Psicología Educativa 20 (2014) 79-87

Psicología Educativa

www.elsevier.es/psed

An introduction to the use of evidence-centered design in test development

Michael J. Zieky*

Educational Testing Service, Princeton, U.S.A.

ARTICLE INFORMATION

Manuscript received: 16/04/2014
Revision received: 26/08/2014
Accepted: 24/09/2014

Keywords:
Evidence centered design
Test development
Test design
Test construction
Validity
Evidentiary reasoning

ABSTRACT

The purpose of this article is to describe what Evidence-Centered Design (ECD) is and to explain why and how ECD is used in the design and development of tests. The article will be most useful for readers who have some knowledge of traditional test development practices, but who are unfamiliar with ECD. The article begins with descriptions of the major characteristics of ECD, adds a brief note on the origins of ECD, and discusses the relationship of ECD to traditional test development. Next, the article lists the important advantages of using ECD with an emphasis on the validity of the inferences made about test takers on the basis of their scores. The article explains the nature and purpose of the "layers" or stages of the ECD test design and development process: 1) domain analysis; 2) domain modeling; 3) conceptual assessment framework; 4) assessment implementation; and 5) assessment delivery. Some observations about my experience with the early application of ECD for those who plan to begin using ECD, a brief conclusion, and some recommendations for further reading end the article.

Introducción al diseño centrado en la evidencia en la construcción de tests

Palabras clave:
Diseño centrado en la evidencia
Desarrollo de tests
Diseño de tests
Construcción de tests
Validez
Razonamiento a partir de la evidencia

RESUMEN

El objetivo de este trabajo es describir qué es y explicar por qué y cómo se utiliza el Diseño Centrado en la Evidencia (DCE) para diseñar y construir tests. Este trabajo está pensado especialmente para personas que ya estén algo familiarizadas con las prácticas tradicionales de construcción de tests pero que desconozcan el DCE. Comienza con una descripción de las características fundamentales del DCE, continua con un breve apunte acerca de su origen y analiza su relación con la práctica tradicional en la construcción de tests. A continuación, se indican las ventajas que conlleva la utilización del DCE, resaltando su impacto en la validez de las inferencias realizadas sobre los sujetos en base a sus puntuaciones en los tests. En el artículo se explica la naturaleza y el objetivo de las 'capas' o etapas en el proceso de diseño y construcción de tests con el DCE: 1) análisis del dominio, 2) modelado del dominio, 3) marco conceptual de la evaluación, 4) implementación de la evaluación y 5) administración de la evaluación. Para terminar, se ofrecen algunos comentarios acerca de la experiencia del autor en la aplicación del DCE para aquellos que estén pensando en empezar a utilizarlo, junto a una breve conclusión y alguna recomendación acerca de lecturas adicionales sobre el tema.

What is Evidence Centered Design?

Major Characteristics of ECD

ECD is a logical, systematic approach to test creation. The primary goal of ECD is to base important aspects of test design, test development, test scoring, and test use on sound evidentiary reasoning. ECD treats assessment as a process of reasoning from the necessarily limited evidence of what students do in a testing situation to claims about what they know and can do in the real world. Mislevy, Steinberg, and Almond (1999) described ECD as a "principled framework for designing, producing, and delivering educational assessments" (p. 1). According to the authors, ECD "ensures that the way in which evidence is gathered and interpreted bears on the underlying knowledge and purposes the assessment is intended to address" (*ibidem*).

ECD is not a set of rigid procedures. It is, rather, a family of practices that helps test developers to clarify the inferences that are to be made about test takers on the basis of their scores, and to

*Correspondence concerning this article should be addressed to Michael J. Zieky. ETS MS 04N. 660 Rosedale Road. Princeton, NJ, USA, 08541. E-mail: mzieky@ets.org

DOI: http://dx.doi.org/10.1016/j.pse.2014.11.003
1135-755X/

determine how best to provide evidence to support those inferences within the constraints of the testing program. ECD also encourages thinking about the interrelationships among the various layers of the entire process of test design, development, and use, emphasizing not only what occurs within each layer, but also how the layers are logically related to each other. As Mislevy and Riconscente (2005) wrote, ECD is "a framework that makes explicit the structures of assessment arguments, the elements and processes through which they are instantiated, and the relationships among them" (p. iv).

A capsule view of the rationales underlying ECD was provided by Mislevy, Almond, and Lukas (2003):

> ECD is based on three premises: (1) an assessment must build around the important knowledge in the domain of interest and an understanding of how that knowledge is acquired and put to use; (2) the chain of reasoning from what participants say and do in assessments to inferences about what they know, can do, or should do next, must be based on the principles of evidentiary reasoning; (3) purpose must be the driving force behind design decisions, which reflect constraints, resources, and conditions of use (p. 20).

The use of evidentiary reasoning ties together the many uses of ECD, ranging from highly sophisticated, computerized assessments that rely on complicated statistical models to more straightforward paper-based tests that use classical measurement theory. What the variations of ECD have in common is a chain of reasoning that includes the following steps: 1) analyzing the domain of knowledge, skills or other attributes (KSAs) of interest; 2) specifying the claims to be made about the relevant attributes of test takers on the basis of the test; 3) deciding on the evidence that is required to support the claims about test takers; 4) developing the tasks that provide the desired evidence within the constraints of the testing program; 5) assembling the tasks into test forms that support all of the stated claims with sufficient evidence to justify use of the test scores; 6) providing scoring rules for tasks, and rules for aggregating scores across tasks, that extract the evidence required to support the claims; and 7) describing explicit logical links among all of the previous steps.

Origins of ECD

Russell Almond, Robert Mislevy, and Linda Steinberg were the primary researchers who developed ECD at Educational Testing Service in the last decade of the 20th century. Mislevy, Almond et al. (2003) credited Messick's views on validity for "the conceptual groundwork" that helped to form ECD. Messick (1989) famously defined validity as, "the degree to which empirical evidence and theoretical rationales support the *adequacy* and *appropriateness* of *inferences* and *actions* based on test scores" (p. 13). The focus on evidence used to support inferences about test takers became the core of ECD.

Mislevy (1994) cited Stephen Toulmin as the creator of the structure of evidentiary arguments that Mislevy used as the basis for ECD. According to Toulmin (1958), every claim is a proposition that has to be based on data, and there has to be a "warrant" that supports the logical connection between the data and the claim. Claims and warrants are very important aspects of ECD as discussed below.

Relationship of ECD to Traditional Test Development

Mislevy, Steinberg et al. (1999) wrote that ECD "is not so much a particular advance in statistics, psychology, or forms of assessment [It is, rather] a coherent framework to harness recent developments of these various types toward a common purpose" (p. 1). It is important to understand that ECD, in operation, becomes a means of formalizing, documenting, and extending the best practices of traditional test development, not a means of supplanting them. For a brief overview of test development, see pages 75-84 of the *Standards for Educational and Psychological Testing* (American Educational Research Association, American Psychological Association, & National Council on Measurement in Education, 2014).

Good test developers have always strived to define the purpose of a test as completely as possible, to decide the best way to meet the purpose that has been established for a test, and to do so within the constraints that have been imposed by the testing program and the client. Traditional test developers determine the KSAs to be measured to meet the purpose of a test and choose the best ways to measure those KSAs within the existing schedule, budget, and other constraints. They generate detailed test specifications, create tasks to measure the selected KSAs at the appropriate levels of difficulty, provide keys or scoring rules for the tasks, assemble tests to meet specifications or write rules that govern computerized assemblies, and describe how to combine item-level observations to generate meaningful scores.

ECD does not make any of those test development tasks obsolete, nor does ECD offer entirely novel ways to perform those tasks. ECD is not used in place of traditional test development. ECD is used to enhance traditional test development. As Mislevy and Haertel (2006) noted, "each innovation is grounded in the same principles of evidentiary reasoning that underlie the best assessments of the past" (p. 1).

What Are the Advantages of Using ECD?

Though ECD has many advantages in other situations, it is least useful in the maintenance of established, ongoing testing programs in which the primary work is writing new tasks that are similar to the existing tasks and assembling new forms of the test that are parallel to existing forms. Many of the most useful aspects of ECD become irrelevant because the decisions they are designed to facilitate have already been made for the initial forms of the test. As long as those initial forms are simply being replicated as closely as possible in parallel forms, the machinery of ECD will bring few improvements.

ECD becomes more helpful for redesigning tests. The fewer constraints there are on the changes that can be made, the more helpful ECD becomes. ECD is even more useful for making new tests of previously measured constructs, and is most useful for measuring new constructs. In fact, the less experience test developers have measuring some domain, the more useful ECD becomes because it helps to ensure that test developers will seek the information they require about the domain to be tested, will clearly specify the claims to be made about test takers, will determine the evidence required to back the claims, will develop tasks that provide the desired evidence, and will score them appropriately.

A primary advantage of ECD is that it helps to build in validity during the test design and development process. The *Standards for Educational and Psychological Testing* (American Educational Research Association, American Psychological Association, & National Council on Measurement in Education, 2014) calls for a "validity argument" supporting the appropriateness of the inferences to be made on the basis of the assessment results. ECD provides a strong foundation for the validity argument by requiring documented, explicit linkages among the purpose for a test, the claims made about test takers, the evidence supporting those claims and the test takers' responses to tasks that provide the evidence. ECD helps ensure that tasks are measuring construct-relevant KSAs and makes it easier to determine if tasks are inadvertently measuring construct-irrelevant KSAs. Note that even if ECD is used, the gathering of validation evidence based on such factors as expert judgments of task content, the empirical relationships among parts of the test, and the empirical relationships of test scores with external variables is still required.

The use of ECD helps to ensure that important decisions about the test are documented. Furthermore, the documentation requirements of ECD help increase the clarity of communications among the many different people who work together to implement a complicated testing program. Almond (personal communication, August, 2014) notes that, because ECD requires all participants to use a common language rather than the jargon of their own disciplines, the participants should be able to communicate more clearly across disciplines in an ECD context than in a traditional context.

The documentation and model-building aspects of ECD are likely to be more time-consuming than traditional test development practices during the early stages of test creation. The reward, however, comes in the ability to document more clearly the evidence of validity, in the ability to develop tasks more likely to be parallel to existing tasks, and in the ability to construct additional forms of the test more likely to be parallel to the initial form.

What Are the Layers in ECD?

ECD divides the entire process of designing, developing, and using tests into five groups of activities called "layers": 1) Domain Analysis, 2) Domain Modeling, 3) Conceptual Assessment Framework, 4) Assessment Implementation, and 5) Assessment Delivery. According to Mislevy and Riconscente (2005, p. 3), "The compelling rationale for thinking in terms of layers is that within complex processes it is often possible to identify subsystems, whose individual components are better handled at the subsystem level." They continue, "Each layer clarifies relationships within conceptual, structural, or operational levels that need to be informed by, or hold implications for, other levels" (p. 5). An additional advantage of dividing the process into layers is that it leads to greater efficiency. It is less costly to make changes in the early layers (Domain Analysis and Modeling), which are designed to help test developers think through important issues, than it is to make changes after incurring the expense of item acquisition.

Though the discussion of the separate layers may give the impression that they are distinct, their borders are porous. In the real world there are always constraints that limit what can be tested and how it can be tested. Sometimes the effects of the constraints can be anticipated and accounted for in the test design. For example, a client may specify that the test has to be administered in an hour or less. The time constraint will limit the number of tasks that can be administered, which will limit the amount of evidence that can be gathered, which, in turn, will limit the number of claims that can be made.

Often, however, the effects of the constraints are not clear at the beginning of the process. In actual implementation of ECD there is much movement back and forth across the layers. Operational work often proceeds on several layers at the same time. Therefore, arguments about which layer should contain a particular aspect of test development are pointless. The ECD process is iterative. Problems encountered in a later layer may force a return to an earlier layer. For example, repeated failed attempts to develop tasks to obtain the evidence needed to support certain claims may make it clear that the evidence is not obtainable within the constraints that have been established by the testing program. If the constraints are inflexible, the claims will have to be revised.

Domain Analysis Layer

Every test is a sample from some domain of KSAs. For example, the important KSAs required of a beginning teacher constitute a domain from which the KSAs measured in a teacher-licensing test are sampled. ECD requires identification of the relevant domain and an investigation of its characteristics. What KSAs are most important? How are they represented? How are the KSAs related to one another? How are the KSAs generally acquired and how are they used in the real world? For example, KSAs typically learned on the job are not appropriate in a licensing test. What kind of work dependent on the KSAs is valued? How is good work distinguished from mediocre or poor work?

In many test development projects, the test developers depend on subject-matter experts to help with domain analysis. In fact, the same sources of information are used in traditional test development and in ECD: committees of subject-matter experts, curriculum analyses, task analyses, surveys of teachers of the tested subject, surveys of job incumbents, states' and professional organizations' content standards, popular textbooks, and the like. The test developers themselves are not expected to become expert in every domain in which they work. They are, however, expected to know how to elicit the necessary information to complete the domain analysis.

Domain Modeling Layer

The domain modeling layer moves from an investigation of the relevant real-world domain to a use of selected aspects of the domain for the purpose of building an assessment argument. The general form of the assessment argument is "If (X), then (Y) because (Z)". X is an observation of test taker behavior or a product of that behavior. Relevant aspects of the behavior or product form the data on which the claim is based. Y is a claim that the test taker has or lacks some KSA or related cluster of KSAs, and Z is the warrant that explains why the behavior or product demonstrates the possession (or lack of) the KSA(s). For example, an aspiring firefighter might successfully complete a task (X) that requires dragging a 200 pound dummy 45 feet through a smoke-filled hallway within a specified time limit. The ability to complete the task is a partial demonstration that the test taker has the strength and speed required to be a firefighter (Y), because the task replicates important physical aspects of the firefighter's job under realistic conditions (Z). The parts of the assessment argument (claims, data, and warrants) are discussed below.

Claims. Claims are the statements that test users want to be able to make about test takers on the basis of their performances on the test. Claims are a way to communicate what test scores mean. Claims may be very general (e.g., test taker can read at the first grade level) or be more specific (e.g., test taker can decode initial consonants). A single test may be the basis for many claims at different levels of specificity. The use of ECD can be thought of as a means of building a chain of arguments and evidence to support the claims that are made about test takers.

Whether general or specific, claims must be clear. One way to assess the clarity of a claim is to determine if it is possible to imagine the evidence that theoretically would be sufficient to demonstrate clearly whether or not a test taker had met the claim. If such a demonstration is impossible, even under theoretically ideal conditions, then the claim needs to be made more precise.

The purpose of a test and the claims to be made about test takers are very closely related. In fact, the purpose of a test can easily be restated as the highest level claim that can be made on the basis of the test results. For example, if the purpose of a test is to determine whether a person can drive competently enough to obtain a driver's license, the highest level claim is that a person who passes the test can drive competently enough to obtain a driver's license.

The high level claim must be supported by lower level claims at increasing levels of specificity. For example, the high level claim about sufficiently competent driving is likely to be supported by a claim about visual acuity, a claim about knowledge of the rules of the road, a claim about knowledge of penalties for infractions, a claim about the ability to operate a car, and so forth. Each of those claims

generates questions that will, in turn, lead to lower level claims. Knowledge of which rules of the road is necessary? Does visual acuity include color vision? What exactly does the person have to be able to do to operate the car competently? For example, is parallel parking required? If so, how much space should be allowed?

Claims made about test takers will vary depending on the type of test being developed. For tests with a pass-fail score, claims will generally begin with a format similar to, "Test takers who pass are able to…. " For tests with proficiency labels such as "basic," "proficient," and "advanced," claims will generally be made for test takers at each proficiency level using a format similar to, " Test takers at the basic level are able to… " For norm-referenced tests, and for tests used predictively, different claims are made about test takers at high, medium, and low score levels.

Beyond those generalities and the fact that all claims should concern attributes of test takers that score users care about, there is no fixed formula or single required format for writing claims. The claims should answer the question, "What do test users want to say about the test taker on the basis of responses to the test?"

Prospective score reports. A useful tool to help define claims is the Prospective Score Report (PSR). As the name implies, the PSR is simply an early mock-up of the final score report or set of score reports for different users. Developing the score report so early in the test development process may seem counterintuitive, but the score report can be thought of as the end product of the entire testing operation. Having the end product in mind as work begins helps test developers ensure that their ensuing work will be sufficient to produce the desired product.

The first step in producing the PSR is identifying the various types of score users who will receive score reports. For a driver's license test, for example, one score user is clearly the state motor vehicle agency and another is the aspiring driver. The next task is to decide what information to include in the score reports for the test to meet its purpose for each intended group of score users. For example, the state motor vehicle agency would certainly want a pass/fail score for each test taker. The aspiring drivers who pass may desire no other information. Those who fail, however, would want to know why they failed so they could strive to improve the knowledge and skills shown to be insufficient, thereby increasing their chances of passing a retest. The cause(s) of the failure, as reported in subscores, would require that additional claims be made, which would, in turn, require that additional evidence be provided by the test. ECD makes very clear that increasing the number of reported scores requires enhancing the test to provide the evidence necessary to support the additional scores. Designing the score reports for various score users early in the test development process is a good strategy to help clarify the claims to be made about test takers and to help specify the information that must be provided by the test to support the claims.

An additional use of the PSR is to convey to clients the effects of certain design decisions in terms they will understand. For example, a mock-up of the score report that might result from three 20-minute essay responses compared to the score report that might result from 80 multiple-choice items is a convincing way to display some of the advantages and disadvantages of the different types of items.

Data. In the domain modeling layer, the observable data on which the claims will be based are described in general terms. Mislevy (2006) described the three types of data commonly used to support claims in ECD. The first type of data includes aspects of the situation in which the person is acting. In the context of assessment, the situation is shaped by the contents of the task (test item) to which the person is responding, the stimulus materials available, and the response mechanism in operation. The second type of data includes aspects of the person's actions in the situation. These data are derived from the test taker's response to the task. The data collected can be as simple as a mark on an answer sheet or as complicated as a complete recording of everything the test taker said and did in an extended simulation exercise. The third type of data includes additional information about the person's history or relationship to the observational situation. What is known about the test taker that may affect the interpretation of a response to the task? For example, whether or not the test taker were an English language learner would affect the claims that could be made based on mathematics tasks with a high reading load.

The problem to be addressed is that tests cannot *directly* measure most KSAs of interest. All that can be measured directly is an observable behavior or product. From that observation, inferences are made about the KSAs of the test taker and the claims that can be made about the test taker. Sometimes, the behavior itself must be evaluated. For example, to determine if a test taker can draw blood correctly for laboratory analysis, it is necessary to observe the test taker's behavior as the blood is drawn. Did the test taker use sterile technique, maintain the client's comfort, and so forth? Just looking at the product resulting from the behavior, the filled test tube, is insufficient.

Often, however, it is possible to evaluate the product of behavior rather than the behavior itself, which is much more efficient. For example, it is not necessary to watch a test taker paint a picture to evaluate the finished painting. Therefore, evaluation of behavior is generally limited to situations in which important information would be lost by a focus on a product.

Much of what it is important to test, however, is not directly observable at all. For example, whether or not a test taker understands a reading passage is rarely discernible from watching the person read. There are usually no outward manifestations that a student in a statistics class understands the difference between the variance and the standard deviation. The job of the test developer is to decide what observable data would allow inferences about the unobservable KSAs. (In a later layer, the job of the test developer is to devise tasks that will elicit the required observable behaviors.)

Some test developers have found it useful to imagine the ideal setting in which to gather data to support the claims to be made. Once the ideal observation is established, the test developers determine which parts of that observation are impossible within the real-world constraints of the testing program. What has to be given up? What substitutes can be made? How closely can the ideal observation be approximated in the test?

Warrants. The "warrants" logically connect the observed data to the claims. The need for elaborate warrants varies with the strength of the link between the data and the claims to be made. Sometimes, the link between the data and the claim is so clear that the warrant becomes self-evident and requires little explanation. For example, it is very straightforward to explain the logical link between a road test that requires a test taker to demonstrate the ability to perform typical driving tasks safely, and the claim that a test taker can operate a motor vehicle without endangering the public.

At the opposite extreme, consider an IQ test. The highest level claim is about the test taker's level of "intelligence". The tested behaviors, however, include recalling strings of numbers, completing puzzles, defining words, etc. A strong warrant is needed to explain why the ability to recall strings of random numbers allows inferences about a test taker's intelligence. When there is a great difference between the observed behavior and the claim that is to be made, the warrant must be comprehensive and convincing.

Accompanying many warrants are potential "alternative explanations" that must be examined and excluded to help ensure that the logical link between the behavior and the claim described in the warrant is correct. For example, excessively hard language in a task may cause English language learners who possess the tested KSAs to respond incorrectly to the task. An important part of the test developer's job is to reduce the likelihood that alternative explanations for the warrants are correct.

Conceptual Assessment Framework Layer

The conceptual assessment framework layer contains many of the "tools" of ECD used by test developers such as the student model, the evidence model, the task model, and the assembly model. Each will be described below, but the frequent use of the word "model" in ECD may require explanation. A model is a term used to refer to a simplified, understandable, usable representation of a far more complex reality. For example, a road map could be called a "terrain model." It is highly useful because it is greatly simplified compared to reality. It contains the information necessary to travel from one location to another by car. It is easy to find and use the desired information because the map does not include the countless details of the actual terrain that are irrelevant for travel by road. A model allows focus on the important aspects of a segment of reality for a particular purpose. A model works to the extent that it captures the components of reality that are relevant to the purpose for using the model and omits the irrelevant components.

Student model. The student model (also called the proficiency model and the competency model) is sometimes used as a simplified representation of a test taker, showing the relevant KSAs that an individual test taker might have. It contains the KSAs that are the focus of measurement and the other characteristics of the test taker that would affect the interpretation of test performance. After the test taker responds to a task, the model can be updated indicating the current estimate of the likelihood that the test taker has (or lacks) the measured KSA(s). Sometimes, the student model (better named the proficiency or competency model in this case) shows the relevant KSAs that the population of test takers might have. In addition to the KSAs and other attributes directly related to the test, the student model may include a description of the intended test takers in terms of what they generally know and are able to do, such as familiarity with computers, need for accommodations, and so forth. In sum, the student model includes the information about test takers necessary to allow test developers to write tasks that are appropriate for the intended population.

The student model is likely to contain more KSAs than are to be included in the score report. For many tests there may be only a single score summed over all of the measured KSAs. Even in tests that report several subscores, there are likely to be more KSAs measured in the test than are individually reflected in the score report. To help with later work on the assembly model, test developers find it useful to differentiate among the KSAs to be reported separately and the KSAs that have been included in the student model to support the reported information.

Evidence model. Evidence models are based on the observable behaviors or observable products of behavior resulting from responses to a particular task. The job of the test developer in creating the evidence model is to describe in detail the aspects of the observable behaviors or observable products that would provide evidence that test takers have the KSAs that are the focus of measurement in a task. An observable behavior or product may provide evidence about several proficiencies. For example, the observed behavior of stopping a car smoothly at a stop sign provides evidence concerning knowledge of the meaning of a stop sign and the ability to apply the brakes appropriately.

As is the case for claims, there is no definitive formula or required format for writing evidence models. The goal is to answer the question, "What would a test taker have to do – or what could a test taker show us – in response to this task to allow us to make the desired claim?"

Other issues that test developers consider when constructing evidence models are: 1) the aspects of the behavior or product that affect the score; 2) the important differences between a good/correct behavior or product and a bad/incorrect behavior or product; 3) the ease or difficulty of observing the important differences; 4) the aspects of the behavior or product that would be most relevant or that would be irrelevant; and 5) the general scoring rules or rubrics for constructed response and performance tasks.

Task model. A task is simply something specific that test developers ask a test taker to do that will be scored, such as select an option in a multiple choice item, write an essay, or take the required action in response to a performance item. A task model is a description of the characteristics that define a group of tasks. The task model should, of course, be linked to the aspect(s) of the evidence model for which it will provide information.

Task models can help test developers design or select appropriate types of tasks to use for a test. The task model requires the test developer to describe the desired attributes of the tasks to be generated. The task model helps test developers determine the various item types that can display the desired attributes before the test developers commit to a particular item type.

The task model describes a family of tasks. It defines a group of situations that would elicit the desired observable behavior or observable product. A task model generally includes 1) a description of the KSAs that the tasks measure; 2) the types of stimulus materials that might be used; 3) a description of what the test taker will be asked to do; 4) descriptions of required task elements and allowable variable task elements; 5) the attributes that affect the difficulties of the tasks that will eventually be produced; and 6) several samples of tasks that could be generated by the model. The sample tasks are very important in helping test developers understand the model.

Assembly model. The assembly model describes what the test as a whole will look like. Assembly models are expanded versions of detailed test specifications. An assembly model contains the information necessary to build parallel forms of the test. (The algorithms used to generate computer assemblies of tests are part of the assembly model for such tests.) Desired statistical attributes of the test as a whole should be indicated in the assembly model, including such data as the target mean difficulty and mean discrimination, the distribution of task difficulty, the desired reliability of the reported score(s), and the intended level of speededness. Ideally, the assembly models should be specific enough that the test forms generated by the same model are interchangeable.

To make the linkages among claims, evidence, and tasks explicit, the assembly models indicate the KSA(s) and claim(s) for which each task provides evidence. Different assembly models can be used with a single pool of tasks to produce a "family" of tests. For example, different assembly models could be used to produce diagnostic tests that serve as study guides and summative tests for decision-making purposes about test takers. (Note that different evidence models would likely be needed as well.)

Assessment Implementation Layer

The assessment implementation layer is closely related to traditional test development jobs of writing items and assembling test forms. One of the tools of ECD used in the assessment implementation layer is the task shell.

Task shells. The task models described above generate a family of tasks, but the tasks that fit the model are not necessarily parallel to each other. A task shell, however, is a way to generate potentially parallel tasks. Task shells use a framework with variable elements and descriptions or lists of what can serve as the variable elements. In the following very simple example, the variable elements are in brackets:

> What is the [mean, median, mode] of the following distribution: [25-30 two-digit numbers in random order]? (non-statistical calculators allowed).

By plugging in different values for the variable elements, multiple tasks can be generated from the shell. If the variable elements are consistent in important characteristics including those that affect difficulty, the tasks might possibly be reasonably parallel. Experience has shown, however, that small variations in tasks generated by the same shell can lead to large differences in difficulty.

In addition to the framework and the variable elements, task shells may contain 1) a statement of the KSAs to be measured by the task; 2) the directions that apply to the tasks generated by the shell; 3) specifications for any stimulus material to be provided with the task; 4) for multiple choice tasks, rules for generating distracters; and 5) general scoring rubrics for constructed response tasks, to be augmented by prompt-specific scoring rules as necessary. Although the logical flow appears to be to create a task shell and use it to create tasks, ECD does not require strict adherence to a sequence of steps. Some test developers have found it very useful to work "backwards" from existing exemplary tasks to create task shells, and many successful tasks are created in the absence of task shells.

If task shells are successful at identifying and holding constant the elements in a task that affect its difficulty and discrimination, it may be possible to reduce pretesting requirements because pretesting a sample of tasks generated by the shell would provide data that could be applied to all of the tasks generated by the shell. It is an empirical question whether or not the pretested sample of tasks generated by the shell will be similar enough in their operating characteristics to reduce the need to pretest all of the tasks generated by the shell. If the variable elements can be sufficiently specified, task shells can facilitate the automated generation of tasks.

Assessment Delivery Layer

As the name implies, in the assessment delivery layer the test is administered and scored. ECD views assessment delivery as generally consisting of four processes, referred to as the four-process architecture: 1) Activity Selection, 2) Presentation, 3) Response Processing, and 4) Summary Scoring (Almond, Steinberg, & Mislevy, 2002).

Activity selection. In the activity selection process, tasks are selected to be presented to the test taker. The selection proceeds according to rules to ensure that the assembly model (see above) is implemented. In traditional linear testing, all of the selection activities can take place before the test is administered. In linear "on-the-fly" testing, tasks are selected as the test taker is responding, but earlier responses to tasks have no effect on later selection of tasks. In adaptive testing, the responses to earlier tasks affect the selection of later tasks. In one common system of adaptive testing, correct responses to earlier tasks lead to the presentation of more difficult tasks; incorrect responses to earlier tasks lead to the presentation of less difficult tasks. The goal is to select the most appropriate items for the ability level of the individual test taker while also implementing the content and skills portion of the assembly model, as the test is being administered. (See Stocking & Swanson, 1993, for an example of an adaptive algorithm that meets content and skills constraints.)

Presentation. Two events take place in this process. The selected task is presented to the test taker and the test taker's response to the task is recorded. In traditional large-scale testing, the test taker reads the task in a printed test book and records a response on an answer sheet. Increasingly, the presentation mode is a computer monitor and the response mode is via mouse or keyboard. In performance tests, the response is some physical activity and may last for an extended period, such as making a sculpture in a visual arts test.

Response processing. ECD does not require any particular form of scoring responses to tasks. What it does require is that scoring be based on a chain of explicit logical connections. In fact, scoring can be thought of as following the same chain of reasoning that guided test construction, but in the opposite direction. That is, test construction moves from claims, to evidence models, to tasks. Task scoring models move from the task level back up through the evidence model to result in judgments about the test taker's status with respect to the KSA's tested by the task. (For information about the role of ECD in automated scoring, see e.g., Bejar, 2011; Scalise & Wilson, 2006.)

Summary scoring. Judgments about the test taker's status with respect to a claim are rarely made on the basis of a single task, however. Therefore, the last of the four processes accumulates scores across the presented tasks. Summary scoring requires the application of a quantitative method of some sort. Scoring systems vary widely in complexity. For example, a test that targets a single ability, with a single score, and an evidence model in which each task connects directly to this single ability could appropriately be scored by simply counting the number of right answers and placing the number on a meaningful scale. On the other hand, a diagnostic assessment that targets multiple abilities, with many subscores, and an evidence model in which there are multiple connections among tasks and different KSAs would more appropriately be scored by a more sophisticated model such as Cognitive Diagnostic Models or Bayes nets (see, e.g., de la Torre & Minchen, this issue; Mislevy, Almond, Yan, & Steinberg, 1999).

Regardless of its complexity, a summary scoring method results in one or more categories (e.g., Basic, Proficient, Advanced; Pass, Fail), or in one or more numbers on a score scale. The appropriate scoring model depends on the kinds of claims that are to be made and on the evidence models and tasks used to support the claims. Therefore, test developers cooperate with measurement statisticians in selecting the appropriate scoring model(s) for use in a test. ECD facilitates that cooperation by making clear the chain of evidence to be maintained by the scoring system and by providing a common frame of reference for the test developers and measurement statisticians.

Some Observations on the Introduction of ECD

I was working in test development at Educational Testing Service (ETS) when ECD was first introduced. In the 20 or so years since then, I have seen ECD evolve from a small research project to a widely used process for test design and development. ECD is now routinely used, but the early applications of ECD at ETS shared certain problems which I believe will be instructive for new users of ECD.

At first, experienced test developers did not intuitively grasp ECD, nor did they immediately see the advantages of its use. There was skepticism, and some resentment at being asked to deal with esoteric vocabulary and to change traditional practices that had served the developers well in the past. (Calling ECD "Principled Assessment Design," which implied that their traditional practices were unprincipled, did not endear ECD to test developers.) There were many complaints about being forced to do unnecessary work and to "waste time" documenting things that "everybody knows". There were arguments about what exactly test developers had to do to justify a claim that they were using ECD. As was the case with test developers, the external committees of subject-matter experts, often used as contributors to the test development process, tended not to be impressed by the introduction of ECD. They saw the machinery of ECD as "overkill" and the vocabulary of ECD as a burden. The introductory problems were compounded because the additional burdens of using ECD were immediate while the advantages took time to accrue.

It took several years of experience for the test developers to become comfortable with ECD. Each use of ECD provided experience with the important tools of ECD such as claim statements, evidence models, task models, prospective score reports, and so forth, which made the next use easier to undertake. It became clear that the

people who designed tests and the people who developed tasks had to work together rather than sequentially to improve communication and avoid needless false starts and excessive rework.

Based on my experience with the introduction of ECD, I suggest that people who plan to initiate the use of ECD with test developers who have not previously used it consider the following steps. 1) Do not promise more benefits than ECD can deliver. Item writing will not become effortless, nor will pretesting become unnecessary. 2) Be clear that more work is required at the beginning stages of test development than people are used to doing and that increases in efficiency will not immediately be apparent. 3) Point out the many ways in which ECD is similar to traditional test development and provide realistic rationales for the differences. 4) Introduce the vocabulary of ECD as it is needed rather than all at once. Provide rationales for the use of the new terms. 5) Do not assume that reading articles about ECD is sufficient. Hands-on training is required. If possible, the instructor should be an experienced test developer who has credibility with the participants. 6) Provide examples of the tools of ECD that are clearly relevant to the work that will be done. 7) Build extra time in the schedule. Expect and encourage iteration. Not everything that is attempted will work the first time. 8) As work proceeds, try to ensure cooperation and communication across the layers of the process. 9) If possible, go through the entire ECD process with one test rather than attempting to introduce ECD in many tests simultaneously. 10) After the completion of each major segment of the work, hold an "after action review" to determine what worked and what caused problems. Discuss how to avoid the problems in the next iteration.

Conclusion

Validity has always been the driving criterion for good test design and development, whether using ECD or not. ECD, however, makes the factors that influence test design explicit and links the myriad decisions made during task creation, test assembly, and scoring into a chain of evidence-based reasoning that better supports an argument for the validity of the inferences made about test takers on the basis of their scores.

Nothing is done in ECD that is at all contrary to good traditional test development practice. Using ECD, however, aspects of the process are specified in far greater detail than is usual in traditional test development. If it is used wisely, ECD can help test developers accomplish work more efficiently. On the other hand, slavish adherence to aspects of ECD that are not necessary in some particular set of circumstances can waste test developers' time. Flexibility and common sense are as important in the practical application of ECD as they are in traditional test development. As readers of this article will have noticed, much vocabulary is employed that may be comfortable for cognitive scientists but is still novel and somewhat opaque for most test developers. In response to the criticism that ECD was just "a bunch of new words for things we are already doing," Mislevy and Haertel (2006, p. 23) wrote

> Evidence-centered design is a framework, then, that does indeed provide new words for things we are already doing. That said, it helps us to understand what we are doing at a more fundamental level. And it sets the stage for doing what we now do more efficiently and learning more quickly how to assess in ways that we do not do now.

Suggested Reading

For more information about ECD, Robert Mislevy (personal communication, 2014) recommended Mislevy, Almond et al. (2003) as the best place to start for someone new to ECD. He recommended the following for general readers and test developers: Mislevy and Riconscente (2005), Mislevy and Haertel (2006), Mislevy, Haertel, Yarnall, and Wentland (2011), and Mislevy, Bejar, Bennett, Haertel, and Winters (2010). For more technical details, Mislevy recommended Almond, Steinberg, and Mislevy (2002), and Mislevy, Steinberg, and Almond (2003). For a number of downloadable papers dealing with ECD and related topics see http://www.education.umd.edu/EDMS/mislevy/papers/

Information about some applications of ECD will illustrate its versatility and utility. The work done by the two consortia (Smarter Balanced Assessment Corporation and the Partnership for Assessment of Readiness for College and Careers) on the Common Core State Standards (see Pellegrino, this issue, for a discussion of the CCSS) offers a chance to check online how some of the concepts in this paper are translated into a program. For information about the uses of ECD in the development of a cognitively-based assessment designed to improve as well as assess school-based learning, see the papers in this volume by Deane and Song, and by van Rijn, Graf, and Deane, as well as Bennett (2010), and Graf (2009). For an application of ECD to large-scale assessment see Huff (2010). For the use of ECD in an academic admissions test, see Sheehan, Kostin, and Futagi (2007). For the use of ECD in the domain analysis for a licensing test, see Tannenbaum, Robustelli, and Baron (2008). For information about how ECD was used in reasoning about accommodations for people with disabilities, see Hansen, Mislevy, and Steinberg (2008). The use of ECD in a test of English for non-native speakers is described in Hines (2010). For more about the use of ECD in language testing, see Mislevy and Yin (2012). For information about the role of ECD in automated item generation, see Huff, Alves, Pellegrino, and Kaliski (2013). For the use of ECD with assessments embedded in simulations and games, see Almond, Kim, and Shute (2014). Finally, for a discussion of the role of ECD in 21st century teaching and learning, see Pellegrino, this issue.

Resumen ampliado [1]

El Diseño Centrado en la Evidencia (DCE) concibe la evaluación como un proceso de razonamiento que parte de la información –necesariamente limitada– acerca de lo que hacen los estudiantes en la situación de evaluación para llegar a afirmaciones acerca de lo que saben y pueden hacer en el mundo real. El DCE es un conjunto de prácticas que sirven para clarificar las inferencias que se pretende hacer acerca de los sujetos en base a sus puntuaciones en los tests y para determinar cómo proporcionar la mejor evidencia posible para poder realizar con garantías dichas inferencias, dentro del marco de las condiciones particulares de cada programa de evaluación. De algún modo, el DCE constituye una manera de formalizar, documentar y ampliar las mejores prácticas de la construcción tradicional de pruebas o tests y contribuye de manera decisiva al argumento de validez del test, al requerir un vínculo explícito y documentado entre el objetivo del test, las afirmaciones que se desea realizar sobre los examinados, la evidencia que hace posible tales afirmaciones y las respuestas de los sujetos a las tareas que proporcionan dicha evidencia.

El DCE concibe el proceso global de diseñar, construir y utilizar tests como un proceso iterativo organizado en cinco grupos de actividades denominadas 'capas', con límites más bien porosos: 1) análisis del dominio, 2) modelado del dominio, 3) marco conceptual de la evaluación, 4) implementación de la evaluación y 5) administración de la evaluación.

Un test es una muestra de algún dominio de conocimiento, destreza o habilidad y el DCE requiere que se identifique el dominio de interés así como una investigación acerca de sus características, en la que junto al constructor del test habrá que contar con comités integrados por expertos en distintas cuestiones (e.g., el campo en cuestión, especialistas en curriculum, profesores del ramo, psicómetras) y por los grupos de interés. No se trata de que el constructor del test

sea un experto en cada área en la que trabaje sino de que sea capaz de obtener la información necesaria para realizar el análisis del dominio.

A partir de esa investigación, en la fase de modelado del dominio se seleccionan determinados aspectos del mismo para construir el argumento de evaluación del test, especificando de manera muy detallada y pormenorizada sus tres elementos: la(s) afirmación(es) que se desea realizar acerca de los sujetos, los datos observables en los que éstas se basan y las garantías necesarias para conectar unas y otros. Se trata básicamente de contestar a la pregunta '¿qué quieren decir las personas que van a utilizar el test en cuestión acerca de las personas que lo han respondido?' y de obtener los datos y la justificación necesaria para poder afirmar justamente lo que se quiere decir sobre ellos. Para ello, resulta muy útil elaborar de antemano los informes con los resultados del test para las distintas audiencias implicadas: esto ayuda a clarificar las afirmaciones que se desea realizar sobre los examinados y a especificar qué información se necesita para justificar que el test permite concluir eso.

El marco conceptual de la evaluación contiene cuatro importantes herramientas del DCE: los modelos de estudiante, evidencia, tarea y ensamblaje. El modelo de estudiante es una representación simplificada del sujeto examinado que muestra los conocimientos/destrezas/habilidades que constituyen el foco de la evaluación, junto a otras características que puede tener cada persona y que podrían influir en su actuación en el test. El modelo de evidencia sirve para responder a la pregunta '¿qué tendría que hacer un examinado –o qué debería mostrarnos– en respuesta a esta tarea para poder realizar la afirmación deseada?': el modelo ha de proporcionar evidencia de que los examinados tienen esos conocimientos, destrezas o habilidades que son objeto de medición. El modelo de tarea describe familias de tareas, esto es, situaciones que elicitan las conductas (o productos) observables que se desea generar, describiendo las características de dichas tareas así como el formato de las preguntas que pueden ayudar a ponerlas de manifiesto e incluyendo varios ejemplos. El modelo de ensamblaje describe cómo será el test y contiene la información necesaria para construir formas paralelas del mismo, incluyendo información sobre sus características psicométricas y sobre qué conocimientos/destrezas/habilidades y afirmaciones proporciona evidencia cada tarea.

En la etapa de implementación de la evaluación, las principales actividades son la redacción de preguntas o ítems y su ensamblaje en las distintas formas del test. Una herramienta útil es la caja de tareas, que trabaja sobre una estructura fija con elementos variables (para los que se proporcionan listas de términos posibles) con el fin de generar preguntas potencialmente paralelas.

En la última etapa del proceso se realizan las siguientes actividades, conocidas como arquitectura de cuatro procesos: (1) selección de las tareas con arreglo al modelo de ensamblaje formulado en la tercera etapa, (2) administración de las preguntas seleccionadas al examinado y registro de sus respuestas, (3) procesamiento de las respuestas con el fin de asignar una puntuación a cada una de las preguntas realizadas y (4) en base a las puntuaciones obtenidas en dichas preguntas, estimación del nivel del examinado en el dominio evaluado para poder realizar las afirmaciones o inferencias previstas. Para ello, se necesita un modelo estadístico que relacione las puntuaciones en las preguntas con el conocimiento/destreza/habilidad evaluado con la prueba (e.g., basado en la teoría clásica, en la teoría de respuesta al ítem, en un modelo de diagnóstico cognitivo), teniendo en cuenta el tipo de afirmaciones que se desea hacer sobre los examinados y el modelo de evidencia definido en la tercera etapa.

El trabajo termina con unas sabrosas reflexiones fruto de la dilatada experiencia del autor, donde se hacen recomendaciones prácticas de indudable interés, se señalan las dificultades habituales que se encuentran al trabajar con este diseño los constructores de tests acostumbrados al tradicional *modus operandi* y se destaca también que el DCE facilita la comunicación entre los distintos grupos de profesionales implicados en el proceso y contribuye a evitar vueltas atrás y la repetición de trabajo de manera innecesaria.

En suma, pese al trabajo extra que supone la construcción de un test cuando se opera con este diseño, su utilización contribuye decididamente a construir el argumento de validez de las inferencias que se realizan acerca de los examinados a partir de sus puntuaciones en el test: el DCE constituye un medio para construir una cadena de argumentos y evidencia en apoyo de las afirmaciones que se desea hacer acerca de los sujetos que responden al test. Este trabajo *encadenado* implica: (1) analizar el dominio de interés, (2) especificar las afirmaciones que se desea realizar acerca de características relevantes de los sujetos, (3) decidir qué evidencia se necesita para poder realizar dichas inferencias, (4) desarrollar las tareas que proporcionarán la evidencia deseada teniendo en cuenta las limitaciones propias de cada programa de evaluación, (5) ensamblar las tareas en las distintas formas del test, (6) puntuar las respuestas a dichas tareas y combinar esas puntuaciones para obtener la evidencia requerida para poder realizar las inferencias previstas y (7) describir de manera explícita y lógica los vínculos que existen entre todos los pasos anteriores.

Conflict of Interest

The author of this article declares no conflict of interest.

Acknowledgements

I wish to acknowledge the very helpful reviews of Russell Almond, Maria José Navas Ara, James Carlson, Marisa Farnum, Maurice Hauck, and Richard Tannenbaum.

Notes

[1]Este resumen ha sido realizado por la editora del número, María José Navas.

References

Almond, R. G., Steinberg, L. S., & Mislevy, R. J. (2002). Enhancing the design and delivery of assessment systems: A four-process architecture. *Journal of Technology, Learning, and Assessment*, *1*(5). Available from http://www.jtla.org.

American Educational Research Association, American Psychological Association, & National Council on Measurement in Education (2014). *Standards for educational and psychological testing.* Washington, DC: American Educational Research Association.

Bejar, I. (2011). A validity-based approach to quality control and assurance of automated scoring. *Assessment in Education: Principles, Policy & Practice, 18*, 319-341. Retrieved from http://dx.doi.org/10.1080/0969594X.2011.555329

Bennett, R. E. (2010). Cognitively based assessment of, for, and as learning (CBAL): A preliminary theory of action for summative and formative assessment. *Measurement: Interdisciplinary Research and Perspectives*, *8*, 70-91. doi: 10.1080/15366367.2010.508686

Deane, P., & Song, Y. (2014). A case study in principled assessment design: Designing assessments to measure and support the development of argumentative reading and writing skills. *Psicología Educativa, 20*, 99-108.

de la Torre, J., & Minchen, N. (2014). Cognitively diagnostic assessments and the cognitive diagnosis model framework. *Psicología Educativa, 20*, 89-97.

Graf, E. A. (2009). *Defining mathematics competency in the service of cognitively based assessment for grades 6 through 8* (Research Report 09-42). Princeton, NJ: Educational Testing Service.

Hansen, E. G., Mislevy, R. J., & Steinberg, L. S. (2008). *Evidence-centered assessment design for reasoning about accommodations for individuals with disabilities in NAEP reading and math* (Research Report 08-38). Princeton, NJ: Educational Testing Service.

Hines, S. (2010). *Evidence-centered design*: *The TOEIC® speaking and writing tests* (Research Report 10-07). Princeton, NJ: Educational Testing Service.

Huff, K. (2010). The promises and challenges of implementing evidence-centered design in large-scale assessment. *Applied Measurement in Education. 23*, 310-324.

Huff, K., Alves, C. B., Pellegrino, J., & Kaliski, P. (2013). Using evidence-centered design task models in automatic item generation. In M. J. Gierl & T. M. Haladyna (Eds.), *Automatic item generation - Theory and practice* (pp. 102-118). New York: Routledge.

Messick, S. (1989). Validity. In R. L. Linn (Ed.), *Educational measurement* (3rd ed., pp. 13-104). Washington, DC: American Council on Education.

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

Mislevy, R. J. (2006). Cognitive psychology and educational assessment. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 257-306). Washington, DC: American Council on Education/Praeger.

Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2003). *A brief introduction to evidence-centered design* (Research Report 03-16). Princeton, NJ: Educational Testing Service.

Mislevy, R. J., Almond, R. G., Yan, D., & Steinberg, L. S. (1999). Bayes nets in educational assessment: Where do the numbers come from? In K. B. Laskey & H. Prade (Eds.), *Proceedings of the fifteenth conference on uncertainty in artificial intelligence* (pp. 437-446). San Francisco, CA: Morgan Kaufmann.

Mislevy, R. J., Bejar, I. I., Bennett, R. E., Haertel, G. D., & Winters, F. I. (2010). Technology supports for assessment design. In B. McGaw, E. Baker, & P. Peterson (Eds.), *International encyclopedia of education* (3rd ed., volume 8, pp. 56-65). Amsterdam, Netherlands: Elsevier.

Mislevy, R. J., & Haertel, G. (2006). *Implications of evidence-centered design for educational testing*. Menlo Park, CA: SRI International.

Mislevy, R., Haertel, G., Yarnall, L., & Wentland, E. (2011). Evidence-centered task design in test development. In C. Secolsky (Ed.), *Measurement, assessment, and evaluation in higher education* (pp. 257-276). New York, NY: Routledge.

Mislevy, R. J., & Riconscente, M. M. (2005). *Evidence-centered design: Layers, structures, and terminology.* Menlo Park, CA: SRI International.

Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (1999). *Evidence-centered assessment design.* Princeton, NJ: Educational Testing Service.

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

Mislevy, R. J., & Yin, C. (2012). Evidence-centered design in language testing. In G. Fulcher & F. Davidson (Eds.), *Routledge handbook of language testing* (pp. 208-222). London, England: Routledge.

Pellegrino, J. W. (2014). Assessment as a positive influence on 21st century teaching and learning: A systems approach to progress. *Psicología Educativa, 20*, 65-77.

Scalise, K., & Wilson, M. (2006). Analysis and comparison of automated scoring approaches: Addressing evidence-based assessment principles. In D. M. Williamson, R. J. Mislevy, & I. I. Bejar (Eds.), *Automated scoring of complex tasks in computer-based testing* (pp. 15-47). Mahwah, NJ: Lawrence Erlbaum Associates.

Sheehan, K. M., Kostin, I., & Futagi, Y. (2007). *Supporting efficient, evidence-centered item development for the GRE® verbal measure* (Research Report 07-29). Princeton, NJ: Educational Testing Service.

Stocking, M., & Swanson, L. (1993). A method for severely constrained item selection in adaptive testing. *Applied Psychological Measurement, 17*, 277-292.

Tannenbaum, R. J., Robustelli, S. L., & Baron, P. A. (2008). Evidence-centered design: A lens through which the process of job analysis may be focused to guide the development of knowledge-based content specifications. *CLEAR Exam Review, 19*, 26-33.

Toulmin, S. E. (1958). *The uses of argument.* Cambridge, England: Cambridge University Press.

Van Rijn, P. W., Graf, E. A., & Deane, P. (2014). Empirical recovery of argumentation learning progressions in scenario-based assessments of english language arts. *Psicología Educativa, 20*, 109-115.