route ID	Testlet History	Lower X_U	Upper X_U	Select Next Testlet
P000001	1	R0000001		0	0	1
P000001	2	R0000002	1	0	6	2
P000001	2	R0000003	1	7	13	3
P000001	2	R0000004	1	14	20	4
P000001	3	R0000005	12	0	15	5
P000001	3	R0000006	12	16	26	6
P000001	3	R0000007	13	7	13	5
P000001	3	R0000008	13	14	26	6
P000001	3	R0000009	13	27	33	7
P000001	3	R0000010	14	14	23	6
P000001	3	R0000011	14	24	40	7
P000002	1	R0000012		0	20	1
P000002	2	R0000013	1	0	6	2
P000002	2	R0000014	1	7	13	3
P000002	2	R0000015	1	14	20	4
P000002	3	R0000016	12	0	15	5
P000002	3	R0000017	12	16	26	6
P000002	3	R0000018	13	7	13	5
P000002	3	R0000019	13	14	26	6
P000002	3	R0000020	13	27	33	7
P000002	3	R0000021	14	14	23	6
P000002	3	R0000022	14	24	40	7

The "Panel ID" is included for look-up purposes. The "Current Stage" is the present state of the test (e.g., "1" indicates that a testlet for Stage 1 must be selected). The "Testlet History" column contains a concatenated string of previously administered testlets within the panel, indexed by the integers 1, 2, 3,….,7, where 1=Testlet 1M, 2=Testlet 2E, 3=Testlet 2M,…, 7=Testlet 3H. This is the list of testlets that provides the score to be used for the routing. The "Lower X_L" and "Upper X_U" columns contain the upper and lower bounds for the adaptive selection. When the current score satisfies, $X_L \leq X \leq X_U$, the score routing rule fires as "true" and the testlet in the "Select Next Testlet" column is given.

Item response theory (IRT) is used determine the actual values. We start by locating one or more points on the proficiency scale, θ_d, each of which corresponds to a particular decision point for routing examinees (i.e., for choosing between which of two possible testlets to administer next). For example, given Testlet 1M, we need to decide between Testlet 2E and 2M (easy vs. moderate) or between Testlet 2M and 2H (moderate vs. hard). Given a particular decision point, θ_d, and the IRT item parameters for a set of k testlets administered up to that point, ξ_i, i=1,…,n_j, j=1,…,k, the corresponding estimated true-score point is $X_d = \sum_{j=1}^{k} \sum_{i=1}^{n_j} P(\theta_d ; \xi_i)$, where $P(\theta_d,\xi_i)$ is the item response function for a particular IRT model. The computed value can be rounded to approximate a number-correct integer score, if needed.

Fortunately, all of these computations can be done for each panel, before it is released for use. Once the number-correct routing scores are determined for a panel, the IRT data are no longer needed. A number-correct scoring function, the routing table, and a simple look-up mechanism, are sufficient to allow each panel to adapt itself.

There are [at least] two methods for locating the routing points on the ability scale. The Approximate Maximum Information (AMI) method empirically determines the cut point(s) using the cumulative test information function for the previously administered testlets and the testlets at the current stage. This method mimics an adaptive test, by choosing the testlet likely to provide maximum information about a examinee, given a current provisional score.

Under the AMI method, the cumulative test information functions (TIFs) are evaluated pair-wise for adjacent testlets within each panel. The AMI method merely finds the optimal decision point on the θ scale for selecting between one testlet or the other, using a maximum information criteria similar to any CAT. That is, the intersection of the TIFs corresponds to the decision point insofar as selecting one or the other testlet. This intersection is relatively easy to find using standard numerical analysis root-finding techniques (e.g., using numerical bisection to find the value of θ at which the information functions are equal because they intersect with one another). For example, assuming the administration of testlet 1M in Panel #001 (see Figure 1), we would like to find two routing points: θ_1 corresponding to the intersection of the TIF curves, $I(1M + 2E) \cap I(1M + 2M)$, and θ_2 corresponding to the intersection of the TIF curves, $I(1M + 2M) \cap I(1M + 2H)$.

Once we locate those two points, we can compute the corresponding estimated true-score values on the test characteristic surface for testlet 1M; that is, we compute $X_1 = \sum_{i \in 1M} P(\theta_1 ; \xi_i)$ and $X_2 = \sum_{i \in 1M} P(\theta_2 ; \xi_i)$. This process of determining the score routing points can be repeated for each of the possible routes in the panel. The results can be then be tabled and packaged as part of the panel data. Note that the routing points, θ_1 and θ_2, and the approximate number-correct cut points, X_1 and X_2, will probably differ from testlet to testlet, unless the TIFs and associated test characteristic curves for the replicated testlets on multiple panels are virtually identical.

A second method of determining cut points is the Defined Population Intervals (DPI) method. The DPI method can be used to implement a policy that specifies the relative proportions of examinees in the population expected to follow each of the three primary routes through the panel. For example, if we determined, as a matter of policy, that we wanted approximately equal proportions of examines in the population exposed to the three primary pathways in our 1-3-3 panel (i.e., 1M+2E+3E, 1M+2M+3M, and 1M+2H+3H), we could find the ability scores associated with the 33rd and 67th percentiles of the cumulative distribution of θ. Assuming θ to be normally distributed ($\mu=0, \sigma^2=1$) the routing points would be $\theta_1 = -0.44$ and $\theta_2 = 0.44$, which can easily be verified from a standard table of values for the unit normal distribution. The approximate number-correct routing scores could then be determined as outlined above for the AMI method.

Bundling Panels

Each panel can be bundled as a data base object. Each panel contains four types of information: (1) a unique identifier; (2) a list of testlets assigned to the bundle; (3) a "map" which assigns the testlets to one of the seven positions within the panel; and (4) a score routing table (see previous section). Of course, a testlet-to-item look-up table and the resource information for the individual items (text, graphics, answer keys, etc.) would also be part of the overall database accessed by the test delivery software.

However, it should be clear that when a particular panel is selected, all of the information needed to administer the testlets and items for that panel is contained (directly or by reference) within the panel "wrapper." In addition to facilitating quality assurance, the panel concept also allows panels to be blocked for certain examinees (e.g., in cases of retesting, previously seen panels can be blocked, including other panels with substantial numbers of overlapping items).

Using Automated Test Assembly to Construct Panels

Automated test assembly (ATA) involves the use of mathematical optimization procedures to select items from an item bank for one or more "test forms," subject to multiple constraints related to the content and other qualitative features. van der Linden (1998) presents an excellent overview of the most popular ATA heuristics and mathematical programming techniques.

A simple example may help for purposes of illustration. We start by specifying a quantity to minimize or maximize. This quantity is called the *objective function* and can be formulated as a mathematical function to be optimized by linear programming algorithms or heuristics. *Constraints* are imposed on the solution, usually reflecting the content blueprint or other qualitative features of the items that we wish to control (e.g., word counts). The constraints are typically expressed as equalities (exact numbers of items to select) or inequalities (upper or lower bounds on the number of items to select).

For example, suppose that we want to maximize the IRT test information at a fixed cut point, denoted θ_0, with a fixed test length of 20 items. We need to define a binary decision variable, x_i, i=1,…,I that indicates that item i is selected (x_i=1) or not (x_i=0) from the item bank. Given this decision variable, the objective function to be maximized is the IRT test information function for the selected items; that is,

$$I(\theta_0) = \sum_{i=1}^{I} I(\theta_0, \xi_i) x_i \qquad (1)$$

where ξ_i denotes the item parameters from the item bank, i=1,…,I (e.g., $\xi_i = \{a_i, b_i, c_i\}$ for the three-parameter logistic model). Now, suppose that we have two content areas, C_1 and C_2, and wish to have at least 5 items from content area C_1 and no more than 10 items from content area C_2. This ATA problem can be modeled as follows:

maximize $\sum_{i=1}^{I} I(\theta_0, \xi_i) x_i$ (maximum information) (2)

subject to:

$$\sum_{i \in C_1}^{I} x_i \geq 5 \quad \text{(constraint on } C_1\text{)} \qquad (3)$$

$$\sum_{i \in C_2}^{I} x_i \leq 10 \quad \text{(constraint on } C_2\text{)} \qquad (4)$$

$$\sum_{i=1}^{I} x_i = 10 \quad \text{(test length)} \qquad (5)$$

$$x_i \in \{0,1\},\ i=1,\ldots,I. \quad \text{(range of variables)} \qquad (6)$$

It is relatively straightforward to extend these basic ATA procedures to a multistage, adaptive testlet environment like CAST (Luecht, 2000). For the type of ATA problem implied by the CAST model described in the previous section, it is useful to employ what Luecht called the “bottom-up” CAST design strategy. This bottom-up strategy essentially treats the test assembly process as a simultaneous, multiple objective function optimization problem where we are simultaneously building one or more versions of seven different tests (i.e., the seven testlets in each panel). Accordingly, implementing this approach requires separate test specifications (statistical targets and content constraints) for each testlet.

Specific to the 1-3-3 CAST configuration shown in Figure 1, we need to design seven independent test information function targets, where the test information function (TIF) was given in Equation 1. That is, we must specify seven unique TIF targets at multiple θ values, approximating seven TIF target curves.

Figure 2 shows a conceptual picture of seven target information functions curves, each corresponding to one of the testlet positions shown in Figure 1. The location of the peak of the TIF curve for each target can vary within and across stages, corresponding to desired changes in the average difficulty of the testlets. The amount of information targeted per testlet may also vary depending up the testing stage and the availability of informative items in the item bank.

In addition to statistical specifications, separate content specifications are required for each of the three stages. That is, there will be one set of content specifications for Stage 1, representing the content requirements for testlet 1M. There will be a second set of content specifications for the three Stage-2 testlets (2E, 2M, and 2H). The implication is that, although those three testlets will be targeted to have different statistical characteristics, they are required to meet the same content specifications within Stage 2. Similarly, a third set of content specifications is required for the three testlets at the third stage (3E, 3M, and 3H).

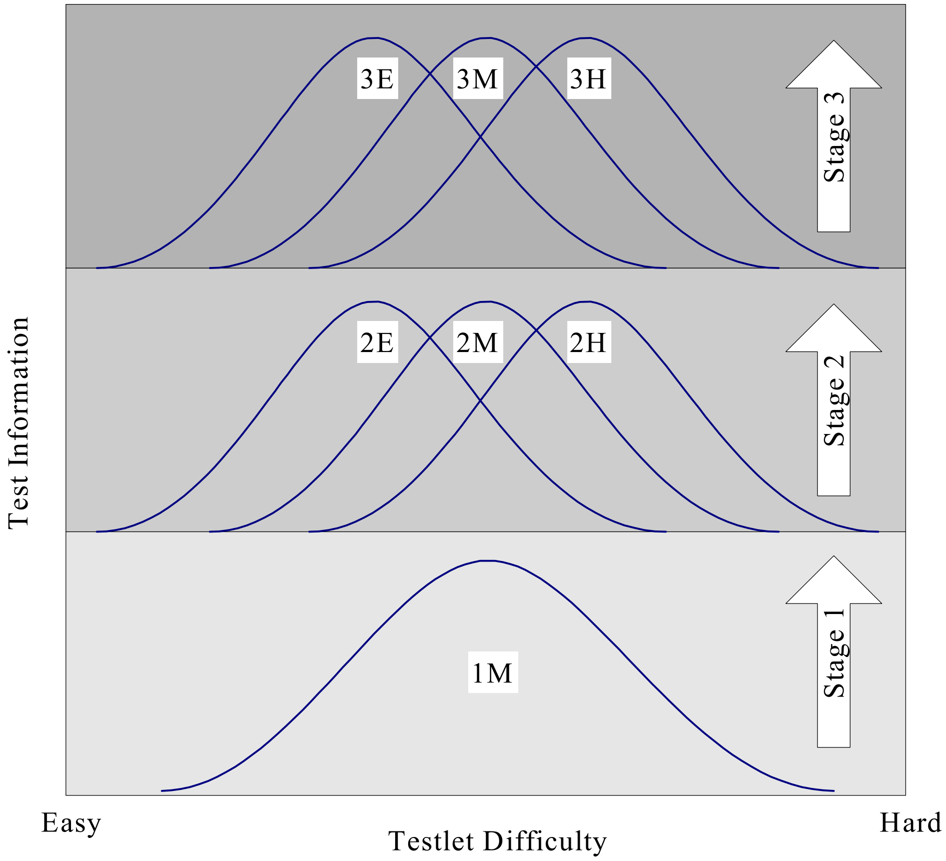

Figure 2. Seven Target Testlet Information Functions for the 1-3-3 CAST Configuration

Based upon extensive work by AICPA test development staff and various subcommittees of content experts, the new blueprint for the computerized UNIFORM CPA EXAMINATION is likely to include slightly fewer than 150 content skills outline (CSO) codes. Furthermore, if the 1-3-3 CAST configuration shown in Figure 1 is adopted (Goldman, personal communication), it appears that the CSOs can be proportionally allocated to each of the three stages. This is an important test-design aspect of the proposed Uniform CPA Examination that greatly facilitates test assembly. If content were not able to be allocated to the individual stages, the bottom-up strategy could not be employed (Luecht & Nungester, 1998; Luecht, 2000).

Finally, we need to specify the length of each testlet and content requirements for each stage in the panel. For the present research purposes, a working assumption has been that each of the four new computerized Uniform CPA Examination sections will be, on average, 60 items in length with 15 to 25 items per any stage. Therefore, under the bottom-up CAST assembly strategy, each of the seven testlets per panel can be treated as a separate test assembly project of approximately 20 items.

These specifications will be used to simultaneously solve seven optimizations models, one for each testlet position in the panel (see Figure 1) and produce multiple, parallel versions of each testlet. Once we have the multiple replications of the seven testlets, we can assemble them, by any number of viable combinatoric means[3], to create different versions of the panels. The unique list of constructed testlets, their assignments to the pathways and stages within each panel, and the associated scoring and routing rules can then be packaged as part of each panel.

A somewhat minor challenge remains in simultaneously solving seven optimization models. That is, we need to devise and solve an ATA model that allows us to simultaneously build multiple replications of the seven individual testlets: 1M, 2E, 2M, 2H, 3E, 3M, and 3H (see Figure 1), each meeting a potentially different set of statistical and content constraints. Research on this aspect of the AICPA's agenda has been somewhat limited by the capability of existing software to meet this specific challenge (Hambleton et al, 2002; also see footnote #2).

In theory, this type of problem can be readily solved using dedicated ATA optimization heuristics (Luecht and Nungester, 1998; Luecht, 1998). However, working computer software that implements this type of 1-3-3 CAST model, using a bottom-up strategy, has not yet been completed and pre-existing CAST test assembly software is not particularly useful for this application. The Appendix presents an algorithm that a new ATA software engine, currently under construction, will employ. This algorithm is based upon Luecht's (1998) normalized weighted absolute deviation heuristic (NWADH).

In operational practice, it is expected that linear programming will be eventually used by the AICPA, since it provides more exact results than heuristics (van der Linden, 1998). Solving this type of multi-target problem with linear programming is somewhat challenging—but not insurmountable. The authors are exploring the use of CPLEX (ILOG, 2002), a software suite that conveniently handles large-scale optimization problems using the branch-and-

[3] Overlap among the panels can be explicit handled via overlapping testlets. Exposure risks for individual testlets and panels can analytically computed or approximated by using simulated response data, based upon characteristics of the target population.

bound algorithms and requisite relaxation techniques to deal with infeasibilities in the linear programming solution (see, for example, Timminga, 1998).

AICPA Item Banks

The likely properties of the AICPA item banks at implementation in 2003 are unknown at this time. Although the AICPA has contracted for supplemental item writing with several qualified organizations to significantly increase the size of their item banks, there remain a number of practical issues to resolve. These include: (a) pretesting the items to obtain usable item statistics; (b) classifying and coding the items for the new item banks; (c) implementing inventory controls to ensure that the quality, types, and characteristics of items produced will be optimal for the type of test assembly model described in the previous section; and (d) estimating exposure risks to the integrity of the item banks over time.

The unknown characteristics of the AICPA item bank (by 2003) present some potential risks. Hambleton, Jodoin, and Zenisky (2002) clearly demonstrated the limitations of implementing an adaptive testlet paradigm in a mastery-testing context, if the item banks are restricted to mirror the characteristics that were used by Hambleton et al in their simulations. It is hoped that the new item banks will provide a much wider distribution of item difficulty and better discrimination across the ability scale. However, the extent to which the new item banks will approach the ideal is difficult to predict with any confidence.

In order to predict the characteristics of the item bank, it obviously is necessary to have statistics. Pretesting efforts are planned for the mid- to latter part of 2002 to provide response data for as many of the new items as possible. In addition, a number of psychometric research studies are underway to explore ways to improve the stability of IRT item parameter estimates based on small sample pretest data and under the proposed, adaptive-testing paradigm.

Coding the items according to the new content classifications is progressing. Content experts and test development staff at the AICPA are classifying and coding the current item bank and all new items based upon the new Uniform CPA Examination content and skills outline (CSO). However, the available supply of items projected to be in the bank with respect to specific demands (i.e., in terms of content and statistical specifications for test assembly) is difficult to predict at this time. Research is also underway to apply mathematical optimization techniques to the broader issue of inventory control for the item banks to support ongoing maintenance of the item banks, beyond 2003 (van der Linden, personal communication). At present, that research is strongly dependent on the [unknown] accuracy of certain current assumptions and predictions.

In the long term, the majority of risks to the item banks can be mitigated by simply increasing the size of the item banks, maximizing the production of quality items, and by effectively controlling item exposure. Item exposure constraints can be implemented within the ATA/CAST framework to achieve any needed restrictions on item reuse across panels. For example, constraints can be introduced during test assembly so that no singular panel shares more than 20 percent of its items with any other panel. This could be done at either the item or testlet level. Optionally, constraints could be placed on the proportion of item reuse allowed when replicating the testlets, themselves. Since the adaptive mechanism operating within each panel is not maximizing test information, *per se*, there is no need to impose item-level exposure controls under CAST (Luecht and Nungester, 1998). Nonetheless, assessing exposure risks requires knowledge about the examinee population. Because of various political limitations that prohibit the AICPA from having access to certain examinee data, it is difficult to make credible predictions about the characteristics of that population, now or in 2003.

Performance-Based Simulations

Devore (2002) described the accounting simulations that are being developed for the new Uniform CPA Examination. The implications for test development are: (a) producing sufficient numbers of high-quality simulations to reduce risks of cheating; (b) pilot testing the accounting simulations; (c) determining efficient, yet accurate, automated methods of scoring the rather complex performance exercises. The test assembly implications are less serious.

For the rollout in 2003, a very small number of accounting simulations will be allotted, virtually with certainty, to several of the new Uniform CPA Examination sections. As a result, current thinking by test development staff is that the simulations will be systematically attached to the panels, based upon their apparent content.

Acknowledgements

The authors wish to thank the American Institute of Certified Public Accountants for their support of this research, especially Gerald Melican and Craig Mills. In addition, the authors are indebted to Ronald Hambleton and Wim van der Linden for their insights on various technical aspects of this work.

References

Devore, R. (April, 2002). *Considerations in the Development of Accounting Simulations*. Paper presented at the Annual Meeting of the National Council on Measurement in Education, New Orleans.

Goldman, A. Personal communication.

Hambleton, R. K., Jodoin, M, & Zenisky, A. (April, 2002). *Impact of selected factors on the psychometric quality of credentialing examinations administered with a sequential testlet design*. Paper presented at the Annual Meeting of the National Council on Measurement in Education, New Orleans.

ILOG, Inc. (2002). *ILOG CPLEX Suite* [Computer programs]. Mountain View, CA: ILOG, Inc.

Luecht, R. M. (1996). *CASTISEL* [Computer software]. Author.

Luecht, R. M. (1998). Computer-assisted test assembly using optimization heuristics. *Applied Psychological Measurement, 22* 224-236.

Luecht, R. M. (April, 1998). *A framework for exploring and controlling risks associated with test item exposure over time*. Paper presented at the Annual Meeting of the National Council in Measurement in Education, San Diego, CA.

Luecht, R. M. (April, 2000). *Implementing the Computer-Adaptive Sequential Testing (CAST) Framework to Mass Produce High Quality Computer-Adaptive and Mastery Tests*. Paper presented at the Annual Meeting of the National Council on Measurement in Education, New Orleans.

Luecht, R. M. Personal communication.

Luecht, R. M. and Nungester, R (1998). Some practical examples of computer-adaptive sequential testing. *Journal of Educational Measurement, 35* (3) 229-249.

Norris, D. G., Russell, T. L., Goodwin, G. F., & Jessee, C. L, (January, 2001). *Practice Analysis of Certified Public Accountants: Technical Report*. Jersey City, NJ: American Institute of Certified Public Accountants.

Timminga, E. (1998). Solving infeasibility problems in computerized test assembly. *Applied Psychological Measurement, 22*, 280-291.

Vos, H. J. & Glas, C. A. W. (2000). Testlet-based adaptive mastery testing, in <u>Computerized Adaptive Testing</u>, Van der Linden, W. & Glas, C.A. W (Eds.), p 289-309. Kluwer: Dordrecht.

van der Linden, W. J. (1998). Optimal assembly of psychological and educational tests. *Applied Psychological Measurement, 22*, 195-211.

van der Linden, W. J. Personal communication.

APPENDIX
An Algorithm for Multi-Target Bundling with ATA

Set-ups

1. Assumption: there is a single item bank (database) containing item statistics and attributes (e.g., content codes)

2. The fundamental unit for test assembly is called a **bundle**. Each **bundle** can represent a testlet, a fixed test form, or a combination of several testlets. Each bundle has a fixed has a set of attribute constraints to control the distribution of content and other categorical features, as well as a specific set of statistical targets (e.g., test information values at a fixed number of θ points)

    a. The attribute constraints can be different for different bundles
    b. The targets can be different for different bundles

3. Bundles can be replicated

Processing

1. The item bank file, the attribute constraints file(s) and the statistical target file (containing multiple targets) are input to the ATA software.

2. *J* bundles are indexed to constraint files and to statistical target fields (e.g., columns, with θ values as rows for test information targets)

3. Item selection tables are created for each replicate of every bundle—the total number of tables is $h=\sum r(j)$, where $r(j)$ is the number of replications of bundle j ($j=1,\ldots,J$). $S(k)$ indexes the tables, $k=1,\ldots,h$.

Algorithm

1. Select a bundle replicate at random
2. Build the necessary NWADH indexes and temporary arrays
3. Choose **one** item via the NWADH (Luecht, 1998) for that bundle table, $S(k)$
4. Select another bundle
5. Repeat step #3. Continue until one item has been selected for all bundle replicates
6. Free all bundles to be selected with equal probability
7. Go to step #1 and repeat until all bundle tables are full