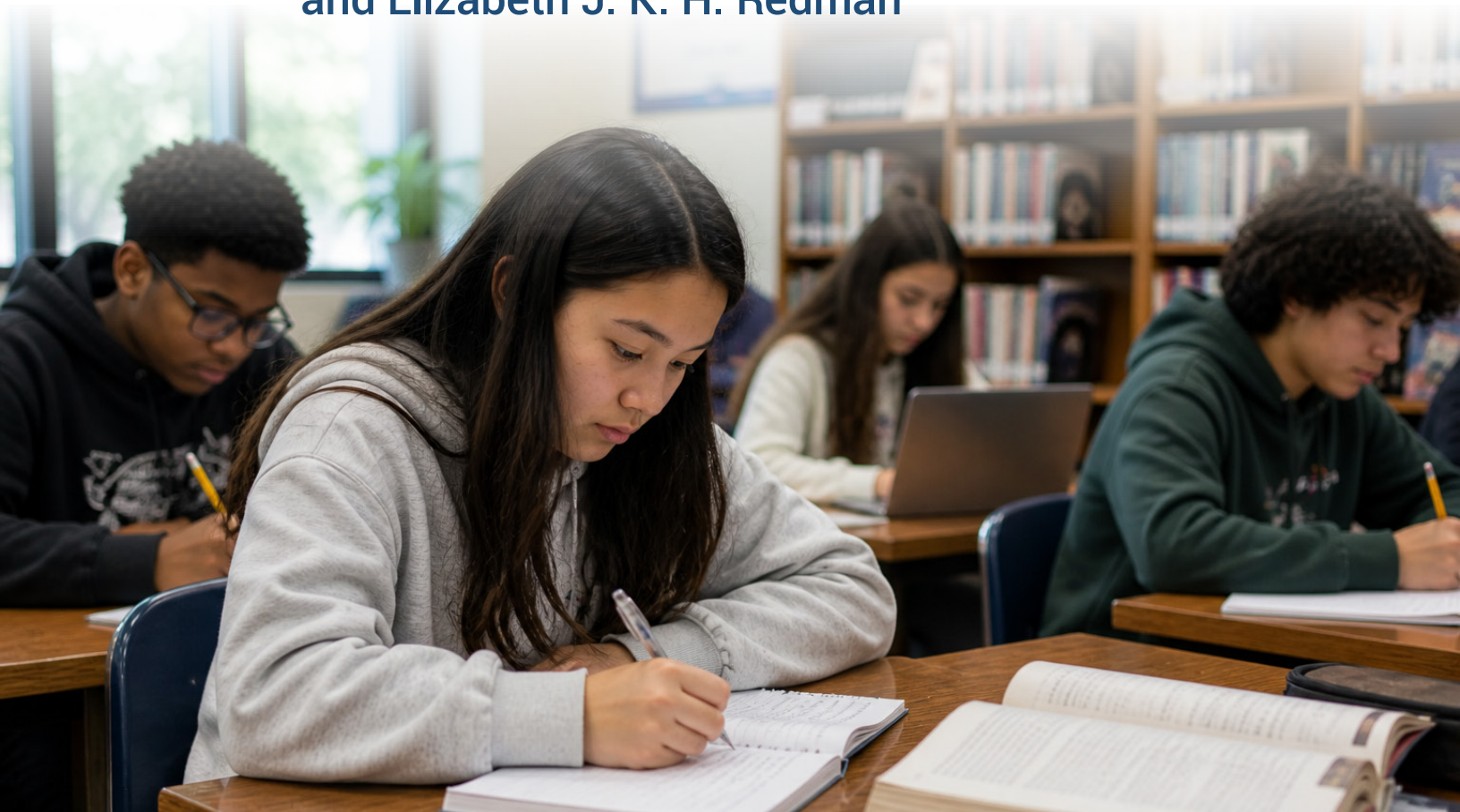
CASE STUDY

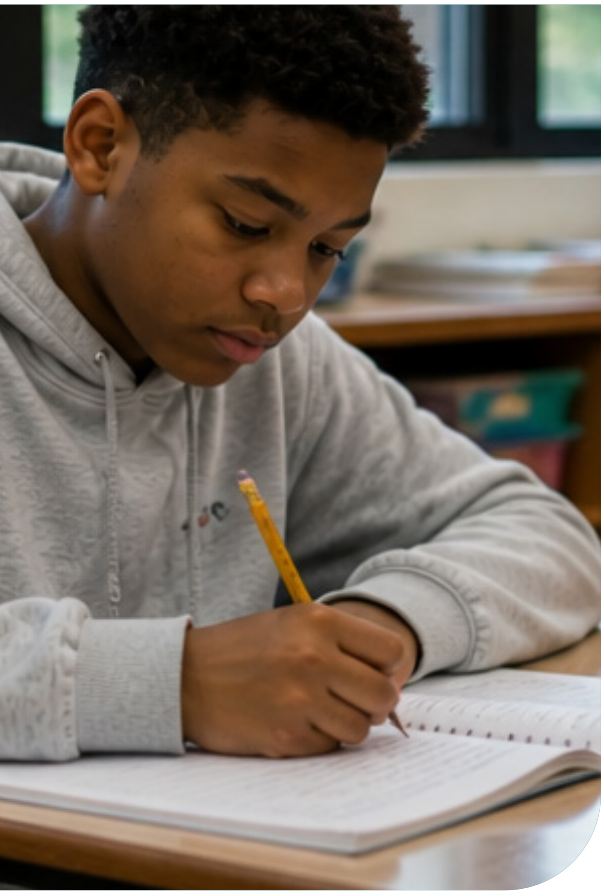
Using Learner-System Interactions as Evidence of Student Learning and Performance: Validity Issues, Examples, and Challenges

Gregory K. W. K. Chung, Tianying Feng, and Elizabeth J. K. H. Redman



Using Learner-System Interactions as Evidence of Student Learning and Performance: Validity Issues, Examples, and Challenges

Gregory K. W. K. Chung, Tianying Feng, and Elizabeth J. K. H. Redman



Abstract

This chapter explores the idea of using learner-system interactions as a source of evidence about students' learning and performance in the context of Principle 3: Assessment design supports learners' processes, such as motivation, attention, engagement, effort, and metacognition; and Principle 4: Assessments model the structure of expectations and desired learning over time. We illustrate how well-designed instructional opportunities in interactive digital environments naturally provide measurement opportunities. These opportunities can result in what we call "measurement without testing": Learner-system interactions that are designed to support students' learning are by definition observable and we believe carry the most relevant information about students' learning. Digital environments enable the collection of fine-grained behavioral data about what, when, and how a learner interacts within that environment.

However, for learner-system interactions to serve as evidence, three design challenges must be addressed: identifying the cognitive demands of the task, identifying the learning-relevant indicators of interest, and developing algorithms to transform low-level behavioral events into high-level indicators that represent learning-relevant processes. If we can observe what learners are doing as they do it and develop the methodology to accurately determine why, then that capability may help move us toward tailored, adaptive, and individualized learning for all students.

Digital environments enable the collection of fine-grained behavioral data about what, when, and how a learner interacts within that environment. The capability to automatically track the behavior of learners in digital environments has existed for years if the system was programmed to log such behaviors. The tracked behavior can range from learners' fine-grained, moment-to-moment behavior to the learners' final answer to a problem. In addition to behavior, the state of the environment can also be tracked and yoked to the learners' behavior.

The utility of tracking learners' responses has been recognized since the 1990s as a viable means to support the measurement of learners' processes and performance in interactive systems (e.g., Chung et al., 1999, 2002; O'Neil et al., 1997; Williams & Dodge, 1993; Young et al., 1997) using software sensors (Chung & Baker, 2003) and physical tasks using hardware sensors (e.g., Chung et al., 2021; Nagashima et al., 2009). Such data capture capability is routinely implemented in educational technology applications such as games, intelligent tutoring systems, training simulations, digital assessments, and large-scale standardized testing programs such as National Assessment of Educational Progress (NAEP) (Bennett et al., 2007; National Center for Education Statistics, 2012, 2020) and PISA (Foster & Piacentini, 2023; Organisation for Economic Cooperation and Development, 2014, 2021, 2023).

One of the most important reasons for tracking learners' behavior is to address questions related to how learners performed on a task, the processes they used to complete (or not) the task, and perhaps most importantly, why they performed the way they did (Feng & Cai, 2024; Jiao et al., 2021; Lindner & Greiff, 2023; Zumbo et al., 2023). Before we can address these questions, at least two conditions need to be satisfied: (a) availability of data on learners' responses in the interactive task *such that those data reflect learners' intentional behavior,* and (b) availability of information on the design features of the task, whether to promote learning or to test learners' knowledge or skills. While these two conditions are apparent for any assessment, it is less obvious how to satisfy them when the task is interactive and involves cognitive demands, including content knowledge, reasoning, and problem-solving processes.

Despite the long history and widespread use of digitally collected process data, there remain challenges in nearly every step of the analytics process: from specifying what behavior to record, how to capture it, the storage format, indicator specification, algorithm development, task design to support measurement, user interface design to support measurement, and incorporating theory into the entire endeavor. Lindner and Greiff (2023) outline key challenges and best practices for the use of process data for assessment purposes. They point out the need for a top-down approach to the design of assessments to ensure theory-grounded interpretation and analysis efficiency, and highlight the labor-intensive nature of process data analyses, including extensive data preparation.

In this chapter, we conceive of learner-system interactions—observable behavioral responses from the learner to some stimulus presented to them by the system, as well as the system's response to a learner's input—as the atomic unit of observation. In digital systems, this conceptualization flows from the capabilities enabled by software and hardware. Software or hardware can be developed to detect and log the learner's actions and system context (i.e., events and states) at the moment the action occurred and then save this packet of information to an external store.

Conceiving learners' interaction as an observation allows us to adopt well-established analytical frameworks and tools from measurement science. If the observation (or collection of observations) is used as a measurement, we can adopt a validity perspective to address issues related to the design and use of learner-system interactions and the design of tasks that can yield informative interactions. We use games as the specific context because of the complexity of interactions available in games (Chung, 2015; Chung & Feng, 2024; Lindner & Greiff, 2023).

A second reason to conceive learners' interactions as observations is that this conception directly addresses two of the Design Principles for Assessment in the Service of Learning:

- Principle 3: Assessment design supports learners' processes, such as motivation, attention, engagement, effort, and metacognition.
- Principle 4: Assessments model the structure of expectations and desired learning over time.

Conceiving learner-system interactions as observations focuses attention on the relation between the design of an interactive task and the learner's responses. Regardless of whether the measurement target is learning, motivation, attention, engagement, effort, or metacognition, how a learner responds to task demands is dependent on the degree to which the task is able to elicit a response representative of the target construct. Appropriate inferences of the learner-system interactions are dependent on the fidelity of the task. The key leverage afforded with learner-system interaction data is that fine-grained behavioral data are now available and using the observations as data should lead to a close examination of the alignment of the learner-system interactions, task design, and target construct—analogous to a test content analysis but at a much finer grain size (i.e., the level of the learner interacting with the digital task). As noted by S. Sireci (personal communication, December 20, 2024), similar to how close attention is given to how well a test represents the construct and how items are designed to measure the construct, learner-system interactions may be "another potential manifestation of the construct and the "new development" is how to capture the intended behaviors and ensure recording of the construct-relevant log data."

In the remainder of this chapter, we first define and present a detailed example of what we mean by learner-system interaction, demonstrating that even a simple game developed for preschool children has a rich set of interactions. We then discuss validity issues and underlying assumptions related to using learner-system interaction as an observation, highlighting the process of going from low-level clicks to an indicator. We then present a detailed example of the design process that led to a game design in which the game mechanics, originally designed to promote learning, could also serve a measurement function. Next, we discuss the challenges involved in using games for measurement purposes. We end the chapter with a brief discussion of outstanding issues and the relation of learner-system interactions to assessment in the service of learning.

Learner-System Interactions as the Atomic Unit of Observation

Modern digital systems are designed to attract and maintain users' attention, and interactivity is a key design feature. For example, learning games are highly interactive, making maintaining learners' engagement a critical design priority. With little engagement, there can be little learning, no matter the quality of the instructional material (Roberts et al., 2016). In contexts where users have a choice among different media, users will choose media designed to have more rather than fewer engagement elements (Roberts et al., 2016). An essential component of engagement is interactivity, which refers to the degree to which learner and system responses depend on each other (Domagk et al., 2010; Janlert & Stolterman, 2017; Kennedy, 2004; Plass et al., 2012).

Why Learner-System Interactions?

Using interactions as a potential source of evidence about learning is attractive for three reasons. First, as a practical matter, interactions can be captured via the software in digital systems. The software can be instrumented to log interactions. Well-designed instrumentation takes into account both the target cognitive demands and what the task allows learners to do (e.g., to engage in interactions that promote learning, to apply their prior or newly acquired knowledge, or to require reasoning or problem-solving to complete the task successfully).

The second reason for using interactions as a potential source of evidence is based on classroom interaction research, which shows robust findings that the nature of interactions between and among teachers and students can influence student learning (e.g., Greer & McDonough, 1999). Furthermore, a reciprocal relationship is established by the participants in the interaction, each being influenced by the other's action and the setting within which the interaction occurs (Young et al., 1997). Thus, how participants interact can determine what is learned (or not) and whether the interaction is productive (or not) (Young et al., 1997). The nature of the interaction—the extent to which a specific interaction episode is productively (or unproductively) related to the target outcome—helps explain why some students profit from instruction while others do not. A striking example is Webb's (1983) reanalysis of classroom interaction variables, which showed that examining only general interactions (e.g., giving or receiving help) led to no relation with achievement. However, when Webb recoded the interactions by type of help, the data revealed significant relations between the type of interaction and achievement. The general finding that the quality of an interaction carries information about students' learning strongly suggests that learner-system interactions are promising sources of evidence of learners' knowledge and potential learning processes.

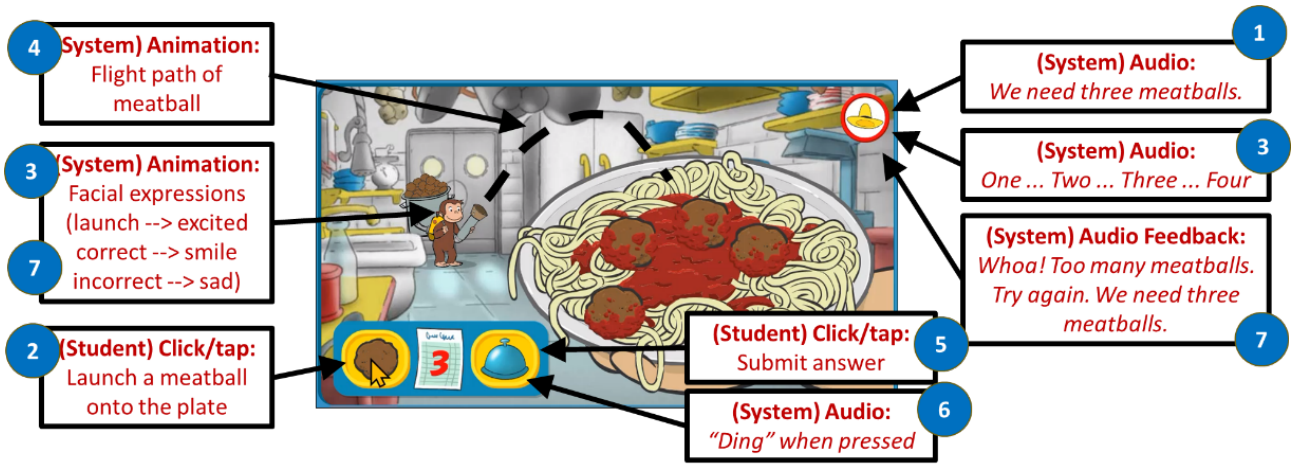
Third, the general methodology of behavioral observations has a long and robust research tradition. Interactions have been used as a data source in studies examining parent-child interactions, couples' communication patterns, teamwork processes, and classroom instruction (e.g., Bakeman & Gottman, 1997; Gottman & Notarius, 2000; Ostrov & Hart, 2013). Learner-system interactions are another form of behavioral observation using a technology-based collection of fine-grained behavior.

Example of Atomic-Level Interactions in a Learning Game

To provide a concrete example of what we mean by learner-system interaction, we present a simple illustrative example using *Meatball Launcher* (https://pbskids.org/curiousgeorge/busyday/meatballs/), a popular PBS KIDS game designed to expose preschool children to counting one to 5 objects upon request. The goal of *Meatball Launcher* is for players to add the number of meatballs specified by the target number shown on the screen to a plate of spaghetti. As shown in Figure 1, the level starts with a system voice-over giving directions (no. 1 in Figure 1). Players first click on the meatball (no. 2), and the system announces the current meatball count (no. 3). Curious George is the name of the monkey, and his facial expression changes from neutral to excited (no. 3, system animation), and he launches a meatball. The meatball flies from George to the plate (no. 4). The player can click on the meatball any number of times, even beyond the target number. When the player (presumably) thinks they have reached the target number, they click on the bell. The system responds with a "ding" (no. 6). The system then gives feedback to the player in two ways: a voice-over stating the attempt was correct or incorrect, and George's face (a smiling face for a correct solution and a sad face for an incorrect solution). The game automatically advances to the next level if the player is correct.

A close inspection of *Meatball Launcher* reveals the range of interactions in one level. For measurement purposes, the critical learning-system interactions are (a) clicking on the meatball button (no. 2 in Figure 1), (b) clicking on the bell (no. 5), and (c) solution correctness (no. 7). *Meatball Launcher* was instrumented from a measurement perspective (i.e., what game interactions could indicate players' counting skills?) and under the assumption that skill development could be described by speed and accuracy. Thus, the following information was collected: timestamp of event, round number, target number, solution attempt, correctness of attempt, and text of the system feedback. From these data, we derived indicators of game progress as a proxy for speed (i.e., mean time per round and maximum round reached) and game performance as a proxy for accuracy (i.e., number of correct first attempts, number of correct attempts overall, and number of incorrect attempts).

Figure 1. *System and Learner Interaction Elements in* Meatball Launcher

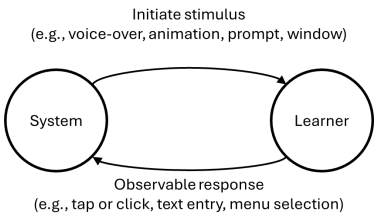
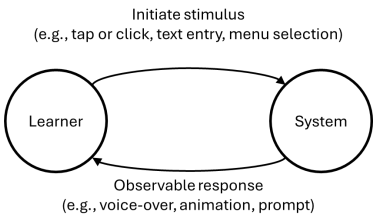


Note. https://pbskids.org/curiousgeorge/busyday/meatballs/

System-Initiated Interaction.

The left panel of Figure 2 shows a system-initiated interaction, where the system first presents some stimulus to the learner, and the learner responds to the stimulus through an action allowed by the user interface. The system-initiated interaction cycle represents a task design where the system needs input from the learner before the system can progress in the game, simulation, or assessment. The form of the learner's response is determined by the task design and expressed through a user-interface action.

Figure 2. *System and Learner-Initiated Interactions and Examples: Complete Interactions*

System-initiated Interaction	Learner-initiated Interaction
	
• *Meatball Launcher* sequence of events example • System audio (no. 1, directions) • Learner response (no. 2, clicks on meatball button) • System audio (no. 3, counts) • System animation (no. 3, facial expression) • System animation (no. 4, meatball flying) • Online multiple-choice example • The system presents the stem, options, and submit answer button • The learner responds by selecting an option	• *Meatball Launcher* sequence of events example • Learner responds (no. 5, clicks on the bell) • System audio (no. 6, ding) • System audio (no. 7, facial expression) • System audio (no. 7, feedback on correctness) • Online multiple-choice example • The learner clicks on the submit answer button • The system acknowledges the submission of the answer but does not provide correctness feedback

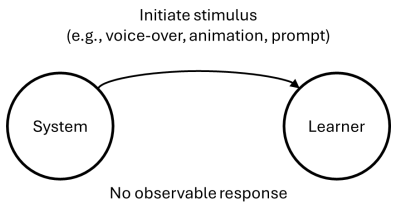
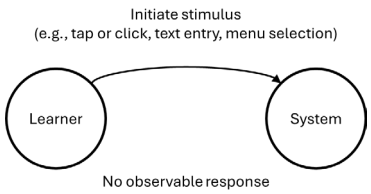
A typical system-initiated interaction is for the system to present a dialog or modal window. The window prompts the learner to make a decision or provides information, and the window cannot be dismissed without the requested action. In an online multiple-choice test, a stem is presented with a multiple-choice item, and the learner chooses an item option. The system can use the learner's inputs to determine the next item to show (in a computer-adapted test) or for feedback (e.g., to acknowledge acceptance of the answer submission). Similarly, in a game, the game may present a dialog for players to select a level to play. Note that the stimulus can be explicit or implicit and use audio, text, images, or graphics. Regardless of the media used, the underlying interaction is initiated with the system and ends with a learner response.

Learner-Initiated Interaction.

The right panel of Figure 2 shows a learner-initiated interaction, where the learner performs some action and the system responds to that action. The learner-initiated interaction cycle allows a task design to be open-ended and allows the learner to decide what action to take and when. The learner's input and the system's response formats are determined by the task design and expressed through the user interface.

The learner-initiated interaction is well-suited for open-ended tasks, where learners may have many potential actions—and thus decisions—to make. This type of task design is often used in digital performance tasks, games, and simulations. If the sequence of operations is important, then this task design can reveal the extent to which learners know or can determine the proper sequence. Likewise, if efficiency is important, then this task design may reveal economy of expression and differentiate between learners who know an existing solution to a problem from those who do not, and from learners who learn the solution over the course of the task. Finally, interactions may be incomplete where the system or learner initiates an action but no response is given as shown in Figure 3.

Figure 3. ***System and Learner-Initiated Interactions and Examples: Incomplete Interactions***

System-initiated Interaction	**Learner-initiated Interaction**
	
• *Meatball Launcher* sequence of events example • The three system-initiated events after the user clicks on the meatball button are examples of this type of interaction where no user inputs are expected. • Online multiple-choice example • Ancillary directions may be given as part of the task (e.g., check your work; keep track of remaining time) where no input is expected of learners.	• *Meatball Launcher* sequence of events example • The learner clicks on parts of the screen that are not designed to receive learner responses. Such off-clicks can be helpful when examining user-interface design (e.g., whether a button is too small or the hit point ambiguous). For example, learners unfamiliar with the interface may think clicking on George will initiate the launching of meatballs. • Online multiple-choice example • Many off-clicks may indicate learners are exploring the system, are bored, or want to exit the test.

Measurement Implications

Conceptualizing learners' behaviors in a digital system as interactions allows us to interpret behavior as a manifestation of cognition—one's choices in a task reflect one's knowledge and thinking. Because we can only observe learners' behavior and must infer the learning processes they use, tasks create situations for learners to demonstrate the use of the target cognitive demands. There needs to be user-interface elements that allow learners to interact with the system in a way consistent with the cognitive demands. For example, suppose a learning game is intended to promote problem-solving and we want to measure learners' problem-solving processes. In that case, the game should present situations where the learner is unlikely to immediately know the solution and provide information sources (e.g., resources, information, feedback, hints, and tutorials) that learners need access to and understand to solve the problem. To observe the problem-solving process, the information sources should be accessible via interactive user-interface elements instrumented to log the interactions. Learners will likely exhibit intentional behavior if the information sources are required to determine the solution to the problem. Problem-solving indicators can be derived from the learner-system interactions directly or through a transformation process.

The utility of learner-system interactions is threefold. First, viewing tasks as composed of learner-system interactions helps us describe general task features suitable for measuring different cognitive demands. Learner-initiated interactions (See Figure 2, right panel) may be well suited for assessing learning processes when the learner decides what to do next in a task. This design is akin to performance assessments. System-initiated interactions (See Figure 2, left panel) may be suitable when measuring specific knowledge or skills.

Second, learner-system interactions support quantitative analysis of the learner's performance and processes in a task. Some interactions may be directly evaluated (e.g., the learner's submission of an answer can be evaluated as correct or not in the example game *Meatball Launcher* as well as a multiple-choice task). Some interactions may need to be part of an algorithm that uses sets or sequences of interactions to derive an indicator, such as when examining learning over the course of the task. Regardless of the level of aggregation and transformations, the learner-system interaction is the atomic unit of observation.

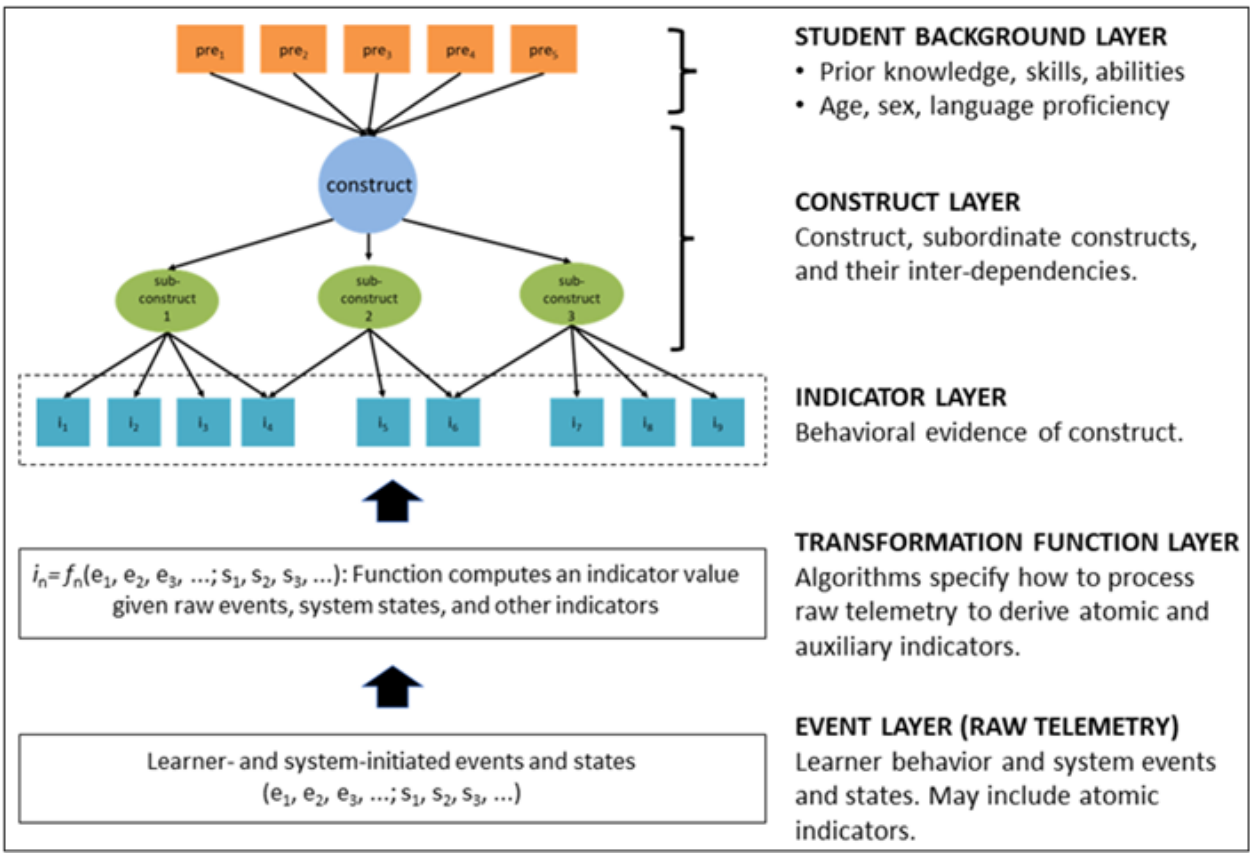
Finally, digital systems directly record interactions whenever they occur, unlike traditional behavioral observations that typically use video or audio recording and rely on human coding of the data using a rubric. Data are generated each time learners perform an action. This data collection method can produce hundreds of interactions per learner regarding their game behavior. Even though these data "come for free," the situation creates a new set of validity concerns. In video coding, the transformation of events into a category or score is through the rater's interpretation of the scene relative to the rubric. The rubric can be inspected and critiqued in light of theory or construct. Inferencing is left to the human rater. In contrast, generating an indicator or score from interaction data is through the coding of algorithms. Data elements are extracted from the raw interactions, transformed, and eventually, a quantitative value is computed for a learning-related variable.

Validity Issues Related to Using Interactions as a Source of Evidence: From Clicks to Constructs

The process of transforming learner-system interactions into an indicator is shown in Figure 4 (Chung & Feng, 2024). In their discussion related to Figure 4, Chung and Feng expressed concern about the difficulty and amount of programming required to transform low-level interaction data into meaningful indicators. The authors asserted the (strong) assumptions encoded in the indicator development process. The assumptions were based on the validity issues identified by the American Educational Research Association, American Psychological Association, & National Council on Measurement in Education (AERA, APA, & NCME, 2014), Baker (1997), and Mislevy et al. (2015) and applied to the situation of using learner-system interactions as evidence of learning. The following list of assumptions underlying the use of learner-system interactions for measurement purposes is from Chung and Feng (2024):

- The construct is an abstraction of human cognition and is not directly observable (AERA et al., 2014; Cronbach & Meehl, 1955; Messick, 1995).
- The construct has well-defined boundary conditions (AERA et al., 2014; Messick, 1995). For example, a clear definition describes the domain, the dependencies between and among the components of the domain, and an explicit relation between the construct (or subconstruct) and observable responses (Mislevy et al., 2015).
- The existence of the construct manifests in learners' generating observable responses. Responses include neurological, physiological, and motor responses at the lowest level; however, we are referring to the level of intentional behavior, speaking, or writing (e.g., see evidence-centered design, Mislevy et al., 2015).
- Learners are malleable (i.e., may learn) with respect to the construct and components of the construct, and their learning is influenced by what they observe, experience, perceive, or imagine. While the possible stimuli span the five senses, we are referring to visual, audio, and haptic inputs typical of learning settings, which may involve various types of static or interactive media, technology, real-world situations, or other people. By malleability, we mean, for example, that learners' skill in adding unit fractions can improve under certain conditions (e.g., when "good" instruction is provided and the learner is attentive to the instruction, exerts effort at processing the instruction and uses productive learning strategies).
- Learners' observable responses covary with changes in the construct in explainable and predictable ways. This assumption directly impacts measurement. If the learners' responses do not change even if they are learning, then it will be impossible to detect learning no matter how sensitive the measurement instrument is. If the learners' responses change for reasons unrelated to the construct, then the measurements will have little meaning. Finally, if learners respond unpredictably when the construct changes, the measurements will be unreliable and may indicate poor construct definition, poor choice of what is observed, or both.

Figure 4. *Conceptual Framework of the Relations Among Telemetry, Algorithms, and Indicators*



Note. Telemetry is synonymous with learner-system interaction.

Event Layer

The lowest layer in Figure 4 is the event layer. The event layer comprises the learner-system interactions, the atomic unit of observations. The learner-system interactions are fine-grained data generated when a user behavior occurs. Note that the software must be instrumented to capture each learner-system interaction. Without instrumenting the software, no behaviors can be logged.

The event layer is important because it provides the raw data on which all other layers are built. The choice of what interaction to log affects what indicators can be derived, what analyses can be conducted, and ultimately, what inferences can be drawn about players. The key design guideline is to log learner and system interactions representing learners' productive and unproductive use of the target knowledge or learning process. State information at the time of the event helps to disambiguate the action or aid in the subsequent creation of auxiliary indicators. See Chung (2015) for additional telemetry design guidelines.

Transformation Layer

The transformation layer in Figure 4 defines indicators in terms of algorithms. Given a definition, the algorithm derives indicator values from the data provided by the event layer.

The transformation layer is important because it provides inputs to a statistical model or procedure, from which inferences of learning are drawn. The transformation layer highlights that a processing stage is needed to transform raw interaction data into indicators—a stage that is often unreported, downplayed, or ignored in the literature. This processing stage is where coding and algorithm development occur. The specifications for the algorithms may be based on theory (i.e., hypothesized behavior under certain conditions), prior research that describes actual behavioral responses under certain conditions (e.g., see Feng's [2019] implementation of Metz's [1993] descriptions of misconceptions related to a pan balance), or data-driven approaches. The algorithm must be made available for inspection and critique because in this layer, solely behavioral responses (i.e., learner-system interactions) are transformed into indicators of learning processes and states that are otherwise unobservable.

The following section presents a detailed example of a game developed to promote learners' understanding of fractions. Baker's model-based assessment framework (Baker, 1997) was used as the general design approach, and Mislevy's evidence-centered design (Misley et al., 2015) was used to focus the linkage among observables, work products, and domain model.

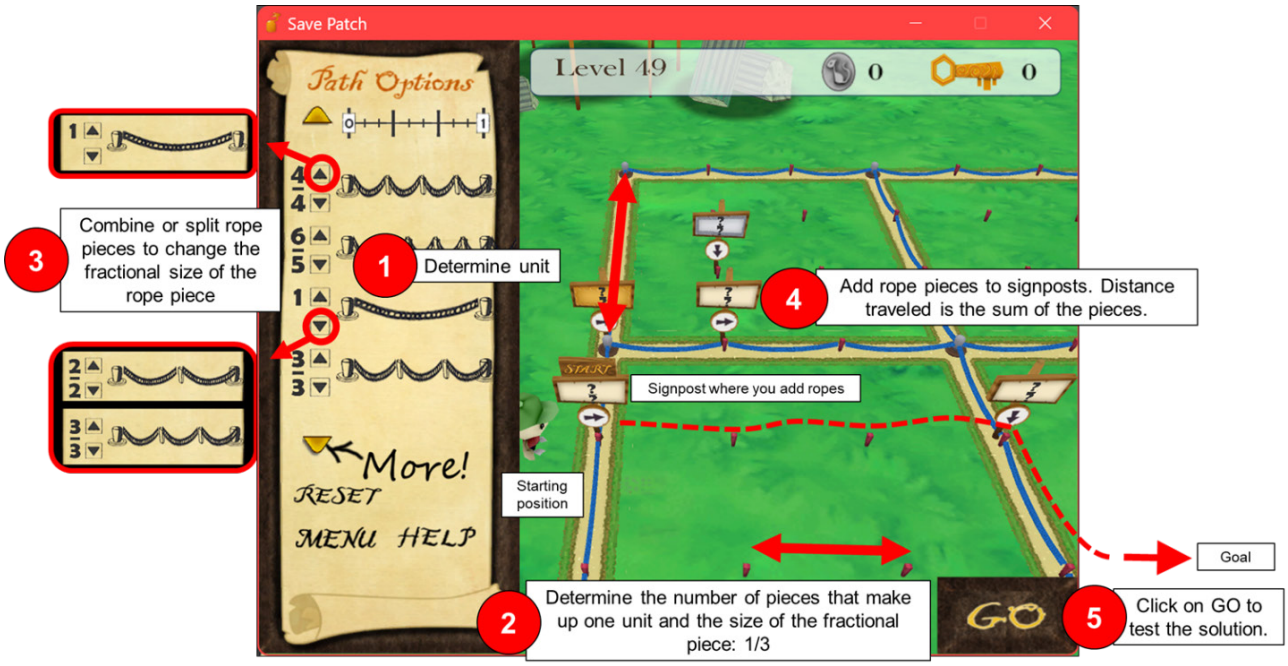
Illustrative Example

The example game *Save Patch* was developed by the Center for Advanced Technology in Schools (CATS) & CRESST (2012). *Save Patch* was designed to support middle-schoolers' learning of rational number equivalence (i.e., fractions). We assert that behaviors, expressed as learner-system interactions during the process of learning, can also be used for measurement purposes. The more the instruction is aligned to explicit learning goals, the more information the interaction carries because the learner will be engaged in processes directly related to the target learning constructs.

Save Patch Game

Save Patch is an example of a game with game mechanics designed to address target learning outcomes directly. This example also shows how game mechanic interactions, originally designed to facilitate learning, can be used for measurement purposes. Based on the 2008 National Mathematics Advisory Panel (NMAP) report, UCLA/CRESST designed and developed a series of games (with *Save Patch* as the best example) to target two core ideas. The first idea is that all rational numbers (integers and fractions) are defined relative to a single unit quantity. The second idea is that rational numbers can be summed only if the unit quantities are identical (e.g., 1/4 + 3/4 is permissible, but 1/2 + 3/4 is not because the units or sizes of the fractions are unequal). Figure 5 shows a screenshot of the game.

Figure 5. *Screenshot of Save Patch Level 49 of 52*



User Interface, Gameplay, and Learner-System Interactions. In *Save Patch*, the setting is an archeological dig site, and the player must help the game avatar retrieve a cat statue some distance away. The avatar can only travel along a one- or two-dimensional grid. The player lays out a path for the avatar by connecting signposts with ropes. The learner-system interactions include selecting the rope piece size and adding the correct number of rope pieces to a signpost. A submit button is included so the player can test the solution. The game only allowed rope pieces of the same size (denominators) to be added together. Fraction manipulation complexity was increased over levels through the grid spacing and rope sizes. For example, in more complex levels, players needed to subdivide two ropes until both ropes had pieces of the same fractional size (i.e., same denominator) (e.g., rope 1: split 1 into 6/6; and rope 2: split 1/2 into 3/6). Table 1 shows the relation between the fraction concepts, the game representation, and the associated learner-system interactions.

Table 1. *Relation Between Fractions Knowledge and Learner-System Interactions in Save Patch*

Fractions concept [a]	Game representation	Cognitive demand and learner-system interaction
A unit can be represented as one whole interval on a number line.	The unit definition of the given grid is indicated by large gray posts (item 1 in Figure 4).	Cognitive demands: • Identify two large gray posts and understand that distance represents the unit (item 1 in Figure 4).
• The size of a fraction is relative to how a unit is defined. • The denominator of a fraction represents the number of identical fractional pieces in a unit.	• Fractional grid pieces between the large gray posts delimited by small posts [i.e., the denominator] (item 2 in Figure 5).	Cognitive demands: • Determine the size of the fractional piece between two small posts (item 2 in Figure 5). • Determine the size of the rope piece that represents the fractional piece between two small posts (item 3 in Figure 5). • Split or combine the rope to get the appropriate fractional piece size [i.e., the denominator] (item 3 in Figure 5). Learner-system interactions: • Click the up or down arrow (item 3 in Figure 5).
• The numerator of a fraction represents the number of identical parts that have been combined. • The units (or parts of units) must be identical to add quantities.	The avatar needs to travel from the starting point to the goal. The path to the goal is along the grid marked by signposts. The distance between 2 signposts is the amount of rope pieces needed. Placing the correct number of rope pieces of appropriate sizes between all signposts along the solution path beats the level.	Cognitive demands: • Determine the path from the starting point to the goal by identifying the signposts. • Determine the direction to travel between signposts. • Determine the number of rope pieces between signposts on the solution path. Learner-system interactions: • Learner: Drag a rope piece onto the signpost (item 4 in Figure 5). • System: Reject rope pieces with denominators different from the pieces on the signpost. • Learner: Click on GO to test the solution (item 5 in Figure 5). • System: After the player clicks GO, the avatar walks along the solution path, traveling the distance in the signpost. If the value is incorrect, the avatar will not reach or will overshoot the next signpost and fail. Level success is indicated with a message indicating completion.

Note. Additional information was logged with each interaction, including a timestamp and the state of the game (i.e., contextual information that includes current level, grid size, grid spacing, level solution set, and interactions-specific information such as correct or incorrect action). [a] CATS & CRESST (2013a).

Addressing Validity

Earlier, we asserted that a game designed for instructional purposes could also be used for measurement purposes. This section describes the elements that make that dual use possible. We briefly describe the design components and the resulting game and game mechanics.

Coherent Design Process.

Save Patch was part of a randomized-controlled trial to test the effectiveness of instructional games on students' understanding of rational numbers (See U.S. Department of Education, Institute of Education Sciences, What Works Clearinghouse, 2015, for a WWC review of the study design). To ensure alignment among instruction, assessment, and professional development, we identified the critical knowledge in rational numbers. The knowledge was gathered from pre-algebra ontologies (Baker, 2012), Common Core Math Standards (National Governors Association Center for Best Practices & Council of Chief State School Officers, 2010), recommendations of the NMAP (2008), and practicing mathematics teachers. A set of knowledge specifications for rational number equivalence was developed from these sources, and the specifications were used to guide the design of the game instruction, fraction knowledge measure, and professional development. Figure 6 shows a snippet of the specifications.

Figure 6. *Excerpt of the Knowledge Specifications Used in the Design of Save Patch*

CATS Knowledge and Item Specifications: Rational Number Equivalence

		Computational Fluency: Students can execute procedures in the domain without the need to create or derive the procedure. Fluid performance is based on recall of patterns or other well established procedures, and is fast, automatic, and error-free. *How is something done?*		**Conceptual Understanding:** Captures demonstration of understanding of the mathematical concepts. *Why is something done?*	
Rational Number Equivalence Knowledge Specifications		**When presented with… *(Assessment Stimulus)***	**Students should be able to…**	**When presented with… *(Assessment Stimulus)***	**Students should be able to…**
1.0.0. Does the student understand the importance of the unit whole or amount?					
	1.1.0. The size of a rational number is relative to how one Whole Unit is defined.	Any rational number	Place it on a number line relative to the whole interval explicitly (0 and 1 labeled) or implicitly (0 and an integer other than 1 labeled) defined.	Apparent contradictions involving rational number such as ¾ < ½ or ½ does not equal ½	Explain that the contradiction can be resolved if their relative wholes must be equal when comparing.
		A unit whole (interval, volume, area, etc.)	Show how much of the whole must be shaded to represent a fractional amount.		
	1.2.0. In mathematics, one unit is understood to be one of some quantity (intervals, areas, volumes, etc.).	A histogram of a certain quantity represented by discrete objects	Identity the unit that each single discrete object represents (e.g. each rose represents thousands of flowers sold on Valentine's Day).	A relationship between a real world measure and a scale model	Explain how what size of unit to use on the model to accurately represent the real world quantity (e.g. 1 inch equals 25 feet since the real world measure is 100 feet and the model can be up to 4 inches in length).
	1.3.0 In our number system, the unit can be represented as one whole interval on a number line.	A number line labeled with consecutive integers that may or may not include zero	Show the unit interval that fits with the given number line or accurately place another non-consecutive integer on the number line.	A number line that is labeled by skip units (2,4,6, etc.) or a line labeled by ½ units that may or may not include zero	Explain how to determine where other integer and rational values should be placed.

Note. CATS & CRESST (2013a).

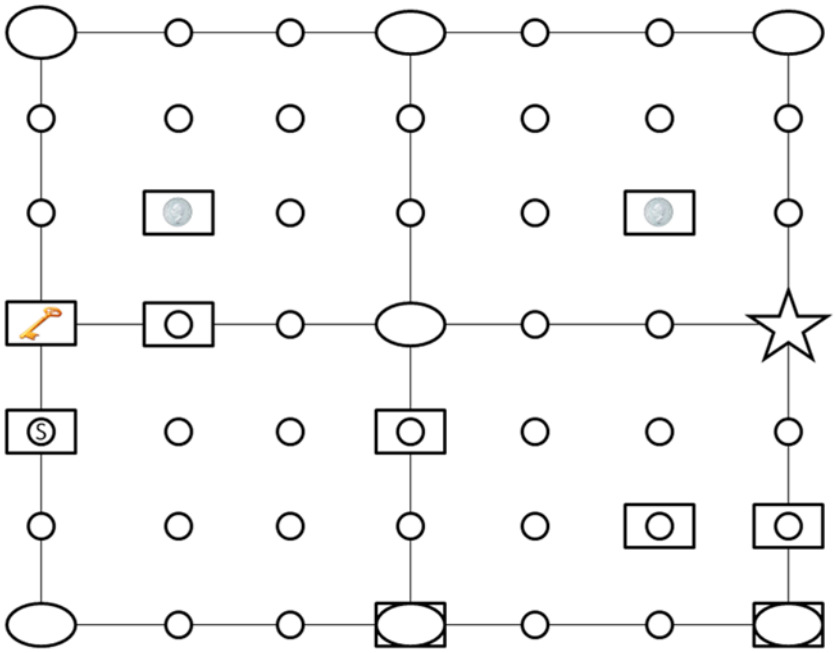
The game level progression was based on the mathematical development of fractions knowledge. The game followed a progression that introduced the game mechanics through tutorials and whole numbers. The game progressed from whole numbers to increasingly more challenging levels that involved complex fraction manipulations. Variation of practice was embedded by having multiple levels on the same topic (CATS & CRESST, 2013b). Table 2 shows the level sequencing. *Save Patch* had a total of 57 levels.

Table 2. ***Level Sequencing for Save Patch***

Math topic for each stage	Level number
Whole unit jumps; adding wholes	1
	Tutorial level on game mechanics
	2 to 4
Identifying correct denominator; scrolling to appropriate denominator, no adding fractions.	Tutorial level on fractions
	5
	Tutorial level on keys and coins
	6 to 9
Identifying correct denominator; inconsistent jumps (sometimes whole, sometimes fractional pieces)	10 to 13
Identifying correct denominator; jump over unit bar	14 to 16
Adding fractions; given correct size ropes	17 to 20
Adding fractions; jumps larger than one unit	21 to 24
Test levels: Add fractions when given different piece sizes.	25 to 28
Conversion of ropes; scroll given the wrong size	29 to 35
Conversion; Given smaller (e.g., 1/6) ropes to put on larger (e.g., 1/3) grid	36 to 38
Conversion; whole jumps, given fractional pieces of coils	39 to 42
Conversion; Common denominator needed	43 to 57

Finally, each level was documented to record the resources, grid layout, solution(s), and relevant knowledge specifications (CATS & CRESST, 2013b). Figure 7 shows the design for level 49. The representation describes the level, facilitating review and revisions during game development, algorithm development, and analysis phases.

Figure 7. *Game Level Design for Level 49*



Level 49	Resources	Math Goals
- W x H = 2 x 2 - Grid unit = 1/3	- **6** 1/6-unit coils - **4** 1/4-unit coils - **11** 1/5-unit coils - **2** 1-unit coils - **3** 1/3-unit coils - **2** 1/2-unit coils - **10** directional signs	Conversion; Common denominator needed

Knowledge Specs Targeted in This Level

1.1.0. The size of a rational number is relative to how one whole unit is defined.

1.3.0 In our number system, the unit can be represented as one whole interval on a number line.

1.3.3. Positive non-integers are represented by fractional parts of the interval between whole numbers.

2.1.0. To add quantities, the units (or parts of units) must be identical.

2.1.2. Positive integers can be broken (decomposed) into parts that are each one unit in quantity. These single (identical) units can be added to create a single numerical sum.

2.1.3. Each whole unit or part of a whole unit (fractions) can be further broken into smaller, identical parts, if necessary.

2.2.0. Identical (common) units can be added to create a single numerical sum.

3.1.0. The denominator of a fraction represents the number of identical parts in one whole unit. That is, if we break the one whole unit into "x" pieces, each piece will be "1/x" of the one whole unit.

4.1.0. The numerator of a fraction represents the number of identical parts that have been combined. For example, ¾ means three pieces that are each ¼ of One Whole Unit.

5.1.0. The numerator is the top number in a fraction

5.2.0. The denominator is the bottom number in a fraction.

Game Solution Description

[(0/3,2/3), U, 4/12]; [(0/3,3/3), R, 4/12]; [(1/3,3/3), U, 4/12]; [(1/3,4/3), R, 16/12]; [(5/3,4/3), D, 12/12]; [(5/3,1/3), R, 4/12]; [(6/3,1/3), U, 8/12]

Game Mechanics That Closely Reflected Mathematical Operations.

Table 3 shows how the learner-system interactions described in Table 1 are transformed (or not) into indicators that can be used in an analysis procedure. Two indicators are shown: (a) game performance and progress indicators, and (b) fraction misconceptions.

The game performance and progress indicators are intended to be general indicators whose definitions can be adopted across different games and tasks. The game performance and progress indicators are intended to be general indicators, the definitions of which can be adopted across different games and tasks. Performance and progress are two common ways of describing human performance, from motor and verbal learning to outcomes, to education and training outcomes (e.g., Ackerman, 1990; Anderson, 1982; Fitts & Posner, 1967; Heitz, 2014). In games that have instructional sequences and learning-based interactions, we have consistently found these performance and progress measures to be associated with external criterion measures in expected ways (e.g., learners who know more, compared to learners who know less [as measured by an external criterion measure of knowledge or skill], demonstrate higher performance in the game, commit fewer errors, and spend less time on levels) (Chung & Feng, 2024).

We think these "common measures" are sensitive to knowledge outcomes because the game is intentionally designed to evoke learning processes. The learner-system interaction represents learners performing actions that use their existing or to-be-learned knowledge. The game levels are sequenced, where later levels build on what was learned in earlier levels. Games that do not require the learner to demonstrate the use of the target knowledge can yield interaction data, but the data will not likely reflect the use of the target knowledge. For games that lack a learning sequence (or curriculum), associating game progress with the degree of knowledge and skill will be tenuous; game progress may be a stronger indicator of engagement than learning.

The second kind of indicator in Table 3 is fraction misconceptions. These indicators show the utility of fine-grained interaction data but also highlight the challenge of using fine-grained data (discussed in the next section). See Chung (2015) for an example of how the data were structured. The learner-interaction data packet included as much information about the situation as we believed could be useful in as many analyses as possible. Given the knowledge specifications focused on the whole unit and its composition of any number of equal-sized pieces (i.e., concepts of numerator and denominator) and that the addition operation could only be performed on pieces of the same size, we surmised it was important to record the specific numerator and denominator values along with the grid piece size for each addition operation. Further, while the correctness of the addition operation could give an overall indication of understanding, we reasoned that we would be able to identify particular misconceptions only with the exact numerator and exact denominator in relation to the specific level. Other examples that illustrate the use of fine-grained learner-system interaction data to infer cognitive processes are given in Chung and Feng (2024).

Table 3. *Selected Examples of Associations Between Learner-System Interactions and External Measure of Counting Knowledge (n = 783 to 851 middle school students)*

Indicator	Validity evidence [a]	Learner-system interactions	Transformation
Game performance and progress indicators [b]			
Correct addition of fractions	0.10**	Add a rope piece to a signpost as often as needed to travel from one signpost to the next.	Because the game interaction closely resembled the act of adding two fractions, we evaluated the learner interactions as correct or incorrect. The indicator is the total number of correct or incorrect additions over the entire game. The game already computes correctness to determine whether the rope piece is permissible. Thus, correct or incorrect additions can be logged directly with no transformations.
Incorrect addition of fractions	– 0.19**		
Number of correct first attempts at solving a level	0.55**	Click on GO to test the solution Level success or failure	Because the game directly logged the results of the solution attempt, the only transformation was to filter the data for the first attempt of each level.
Number of correct attempts at solving a level	0.43**		No transformations were needed because the game directly logged the results of the solution attempt.
Number of incorrect attempts at solving a level	– 0.45**		
Fractions misconceptions [c, d, e]			
Unitizing error		Add a rope piece to a signpost as often as needed to travel from one signpost to the next.	Because of the detailed logging of each operation (i.e., current level, grid size, grid spacing, level solution set, correct or incorrect action), unique tokens could be formed encoded the adding rope interaction event and the specific fraction values of the rope chosen by the learner. The tokens were then clustered using a fuzzy cluster algorithm. The clusters were labeled based on how the interactions comported with the extant research on fraction misconception.
Saw as one unit	– 0.28***		
Saw as wholes	– 0.22***		
Partitioning error			
Counted hash marks	– 0.21***		
Counted hash marks and posts	– 0.29***		
Unitizing and partitioning error			
Saw as one unit and counted hash marks	– 0.37***		
Saw as one unit and counted hash marks and posts	– 0.50***		
Iterating error			
Wrong numerator	– 0.44***		
Converting to wholes error			
Saw as a mixed number	– 0.11**		

*$p < .05$. **$p < .01$. ***$p < .001$.

[a] Spearman nonparametric correlation (ρ) between the indicator and an external measure of fractions knowledge. See Vendlinski et al. (2010) for a description of the measure. [b] Chung and Roberts (2018). [c] Chung and Feng (2024). [d] Kerr & Chung (2012). [e] Kerr (2014).

Challenges

In this section, we address what we see as three major challenges of developing indicators from learner-system interactions. While these challenges are discussed in relation to games, in our experience, the challenges surface whenever fine-grained data are used to infer high-level processes. Regardless of whether the collection system is software or hardware, or the task is game-based or not, we believe the challenges remain the same.

01

Challenge 1: Identifying the Cognitive Demands of the Game

Given a learning game, how do we examine the game and identify what the game is intended to teach? How do we determine whether a learning-system interaction (i.e., game mechanic) is useful for measurement? While game developers may be the obvious first choice, they are typically not trained in the learning or measurement sciences. The vocabulary used in the learning and measurement sciences may not mean the same thing to game designers as it does to learning and measurement specialists.

Addressing Challenge 1: Feature Analysis

Thus, one method to better understand the learning opportunities presented by a game is through an in-depth qualitative analysis ("feature analysis") of the game and its interaction opportunities. Feature analysis is the qualitative coding of an object (e.g., game, video, test item, intervention, assessment setting) against a set of properties. The properties are defined a priori (though often refined during the analysis process) and reflect aspects of the intervention hypothesized to influence student learning. The concept of feature analysis has its roots in Gordon (1970), in which he mentions qualitative analysis of assessments to describe cognitive functions to identify learning experiences required to promote positive academic outcomes more effectively. Subsequent development by Tatsuoka (1983) quantified this approach via her rule-space methodology, which mapped test items to a set of knowledge attributes to create a "Q-matrix." This item-attribute Q-matrix could then be subjected to quantitative analysis to examine, for example, the particular knowledge components (e.g., basic concepts and operations in fractions and decimals) (Tatsuoka et al., 2004). A key theoretical contribution of Tatsuoka's work was that the test items possessed certain attributes that could be reliably identified from a cognitive or knowledge perspective. A key CRESST insight was that this approach of describing features of the assessment space could be extended to the instructional space (e.g., games, videos, tasks) and the setting in which the child is observed (e.g., the classroom) or any other object or element that is hypothesized to affect a child's learning (Baker, 2015a, 2015b; Baker et al., 2015; Chung & Parks, 2015; Redman & Kennedy, 2017). When statistical analyses are conducted on the relationship between the features and performance, the results may identify potential growth areas for students, identify content areas amenable to instruction, and provide a method for comparability and prediction of student performance (Baker, Cai et al., 2015; Baker, Madni et al., 2015).

For games, we use a standardized set of features that describe the interaction opportunities of a game (Chung & Parks, 2015; Redman & Kennedy, 2017). These are features related to the type of input the player is allowed to submit to the game, the kind of feedback provided to the player by the game, and how the game presents the targeted constructs. This feature set is based on media research, instructional practices, and CRESST's experience creating and studying educational games. Some feature set iteration may be necessary during analysis as salient features emerge that were not initially included in the list. Feature set iteration generally occurs at the beginning of the analysis process before the bulk of the games have been rated. However, revision of the feature list may be warranted even in later stages of the analysis if a salient novel feature is discovered in a new game or there is cause to amend a definition to more accurately and reliably rate the games. Whenever the feature list is revised, all already analyzed games must be re-rated with the new features and definitions in mind. At its core, the process endeavors to develop a stable and inclusive feature list that can be reliably applied across various games.

The utility of having a qualitative method of evaluating a game was examined by Redman et al. (2023). One objective of Redman et al. was to investigate whether games classified as having more learning potential, compared to games classified as having less learning potential, would show in-game performance gains (presumably due to learning of the content). An initial feature analysis of 15 existing PBS KIDS games with high data quality was conducted, a process that yielded 12 games that had alignment of learning goals, gameplay, and measurement potential. A more in-depth feature analysis was then done using the features in Appendix A. The analysis resulted in three games classified with a learning potential of not likely or less likely, and four games classified as likely. Data collection occurred over five months and analysis was conducted with data from five of the games. Learning was modeled with a two-timepoint latent variable model where the inputs to the model were gameplay performance indicators. The two games classified as *not likely* or *less likely* to have a learning potential had a change in latent ability scores of .08 and .12, with both games having effect sizes of 0.07. The two games classified as *likely* had a change in latent ability score of 0.30 and 0.42 (the third game's model did not converge), with effect sizes of 0.59 and 0.56, respectively.

These results are consistent with the idea that the qualitative rating of games using learning-focused features in Appendix A can detect a game's learning potential a priori. Stated another way, the features in Appendix A—particularly the instruction and feedback features—provide guidance on the learner-system interactions that may be sensitive to learning.

Challenge 2: Identifying Potential Game-Based Indicators and Developing Algorithms to Derive Those Indicators From the Atomic Units of Evidence

The crux of high-quality learning process data is challenge 2. Challenge 2 arises because what constitutes data is wholly defined by the software that is implemented to capture and record the data. Early decisions about what behavior to log and at what granularity, when to log it, and what format to store the data can substantially impact downstream processes.

Deciding on What Behavior to Log

The first step is necessary but insufficient in extracting meaning from fine-grained learner-system interaction data. Defining what constitutes an atomic unit is crucial for subsequent analysis. Too fine-grained data logging (e.g., cursor movements) may result in unwieldy data and require extensive post-processing coding to reduce it to a usable form. Too coarse-grained data (e.g., logging only the solution submission) may omit highly informative behavior of learners' decisions and choices and preclude any possibility of examining fine-grained process questions. For example, in *Save Patch*, if learners' adding ropes to signposts were not recorded, or if the denominators of the rope and the current denominator in the signpost were not recorded, then it would be unlikely that any misconceptions could be identified through gameplay.

Another example is related to the fidelity of experience. For instance, game instructions are often presented through tutorials describing the game goals and game mechanics. If the tutorial is made skippable (e.g., by clicking a dismiss button)—as is often the case in games to maintain an enjoyable experience—then some players may skip the tutorial and later in the game not know what to do. The gameplay of these players may differ significantly from that of those who went through the tutorial. However, if the learner-system interactions on the tutorial were not logged, we would have no way of knowing whether the tutorial was skipped. Knowing whether learners skipped the tutorial allows us to describe learners more precisely, conduct more refined analyses about learning, and inform developers about usability issues.

Another important decision point when deciding what to log is the sampling policy. When and how frequently to sample the behavior can have essential post-processing implications. For example, logging that uses continuous sampling (e.g., 128 samples per second) may be appropriate for situations where the behavior is continuous, such as when measuring learners' fine-grained motor skills (e.g., see Nagashima et al., 2009). However, based on our experience attempting to make sense of data from learning games using continuous sampling of the entire game world, we think a more effective sampling scheme is event-based sampling that uses learners' overt behavior to trigger the logging of an interaction. Continuous sampling is simple to implement but records the state of the game world at fixed time intervals. This type of data requires substantial coding to extract events of interest. In contrast, event-based sampling requires modifying the game software and is more complex because decisions about what to log and at what granularity are necessary. Event-based sampling focuses on what interactions may be of interest a priori.

Finally, a related issue is the structure of the data logged. As a practical matter, the logging format can influence the amount of programming effort required to extract data. Design decisions about the data format include expressiveness, compactness, and with large datasets, computing and storage resources. Chung (2015) presents some guidelines on the design and implementation of telemetry, as do others (e.g., Hao et al., 2016).

Developing Indicators Rests on Algorithms and Coding

The rationale for capturing learner-system interactions is to use these interactions as inputs to algorithms to derive indicators of learning processes and outcomes. The indicators themselves can be used directly or as inputs to measurement models of higher level constructs (See Figure 4). The practical question is how to transform a sequence of low-level behaviors into indicators that reflect learners' thinking.

Chung et al. (2023) provide a concrete example of this challenge, highlighting why we assert that indicator development is, in fact, algorithm development and coding guided by a learning and measurement perspective. The game targeted computational thinking, and the measurement question was how to evaluate a player's toy design, which is composed of four parts, in a way that accounts for the outcome (whether it satisfies the design requirements or not) and reflects the problem-solving and debugging processes.

In the game, the player is tasked with designing a toy that meets certain specifications. The player is given various toy parts to use during the building phase, and then in the test phase the player can test the toy to see if it meets the requirements. If the toy does not meet the criteria, the player has to adjust the toy's design.

Our approach was to develop a method for comparing the four components of the learner's toy design to reference solutions. This method allows for the computation of several similarity scores for any toy design: a composite score for the overall design and scores for each toy component. We reasoned that the quality of the design is indicative of the learner's problem-solving and debugging process outcomes. Computing overall and component scores for each attempt allows for tracking progress over time.

Appendix B shows a snippet of an indicator design document we developed for the PBS KIDS game *Toy Maker*, detailing how to compute the overall composite score and component scores for each toy part. Appendix B shows that establishing a common vocabulary is the first step. Measurement considerations in light of the game design are a critical next step. The general requirements for the indicators are identified, such as being able to measure changes in players' responses (in our case, determining how close a player's design is to a solution given the game presents a problem-solving task), being able to compare players in a consistent way (given that players may approach the game in different ways), and being able to differentiate players who use different problem-solving strategies as reflected in different but acceptable game designs.

To achieve the measurement requirements, we examined the solution space for the most critical elements (i.e., what constitutes a valid design, what contributes to a valid design, and how we can operationalize the detection of a valid design). Our solution was to establish a set of rules that reflect whether the player satisfied a specific condition for a particular part as well as the parsimony of the solution. The use of component rules allows for flexibility in terms of how the rules can be weighted for scoring purposes (or not) and the ability to describe players' performance for each toy (e.g., reporting which rules were met or not for each toy).

We included Appendix B to provide insight into the actual indicator development process used in a game designed for 6-year-old children. We wanted to emphasize that algorithm development and coding are essential parts of indicator development, which is unavoidable when working with interactive data in digital systems (coding was required to derive indicators for each game example presented in this chapter). The coding level of effort is influenced tremendously by which learner-system interactions are logged, the game's complexity, the extent to which interactions can be evaluated, and the availability of reference structures for comparison purposes.

Addressing Challenge 2

The most effective way to meet challenge 2 is to adopt a measurement perspective centered around learning (Baker, 1997) with a focal point on the relation between fine-grained behavioral interactions and attendant cognition. As an intellectual tool, a measurement perspective naturally leads to two fundamental questions: What is to be measured? How is it to be measured? Indicator development involves, in the end, an algorithm to be coded to operate on fine-grained behavioral data. Thus, reasoning about how a task design shapes a learner's behavioral responses can reveal the likely cognitive demands of a task. Likewise, reasoning about how a user-interface design enables or constrains learners' ability to express their thinking can reveal the evidentiary value of a learner-system interaction.

Addressing the "what-to-measure" and "how-to-measure" questions often results in an iterative process. For example, desiring to measure problem-solving leads to more questions and increasing definitions about the cognitive demands: Problem-solving about what? What types of problem-solving can be expected of learners (e.g., trial-and-error vs. means-ends)? Under what conditions are learners solving problems (e.g., closed-ended or open-ended problems, resource availability, type of feedback), and with what kinds of learners (e.g., degree of prior knowledge)?

A similar definitional process occurs during indicator development. Given the task design, what learning processes are learners likely to use during the task? How are learning processes expressed through the user-interface elements? For example, terms like "performance," "learning," and "proficiency" can be characterized in different ways and are unlikely to be immediately operationalized. Thus, deconstructing these terms into increasingly more precise definitions will help identify learner-system interactions that can satisfy the definition in the context of the task and cognitive demands. The definitional process may also reveal whether the interactions can be evaluated and used directly or need to be transformed, combined, or evaluated in the context in which the action occurred. This degree of detail is necessary because at some point, code will need to be written to transform the raw behavioral data into an indicator value. As was realized over 50 years ago in software engineering, software development projects are more likely to succeed when clear, precise, and complete requirements are documented (Brooks, 1975).

03 Challenge 3: Gathering Validity Evidence

The crux of credible indicators is the third challenge: validity evidence. Challenge 3 is important because it involves a potentially complex set of transformations performed on the learner-system interaction data to derive indicators of learning processes. The process of transitioning from the event layer to the indicator layer in Figure 4 is realized by algorithms and code, and thus, the transformations may not be easy to inspect and evaluate with respect to the relation between learners' behavior and presumed learning processes. Furthermore, the "degrees of freedom" in learner-system interactions are mediated by the design of the task, necessitating careful attention to learners' responses. Learner-system interactions are highly dependent on the design of the software and user interface—the universe of learners' behaviors is defined by the actions allowed by the user interface.

Addressing Challenge 3

AERA et al. (2014) define standards for validity and the various forms of validity evidence (pp. 23). Sireci and Benítez (2023) provide concrete examples of validation and validity evidence in the context of educational testing. When validation concepts are applied to game-based indicators, both qualitative and quantitative methods can be used for evidence gathering. In this section, we present examples drawn from our own work to illustrate the validation process. In general, our objective is to critically evaluate the extent to which the information encoded in the game-based indicators captures the relevant and meaningful aspects of the target constructs. Qualitative strategies include examination of the game design, game mechanics, and gameplay. Quantitative strategies include the examination of bivariate relations between various game-based indicators and external tests targeting the same construct and, most recently, joint validation of game-based indicators against other outcomes.

Qualitative Approaches

Because learner-system responses are highly dependent on the design of the software and user interface, the game design, game mechanics, and gameplay are all examined. For example, the game design and game mechanics undergo a feature analysis as described in the previous section, *Addressing Challenge 2: Feature Analysis*. Additionally, observation of learners' actual gameplay is critical to explaining unusual learner-system interactions as well as uncovering potential issues with the data. Learners are allowed to play as naturally as possible with essentially no help or intervention from the researchers. This type of observation provides information on where in the game players get stuck and what aspect of the game is confusing (e.g., not understanding the goal; incomplete understanding of the game controls and interface elements; unclear, ignored or missed directions, help, hints, and feedback).

Another qualitative approach is what we refer to as "reverse response process validation." By "reverse," we mean developing game-based indicators using extant microgenetic studies (e.g., Metz, 1993; Siegler, 2007) where we assume that the research base, findings, and theory impart validity. Microgenetic analysis densely samples observations of how learners use their knowledge (or not), how they develop and discover strategies (or not), and how learners transition toward mastery within a subject. These observations are of fine-grained, real-time behaviors that likely covary with the unfolding learning process. Microgenetic studies typically have a line of research that includes theoretical frameworks and prior findings.

For example, we developed a game-based indicator algorithm based on Metz's (1993) microgenetic analysis for the PBS KIDS' game *Pan Balance* (https://pbskids.org/sid/games/pan-balance). In her study, Metz examined how preschoolers built and refined their procedural and diagnostic knowledge of weight and the use of a pan balance. Metz identified patterns of misconceptions that accompanied changes in knowledge. One such misconception, called "higher is heavier," occurs when children mistakenly interpret the higher side of the pan as containing the heavier object. We found that children exhibiting this misconception tended to show less raw change from the pretest to the posttest (Chung & Feng, 2024; Redman et al., 2018).

Quantitative Relations to Other Measures

A conventional way to gather quantitative validity evidence is to evaluate the relationships between game-based indicators and externally validated measures, given that both the externally validated measure and the game share the same set of cognitive demands. One important function of the external measure is to serve as a reference measure of knowledge and skills. The use of experimental conditions and subgroups allows testing of game-based indicators to check whether the indicator values reflect expected directions. For example, an essential property of a measure when learning is involved is instructional sensitivity (Baker, 1997). Assuming instruction was effective, the indicator value prior to instruction should be lower compared to the indicator value post-instruction. Similarly, the indicator value should be higher for learners who already possess the target knowledge or skill compared to those who do not possess the target knowledge or skill. If these relations exist with the external measure, then a similar pattern should also exist with the game-based indicators. Such a pattern of results would be strong validity evidence. Table 4 summarizes the different kinds of comparisons that can be done to provide validity evidence.

Table 4. ***Summary of Potential Validity Evidence***

General analysis		Potential validity evidence
1.	Correlation between external pretest scores and GBIs (gameplay on early rounds).	GBIs are sensitive to preexisting skills and knowledge.
2.	Correlation between external posttest scores and GBIs (gameplay on later rounds).	GBIs are sensitive to learned (or existing) skills and knowledge.
3.	Correlation between the gain scores of the external measures (posttest–pretest) and gain scores of the GBIs (later rounds–early rounds).	GBIs are sensitive to the degree of learning of skills and knowledge.
4.	Subgroup analyses: Compare GBIs of players who *learned* to players who did not learn over the course of the intervention. "Learned" is defined as a positive pretest to posttest gain on the external measure.	If the GBIs of players who learned (vs. players who did not learn) show differences in the expected direction (e.g., show more use of productive processes, less use of nonproductive processes, less errors), then this result would suggest that players who learned use more productive processes than players who did not learn.
5.	Subgroup analysis: Compare (early round) GBIs of players who scored high on the *pretest* external assessment to GBIs of players who scored low.	If the GBIs of players who have high pre-existing skills and knowledge (vs. players who have low pre-existing skills and knowledge) show differences in the expected direction (e.g., show more use of productive processes, less use of nonproductive processes, less errors), then this result would suggest that players who have higher skills and knowledge use more productive processes than players with lower skills and knowledge.
6.	Subgroup analysis: Compare (late round) GBIs of players who scored high on the *posttest* assessments to GBIs of players who scored low.	If the GBIs of players who have high skills and knowledge at the end of the game (vs. players who have low skills and knowledge) show differences in the expected direction (e.g., show more use of productive processes, less use of nonproductive processes, less errors), then this result would suggest that players who have higher skills and knowledge use more productive processes than players with lower skills and knowledge.

Note. GBI = game-based indicators.

Chung and Feng (2024) present game-based indicator validity evidence for various games and additional examples exist involving different games and interactive systems, external measures, ages, interventions, and type of process data (e.g., Chung & Baker, 2003; Chung et al., 2002; Choi, Parks et al., 2021; Choi, Suh et al., 2021; Feng, 2019; Feng & Cai, 2024; Kerr, 2014; Kerr & Chung, 2012; Nagashima et al., 2009; Redman, Chung, Feng et al., 2020a; Redman, Chung, Griffin, & Parks, 2020b; Redman et al., 2018, 2021, 2023; Teng & Chung, 2025).

Joint Modeling of Game-Based Indicators and Validation of Multiple Sources of Evidence

Advances in methodology now allow the validation of multiple game-based indicators that collectively describe a single phenomenon of interest. Information encoded in these indicators can be integrated into a larger, reliable system for measuring performance and learning.

Correlational analysis and multiple linear regression, two of the most used analysis techniques (Zhu et al., 2023), fall short when the goal is to analyze multiple indicators targeting the same construct simultaneously, jointly model these indicators with other measures, or examine relationships without aggregating variables to the player level.

Paradigms of latent variable modeling, such as item response or factor models, offer flexible means to accommodate a wide range of analytic decisions (Skrondal & Rabe-Hesketh, 2004). Item response theory, multilevel modeling, and diagnostic classification modeling have been applied to analyzing one or more data sources collected in game-, simulation-, or computerized task-based research (e.g., Choi, Suh et al., 2021; Feng & Cai, 2024; Liu et al., 2018; Reese et al., 2015).

By examining the relationship between a game-based latent factor, measured by a set of game-based indicators, and one or more assessment-based latent factors, measured by sets of items, we can gauge the extent to which learners actions in an interactive system, such as a learning game, correlate with learning outcomes. From a validation perspective, this approach is tantamount to being able to validate multiple game-based indicators against external assessment item responses. Feng and Cai (2024) demonstrated the analytic benefits of jointly modeling diagnostic indicators, derived from gameplay process data for each in-game task, with traditional pretest-posttest item response data collected in game-based evaluation research.

A benefit for learning research is the ability to connect students' interactions—such as patterns of misconceptions—in a low-stakes, game-based setting, with changes or hindered changes in their educational outcomes that are typically valued in higher stakes settings. One implication of being able to validate diagnostic indicators, whether through a model-based approach or others, is the ability to use these indicators to monitor learner-system interactions and provide feedback that is both relevant and timely.

The qualitative and quantitative approaches described yield a range of validity evidence, from response process evidence to statistical evidence (AERA et al., 2014), and support *Principle 3* (assessment design). Collectively, these approaches are intended to identify the underlying reasons driving learners responses and to test for patterns of relations consistent with expectations (Sireci & Benítez, 2023). The evidence collected is useful for adjusting the design of the task, particularly when learners respond to the game in unexpected ways or when the game-based indicators reveal unexpected relationships. Such incongruence, left unaddressed in the game design or unidentified, becomes pernicious at the analysis and interpretation stage. Behavior that may appear productive may actually be due to reasons entirely unrelated to the target knowledge or skill, leading to biased results and improper inferences.

Discussion

One reason for using technology-based tasks for measurement purposes is that with judicious task design, software can be developed to elicit from a learner complex cognitive processes (e.g., problem-solving, reasoning, creativity, self-regulation, adaptivity, metacognition, collaboration) in the context of some content domain (Baker, 1997; Baker et al., 2016) and offer a more scalable option than other modes such as hands-on performance tasks. A second reason is that the task can be instrumented to automatically track both the process a learner uses to complete a task and the performance outcome of the task. These two capabilities enable the development of rich and highly interactive tasks, scalable administration, unobtrusive behavioral observations, and automated scoring of learner processes and task outcomes.

Although the first reason is widely accepted, as evidenced by the inclusion of technology-enabled tasks in large-scale testing programs (e.g., NAEP and PISA), the second reason—the promise of process data—continues to face challenges (Feng & Cai, 2024; Lindner & Greiff, 2023). To fully realize assessment in the service of learning, not only to understand what learners know and can do, but also to measure the learning processes students are using (or not using) and using that information for instructional purposes, the shortcomings surfaced by Lindner and Greiff (2023) and others (e.g., Chung & Feng, 2024) need to be addressed. Advances are required to move the indicator development process from an artisan activity to an engineering process. The opaqueness of the indicator development and the direct impact algorithms and coding have on the indicator's value led Chung and Feng (2024) to assert that advances are needed in three areas: traceability, interpretability, and algorithm generalizability. *Traceability* refers to the ability to trace how the raw interactions are processed and transformed (e.g., filtered, aggregated, recombined) into a quantitative value. Given an algorithm, *interpretability* refers to how one interprets the value produced by an algorithm—the meaning ascribed to the indicator in light of the assumptions, constraints, and transformations encoded in the algorithm. *Algorithm generalizability* refers to how well an algorithm, based on a theory, encodes the rules and conditions described or predicted by the theory. Algorithm generalization occurs by applying the algorithm (with modification to adjust for task-specific surface features) to generate indicators on tasks that may differ in format, mechanics, content, or even learning goals.

The idea of collecting interaction data in digital systems is not new. What is new is that we have conceptualized learner-system interactions as an observation to explicitly support measurement and thus a concomitant focus on validity. Interestingly, a side effect of the challenges in indicator development may be an increased focus on the meaning of learner-system interactions. To develop software, detailed specifications of what to produce is needed. This demand for detail and definitions may increase awareness of how learner-system interactions represent evidence of the target knowledge or learning process.

In this chapter, we have attempted to illustrate how well-designed instructional opportunities in interactive systems provide measurement opportunities. These opportunities can result in what we call measurement without testing: Learner-system interactions that are designed to support students' learning are, by definition observable, and we believe they carry the most relevant information about students' learning. Observing learners' interactions in digital systems, whether games or simulations, is still the only scalable method for observing large numbers of students compared to other forms of observation, such as video recording, audio recording, eye tracking, EEG, fMRI, and physiological and motor monitoring. If we can observe what learners are doing as they do it and accurately determine why, then that capability may help move us toward tailored, adaptive, and individualized learning for all students.

Appendix A:

Learning-Related Features and Definitions

Category Feature	Feature Description
Learning-Goal Alignment	
Learning Goal Aligned with Gameplay	Game requires players to access and use the targeted learning goal(s) in order to play the game. Gameplay is not tangential to the targeted learning goals and the player must understand the targeted learning goals in order to be successful in the game. When the gameplay and learning goals are *not* aligned, a player is able to succeed at the game without demonstrating understanding of the learning goals.
Learning-Goal Explanation	
Explanation Provided	Game provides introduction to or background information for the targeted learning goal(s) before or during gameplay (this information is distinct from feedback). This information *may* be presented in an in-game tutorial, but the presence of a tutorial does not automatically imply an explanation of the target learning goal(s).
Type of Progression	
Fixed Learning-Goal Complexity	Gameplay tasks/rounds/levels do not change in complexity over the course of the game. This does not mean that the tasks are exactly the same over the course of the game, just that they do not change in level of complexity or difficulty.
Increasing Learning-Goal Complexity	Gameplay tasks/rounds/levels presented become more complex or "hard" as the player advances in the game. Learning-goal complexity refers to the targeted learning goals that are present in the game. Learning-goal complexity may increase by way of the inclusion of additional learning goals as the game progresses.
Adaptive Game Progression	The game serves up tasks/rounds/levels based upon player performance. This means that the level order is dependent on player input and will not necessarily be the same for each player.
Fixed Game Mechanic Complexity	Gameplay tasks/rounds/levels do not change in game mechanic or user interface complexity over the course of the game. This does not necessarily mean that tasks are exactly the same in terms of game mechanics over the course of the game, just that they do not change in level of complexity or difficulty. This is entirely independent of learning-goal complexity, which relates to the targeted learning goals.
Gameplay Type	
Judgment/Decision Making	Players are asked to make judgments/decisions based on their understanding of the target learning goal(s).
Input Submission	
Intentional Submission	Players must intentionally submit their answer/response to stimuli in the game. This may take the form of a submit button (like *Pan Balance* or *Meatball Launcher*). The point is that submission must be intentional as some games may allow players to manipulate the game space and automatically accept a correct response (whether the player means to submit it or not). A gamified assessment (in which a player must select a correct response from several items, similar to a multiple choice question) does not count as an intentional submission unless there is a further step to confirm the selected response is the intended answer.
Creative Submission	Game requires players to create something or perform an activity. This is different than a game that requires the player to select a correct object or response.
Instruction and Feedback	
Demo	The game provides a demonstration that explains and shows the gameplay to players by walking them through a task/gameplay. The demo may include directions about the target learning goals and/or gameplay. Most games begin with some sort of background information or gameplay directions; these do not count as a demo unless the player sees a demonstration of the gameplay. The demo cannot be interactive.

Category Feature	Feature Description
Tutorial Level	The game provides a tutorial that requires players to participate in a demonstration of how to play a level/task or how to manipulate specific elements of the game. The tutorial must be interactive (i.e., require player participation or input).
Demo Skip Option	Game allows players to skip the demo, if desired. Presence of a skip button does not necessarily indicate the presence of a demo. Some games allow players to skip the instructions or background story/information. Note: Demo skip option may only be present if a demo exists and is able to be skipped.
Individual Learning-Goal Presentation	If there is more than one learning goal targeted by the game, it presents the learning goals individually (not at the same time).
Modal Feedback	Modal feedback requires players to attend to the feedback while it is being given. Gameplay and game interactions are disabled while feedback is being delivered. This means that players are unable to skip feedback.
Audio-Visual Feedback	Feedback is provided both in audio and visually (text or other visual clue).
Correct Answer Acknowledgment	Feedback or acknowledgment of a correct input is provided without elaboration about why it is correct (e.g., "good job!" or "that's right" or another audio or visual clue that indicates success).
Correct Answer Elaboration	Feedback acknowledges the correct input but ALSO explains why it is correct by elaborating on the target learning goal (e.g., "you are right, that block is taller than the other ones"). Elaboration does not need to occur for each round of feedback, but should be marked if it is present at any point in the game.
Incorrect Answer Acknowledgment	Feedback or acknowledgment of an incorrect input is provided without elaboration about why it is incorrect (e.g., "try again" or "that's not right" or another audio or visual clue that indicates an incorrect input).
Incorrect Answer Elaboration	Feedback acknowledges the incorrect input but ALSO explains why it is incorrect by elaborating on the target learning goal (e.g., "that's not right, that block isn't taller than the other ones"). Elaboration on the game mechanic (e.g., "those dinosaurs are not in the right order") does not count as Incorrect Answer Elaboration. Repetition of the task prompt after signaling an incorrect answer does NOT count as elaboration if it does not also include some explanation of what was incorrect. Elaboration does not need to occur for each round of feedback, but should be marked if it is present at any point in the game.
Graduated Feedback	Feedback becomes progressively more explicit or helpful as more errors are made by the player. This includes removal of incorrect answer options for selected response tasks, hints about the correct answer or how to complete the task, and other means for helping the player successfully advance in the game.
Constructive Processes	
Prediction	Game asks the player to predict outcome(s) based upon given information or game states. Usually prompts will ask, "what will happen next?" "What will happen if…" or ask the player to manipulate variables in the game space to effect a certain outcome. This does not apply to games that just ask a player to select a correct object or response (in the vein of a selected response assessment).
Reflection	Game asks the player to explicitly reflect on their answer or input (e.g., compare it to prediction/hypothesis, think about whether something worked, etc.).
Questioning	Game asks rhetorical questions about the target learning goal(s). This occurs (more often) in exploration games where the questions are rhetorical because the player is not required to answer them as part of gameplay.
Debugging/Correction	Game asks the player to correct or refine input based upon feedback. For example, if the player builds something that is unsuccessful, and the game asks the player to improve it so that it works better, this is debugging/correction. However, the game must present the player with their original creation/submission to fix, and not have them start again/redo their creation/submission.

Appendix B:

Indicator Design Document

Note: This appendix is an excerpt from an indicator design document for a game in a current study. Identifying names of the game has been renamed to generic labels.

Definitions

The following terms are used throughout this documentation. They are used to establish a shared language when we discuss various game-based indicators and the algorithms used to implement the indicators.

- **Toy type:** A term used to refer to one of the three types of toys that players can make in the game. These types are: Toy Type 1, Toy Type 2, and Toy Type 3.
- **Design category (category):** A term used to refer to the part, the color, the sizing, or the power/battery of a toy design.
- **Task (in-game task, game task):** A term used to replace the typically used "game level" to avoid possible confusion with downstream statistical analyses, where the term "level" means something distinctly different from a game level (e.g., in multilevel modeling, a level refers to an aggregation level—item level, student level, classroom level, school level). In the game, a task is the player making a toy. There are a total of nine tasks in the game, three tasks per type of toys—Toy Type 1, Toy Type 2, and Toy Type 3—that players can make.
- **Level:** Not used in this document. In this document, "game level" is referred to as "task" or "in-game task" or "game task."
- **Rule**: The set of conditions that satisfy part of all of the criteria for a given toy, task, and toy component (part, size, color, or power). A task may have multiple rules that if all are satisfied, indicate the player has a solution for the task.
- **Attempt:** The window of gameplay that starts with selecting (or reselecting) parts and modifiers of a toy, confirming the toy design, building the toy, testing the toy, observing the results of the testing, and ends with trying again if the testing fails. Each task can have more than one attempt.
- **Confirming design:** This refers to when a player clicks the right arrow button to confirm their toy design (before building).
- **Testing design:** This refers to when, after the toy is built, a player clicks the test button to test their toy design and observe if the design passes the test by meeting all criteria for a given task.
- **Player's confirmed toy design (player's design):** This refers to a player's confirmed toy design that is then used to build the toy.
- **Task solution:** This refers to a pre-specified compact solution for a task in the game.

Compact Solution Per In-Game Task

Rationale and Context

The goal of having a set of compact solutions is to facilitate analytics. By "compact," we mean the most parsimonious (also see the second section named "properties of a compact solution"). By "facilitate analytics," we mean the following activities, most of which are concerned with the development of indicators that describe and differentiate players' in-game performance or progress:

1. The development of indicators that gauge some kind of changes in player response' quality or closeness to the goal state requires that we know one or more references that could represent the goal state. In other words, convergence [to] or divergence [from], or being productive or unproductive, is always gauged with respect to at least one reference.

2. We need to establish a consistent way for comparing performances between players. A compact solution, specified per task, is one such reference that enables between-player comparisons.

3. We might also be interested in differentiating players who complete the same task but with different strategies, assuming such differences would relate to players' varying degrees of problem-solving or debugging abilities. For example, for the same task, Player A could complete the task with the most compact solution, whereas Player B completes the task with redundant modifiers used.

How exactly we want to score players' performances, such as to what extent we are concerned about a player's design being fully correct or being the most parsimonious, can be decided when we develop the scoring algorithm.

Properties of a Compact Solution

A compact solution has the following properties that are applied to each of the three parts of a toy (e.g., a toy type 3 has three parts: a body, a door, and a decoration):

- There are no "unnecessary but not incorrect" modifiers added. If a part requires no size, color, and/or power modifier, leave the corresponding modifier section blank (e.g., an empty list).
- If no specific part is needed to pass the test, leave the part section blank (e.g., an empty list).
- For example, for one of the solutions of Toy Type 3 Task 1, a player can use any of the house bodies with one orange modifier and two small modifiers, can use any of the doors, and can use any of the decorations. Then for this solution and for each of the three parts, the part name is left blank (e.g., an empty list).
- If there is a specific part needed to pass the test, use the name of the specific part (e.g., a list with only one element, where the element is the part name).
- For example, for one of the solutions of Toy Type 3 Task 1, a player can use the first house body with one orange modifier and one small modifier, provided that the player also uses the fourth decoration, along with any of the doors. Then for this solution and for the body part and the decoration part, we specify the name of the first house body (*fairy*) and the name of the fourth decoration (*flag*).

Generally, we assume that for each part (p = 1, 2, or 3) of a compact task solution, we have specified the following information:

1. Names of the specific parts needed; leave it blank if it does not matter which specific part can be used.
2. Size modifier(s) needed; leave blank if there is no requirement about Part p's size.
3. The color modifier needed; leave blank if there is no requirement about Part p's color.
4. Whether the power modifier needs to be included or explicitly excluded; leave blank if there is no requirement about Part p's power inclusion or exclusion.

Set of Compact Solutions for Each Task

The following section provides details on how various indicators are derived from players' submitted responses, as well as the finalized design score, which will be used as the primary outcome for modeling. The final score incorporates multiple facets of performance, and it is the most sensitive to incremental changes in the levels that the game requires players to beat.

Algorithm: Converging to and Diverging From a Solution

1.1 Sub-construct

Begin to notice where errors exist in algorithms (sequences) and attempt to fix them (debugging) (e.g., a child recognizes that they need to put larger blocks at the bottom of a block tower to keep it from falling).

1.2 Overview

The basic approach to indicator developed for the game is to detect which rules are satisfied. For each toy, rules are formed for the combination of three parts, the color modifier, the size modifier, and the power modifier. Then a player's submitted solution is checked against the solution set, and each rule is evaluated to return true or false. The rules have hierarchy (e.g., Rule 0, Rule 1, Rule 2, or higher) such that the more the higher-level rules are satisfied, the closer the player is to a solution. The use of rules satisfied and unsatisfied (or met and unmet) also provides flexibility in terms of how the rules can be used for scoring purposes. An example is presented at the end to measure converging to or diverging from a solution set.

In the general approach, rules are defined to help determine whether a player's submitted solution attempt is getting closer to meeting the beat-round criteria. The different types of rules are:

- **Rule 0** is used to check if a player has added any unnecessary modifiers.
- **For Rule 1 and above,** the more rules that are satisfied, the closer the performance is to a solution and thus the closer the player is to beating an in-game task (i.e., making a toy that would pass the test). For example, each of the toy types (Toy Type 1, Toy Type 2, and Toy Type 3) in the game has three in-game tasks (Task 1 to Task 3).

1.3 Data Structures

- Set of possible compact solutions for each task of each toy type.
- A player's confirmed toy design (player's design).

1.4 General Approach

- Create a solution set of all possible solutions for each task of all toys (9 total)
  - 3 toy types
  - 3 tasks per toy type
- Given a player's confirmed toy design and a pre-specified task solution
  - Check if the player's design contains the correct part(s)
    - For example, a task in the toy type 3 section has three parts: the house body, the door, and the house decoration
  - Check if the player's design contains the correct color modifier for the right part, and if the correct color modifier is in the last position
  - Check if the player's design contains the correct size modifier(s)
    - Check for the any size modifier
    - Check if the size modifier is added to the right part
    - Check if the overall sizing effect (after executing all size modifiers) is in the correct direction
  - Check if the player's design contains a power modifier if required, or does not contain a power modifier when not required

1.5.1 Part Checking

Because for some tasks in the game there are certain combinations of toy parts that affect task completion, all parts of the player's toy design are evaluated jointly.

Binary Representation for Handling Part Interaction

To account for the interactions between parts, we use a binary representation to record the combination of all three parts in a player's toy design. Each part (Part01, Part02, or Part03) consists of five unique components, resulting in a total of 15 distinct components across the three parts. We represent the presence and absence of these components using a sequence of 15 binary digits (ones and zeros), with each digit corresponding to a specific component, arranged from Part01 to Part03, top to bottom. For each solution, a digit at index i is set to 1 if the solution includes the component at that index, and 0 if the component is not required. We then check if the binary representation of the parts used in a players' response matches any of the binary representations associated with a compact solution. Note that for solutions involving OR relationships, multiple binary representations can be associated with the same solution.

If there is a part requirement:

- Rule 1. Check if the task-required part is the same as the player's selected part; return true if the rule is satisfied and false if not satisfied

If this is no part requirement (when the player can use any part):

- Rule 0. Return true.

1.5.2 Color Checking

If there is a color mod requirement:

- Rule 1. Check if the player added any color modifier when there is a color mod requirement, return true if the rule is satisfied and false if not satisfied.
- Rule 2. Check if the player added any color modifier to the right part, given Rule 1 is true, return true if the rule is satisfied and false if not satisfied.
- Rule 3. Check if the player added the right color modifier, given Rule 1 and Rule 2 are true, return true if the rule is satisfied and false if not satisfied.
- Rule 4. Check if the player added the right color modifier as the last color modifier, given Rules 1–3 are true; return true if the rule is satisfied and false if not satisfied.

If this is no color mod requirement:

- Rule 0. Check if a player added any color modifier; return true if the rule is satisfied and false if not satisfied.

1.5.3 Size Checking

If there is a size mod requirement:

- Rule 1. Check if the player added any size modifier when there is a size mod requirement; return true if the rule is satisfied and false if not satisfied.
- Rule 2. Check if the player added any size modifier to the right part, given Rule 1 is true; return true if the rule is satisfied and false if not satisfied.
- Rule 3. Check if the player overall achieved the same sizing direction (e.g., big or bigger), given Rule 1 and Rule 2 are true;

return true if the rule is satisfied and false if not satisfied. This can be achieved by one of the following:

a. Adding the right number of the right type of size modifier, or

b. Adding modifiers that have the same sizing effect as the solution, but the number of modifiers added differs from the solution, or

c. Adding modifiers such that their sizing effects balance out to be the desired sizing effect (e.g., if a task wanted one small modifier, and the player added two small modifiers and one big modifier, then overall the toy was down-sized once).

If this is no size mod requirement:

- Rule 0. Check if a player added any size modifier; return true if the rule is satisfied and false if not satisfied.

1.5.4 Power Checking

If there is a power mod requirement:

- Rule 1. Check if the player added the power modifier to the right part when the task asks for it, or if the player did not add the power modifier when the task explicitly did not ask for it; return true if the rule is satisfied and false if not satisfied.

If this is no power mod requirement:

- Rule 0. Check if a player added the power modifier; return true if the rule is satisfied and false if not satisfied.

1.6 Player Solution Evaluation

```
Given Task x of Toy Type y
  For each pre-specified solution for Task x
    For each player's confirmed toy design z (Attempt z):
    – Compute the number of rules met for the toy's part, size, color, and power
    – Compute the number of rules unmet for the toy's part, size, color, and power
```

The use of rules allows for flexibility in terms of how we weigh the rules for scoring purposes (or not), being able to describe players' performance by toy (e.g., reporting which rules were met or not for each toy).

The example below shows what is possible once we know which rules are met or unmet. One possible scoring rubric given the list of rules met or not met is presented in Section 2.6.1.

1.6.1 Relationship Between Indicators and Performance

Because Rule 1 through Rule 4 are defined in order of increasing difficulty of being met, meeting both Rule 1 and Rule 2 (or higher) indicates better performance than just meeting Rule 1. For instance, merely adding a color modifier without noticing which part needs the color or that the added color modifier's effect will be overridden by another color modifier added after it would only satisfy Rule 1, not Rule 2 and above.

1.6.2 Scoring Rubric

```
Given Task x of Toy Type y
For each pre-specified Solution s for Task x
  For each player-confirmed toy design:
   – If the task has a category requirement (part, color, size, or power),
     satisfying one rule within that category adds 1 point for that category.
   – For any category that is not required by the task, the rules do not count
     towards scoring. For example, Solution S may not require the player to add a
     color modifier, then all rules specified under Section 1.5.2 do not apply.
   – We assume that Rule 0 does not contribute to the scoring process.
   – For each category (part, color, size, or power), a category-specific score is
     computed.
   – An overall design score is the sum of four category-specific design scores,
     divided by the maximum number of points that can be earned for Solution S.
     – Overall design score = part score + color score + size score + power score
```

The rationale for not counting Rule 0 and the like is as follows. As long as the player does not add any part or modifier that leads them away from a solution, we do not penalize them for adding unnecessary modifiers or parts.

We can use the resulting score and changes in scores to gauge, over multiple attempts by one player, the extent to which the submitted toy designs converge to or diverge from a pre-specified solution, and to identify which specific rules are met or unmet.

References

Ackerman, P. L. (1990). A correlational analysis of skill specificity: Learning, abilities, and individual differences. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 16*(5), 883–901.

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

Anderson, J. R. (1982). Acquisition of cognitive skills. *Psychological Review, 89*(4), 369–406. https://doi.org/10.1037/0033-295X.89.4.369

Bakeman, R., & Gottman, J. M. (1997). *Observing interaction: An introduction to sequential analysis* (2nd ed.). Cambridge University Press.

Baker, E. L. (1997). Model-based performance assessment. *Theory Into Practice*, *36*(4), 247–254. https://doi.org/10.1080/00405849709543775

Baker, E. L. (2012). *Ontology-based educational design: Seeing is believing* (CRESST Resource Paper No. 13). University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing (CRESST).

Baker, E. L. (2015a, April 16–20). Feature analysis as a technology design and evaluation tool. In G. K. W. K. Chung (Chair), *Design issues regarding the use of games and simulations for learning and assessment* [Roundtable]. American Educational Research Association Annual Meeting, Chicago, IL, United States.

Baker, E. L. (2015b, April 16–20). *The design and validity of new assessments: Windows on architecture, art, & archaeology* [Invited speaker session]. American Educational Research Association Annual Meeting, Chicago, IL, United States.

Baker, E. L., Cai, L., Choi, K., & Madni, A. (2015, June 22–25). *Functional validity: Extending the utility of state assessments* [Conference session]. 2015 National Conference on Student Assessment, San Diego, CA, United States.

Baker, E. L., Chung, G. K. W. K., & Cai, L. (2016). Assessment gaze, refraction, and blur: The course of achievement testing in the past 100 years. *Review of Research in Education, 40*, 94–142.

Baker, E. L., Madni, A., Michiuye, J. K., Choi, K., & Cai, L. (2015). *Smarter Balanced Assessment Consortium: Mathematical reasoning project quantitative analyses results: Grades 4, 8, and 11.* Smarter Balanced Assessment Consortium.

Bennett, R. E., Persky, H., Weiss, A. R., & Jenkins, F. (2007). *Problem solving in technology-rich environments: A report from the NAEP Technology-Based Assessment Project* (NCES 2007–466). U.S. Department of Education, National Center for Education Statistics. https://eric.ed.gov/?id=ED497845

Brooks, F. (1975). *The mythical man-month: Essays on software engineering*. Addison-Wesley Publishing Company.

Center for Advanced Technology in Schools, & CRESST (2012). *CATS-developed games* (Resource Paper No. 15). University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing (CRESST).

Center for Advanced Technology in Schools, & CRESST (2013a). *CATS knowledge and item specifications: Rational number equivalence* (Revision 10/25/13). University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing (CRESST).

Center for Advanced Technology in Schools, & CRESST (2013b). Save Patch *tutorial and game level design: RN1.6*. University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing (CRESST).

Choi, K., Parks, C. B., Feng, T., Redman, E. J. K. H., & Chung, G. K. W. K. (2021). *Molly of Denali analytics validation study final report* (Deliverable to PBS KIDS). University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing.

Choi, K., Suh, Y. S., Chung, G. K. W. K., Redman, E. J. K. H., Feng, T., & Parks, C. B. (2021). *A secondary analysis of the Molly of Denali RCT data: Examining the relationship among game-based indicators, video usage, and external outcomes using advanced psychometric modeling and population data* (Deliverable to EDC). University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing.

Chung, G. K. W. K. (2015). Guidelines for the design, implementation, and analysis of game telemetry. In C. S. Loh, Y. Sheng, & D. Ifenthaler (Eds.), *Serious games analytics: Methodologies for performance measurement, assessment, and improvement* (pp. 59–79). Springer.

Chung, G. K. W. K., & Baker, E. L. (2003). An exploratory study to examine the feasibility of measuring problem-solving processes using a click-through interface. *Journal of Technology, Learning, and Assessment, 2*(2). http://ejournals.bc.edu/ojs/index.php/jtla/article/view/1662

Chung, G. K. W. K., de Vries, L. F., Cheak, A. M., Stevens, R. H., & Bewley, W. L. (2002). Cognitive process validation of an online problem solving assessment. *Computers in Human Behavior*, *18*(6), 669–684.

Chung, G. K. W. K., & Feng, T. (2024). From clicks to constructs: An examination of validity evidence of game-based indicators derived from theory. In M. Sahin & D. Ifenthaler (Eds.), *Assessment analytics in education* (pp. 327–354). Springer International Publishing. https://doi.org/10.1007/978–3-031–56365–2_17

Chung, G. K. W. K., O'Neil, H. F., & Herl, H. E. (1999). The use of computer-based collaborative knowledge mapping to measure team processes and team outcomes. *Computers in Human Behavior*, *15*(3–4), 463–494. https://doi.org/10.1016/S0747-5632(99)00032-1

Chung, G. K. W. K., & Parks, C. (2015). *Feature analysis validity report* (Deliverable to PBS KIDS). University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing.

Chung, G. K. W. K., Redman, E. J. K. H., & Choi, K. (2023). *Wombats analytics evaluation–Final plan* (Deliverable to PBS KIDS). University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing.

Chung, G. K. W. K., & Roberts, J. (2018, April 13–17). Common learning analytics for learning games. In E. L. Baker (Chair), *Games and simulations: Learning analytics and metrics* [Symposium]. American Educational Research Association Annual Meeting, New York, NY, United States.

Chung, G. K. W. K., Ruan, Z., & Redman, E. J. K. H. (2021, April 9–12). *A qualitative comparison of young children's performance on analogous digital and hands-on tasks: Assessment implications* [Paper presentation]. American Educational Research Association Annual Meeting, Virtual Conference, United States.

Cronbach, L. J., & Meehl, P. E. (1955). Construct validity in psychological tests. *Psychological Bulletin, 52*(4), 281–302. https://doi.org/10.1037/h0040957

Domagk, S., Schwartz, R. N., & Plass, J. L. (2010). Interactivity in multimedia learning: An integrated model. *Computers in Human Behavior*, *26*(5), 1024–1033. https://doi.org/10.1016/j.chb.2010.03.003

Feng, T. (2019, April 5–9). *Using game-based measures to assess children's scientific thinking about force* [Poster presentation]. American Educational Research Association Annual Meeting, Toronto, Canada.

Feng, T., & Cai, L. (2024). Sensemaking of process data from evaluation studies of educational games: An application of cross-classified item response theory modeling. *Journal of Educational Measurement*, 12396. https://doi.org/10.1111/jedm.12396

Fitts, P. M., & Posner, M. I. (1967). *Human performance*. Brooks/Cole.

Foster, N., & Piacentini, M. (Eds.). (2023). *Innovating assessments to measure and support complex skills*. OECD. https://doi.org/10.1787/e5f3e341-en

Gordon, E. W. (1970). Toward a qualitative approach to assessment. *Report of the Commission on Tests, II. Briefs* (pp. 42–46). College Entrance Examination Board.

Gottman, J. M., & Notarius, C. I. (2000). Decade review: Observing marital interaction. *Journal of Marriage and Family*, *62*(4), 927–947. https://doi.org/10.1111/j.1741-3737.2000.00927.x

Greer, R. D., & McDonough, S. H. (1999). Is the learn unit a fundamental measure of pedagogy? *The Behavior Analyst*, *22*(1), 5–16. https://doi.org/10.1007/BF03391973

Hao, J., Smith, L., Mislevy, R., von Davier, A., & Bauer, M. (2016). Taming log files from game/simulation-based assessments: Data models and data analysis tools. *ETS Research Report Series*, *2016*(1), 1–17. https://doi.org/10.1002/ets2.12096

Heitz, R. P. (2014). The speed-accuracy trade-off: History, physiology, methodology, and behavior. *Frontiers in Neuroscience*, *8*. https://www.frontiersin.org/articles/10.3389/fnins.2014.00150

Janlert, L.-E., & Stolterman, E. (2017). *Things that keep us busy: The elements of interaction*. MIT Press. https://doi.org/10.7551/mitpress/11082.001.0001

Jiao, H., He, Q., & Veldkamp, B. P. (2021). Editorial: Process data in educational and psychological measurement. *Frontiers in Psychology*, *12*. https://www.frontiersin.org/articles/10.3389/fpsyg.2021.793399

Kennedy, G. E. (2004). Promoting cognition in multimedia interactivity research. *Journal of Interactive Learning Research*, *15*(1), 43–61. https://www.proquest.com/docview/1468384849/citation/131F6A71935242F4PQ/1

Kerr, D. S. (2014). *Into the black box: Using data mining of in-game actions to draw inferences from educational technology about students' math knowledge* [Unpublished dissertation, ProQuest No. 3613716]. University of California, Los Angeles.

Kerr, D., & Chung, G. K. W. K. (2012). Identifying key features of student performance in educational video games and simulations through cluster analysis. *Journal of Educational Data Mining, 4*(1), 144–182.

Lindner, M. A., & Greiff, S. (2023). Process data in computer-based assessment: Challenges and opportunities in opening the black box. *European Journal of Psychological Assessment, 39*(4), 241–251. https://doi.org/10.1027/1015-5759/a000790

Liu, H., Liu, Y., & Li, M. (2018). Analysis of process data of PISA 2012 computer-based problem solving: Application of the modified multilevel mixture IRT model. *Frontiers in Psychology, 9*(1372).

Messick, S. (1995). Validity of psychological assessment: Validation of inferences from persons' responses and performances as scientific inquiry into score meaning. *American Psychologist, 50*(9), 741–749. https://doi.org/10.1037/0003-066X.50.9.741

Metz, K. E. (1993). Preschoolers' developing knowledge of the pan balance: From new representation to transformed problem solving. *Cognition and Instruction, 11*(1), 31–93. https://doi.org/10.1207/s1532690xci1101_2

Mislevy, R. J., Riconscente, M., & Corrigan, S. (2015). Evidence-centered assessment design. In S. Lane, M. R. Raymond, & T. M. Haladyna (Eds.), *Handbook of test development* (2nd ed., pp. 40–63). Routledge. https://doi.org/10.4324/9780203102961

Nagashima, S. O., Chung, G. K. W. K., Espinosa, P. D., & Berka, C. (2009). Sensor-based assessment of basic rifle marksmanship. *Proceedings of the I/ITSEC*, Orlando, FL.

National Center for Education Statistics. (2012). *The nation's report card: Science in action: Hands-on and interactive computer tasks from the 2009 science assessment* (Report No. NCES 2012–468). Institute of Education Sciences, U.S. Department of Education. https://nces.ed.gov/nationsreportcard/pdf/main2009/2012468.pdf

National Center for Education Statistics. (2020). *2017 NAEP transition to digitally based assessments in mathematics and reading at grades 4 and 8: Mode evaluation study* [White Paper]. Institute of Education Sciences, U.S. Department of Education. https://nces.ed.gov/nationsreportcard/subject/publications/main2020/pdf/transitional_whitepaper.pdf

National Governors Association Center for Best Practices & Council of Chief State School Officers. (2010). *Common Core State Standards for Mathematics*.

National Mathematics Advisory Panel. (2008). *Foundations for success: The final report of the National Mathematics Advisory Panel*. U.S. Department of Education.

O'Neil, H. F., Jr., Chung, G. K. W. K., & Brown, R. (1997). Use of networked simulations as a context to measure team competencies. In H. F. O'Neil Jr. (Ed.), *Workforce readiness: Competencies and assessment* (pp. 411–452). Erlbaum.

Organisation for Economic Co-operation and Development (OECD). (2014). *PISA 2012 Results: Creative problem solving: Students' skills in tackling real-life problems* (Volume V). OECD Publishing. http://dx.doi.org/10.1787/9789264208070-en

Organisation for Economic Co-operation and Development (OECD). (2021). *OECD digital education outlook 2021: Pushing the frontiers with artificial intelligence, blockchain and robots*. https://doi.org/10.1787/589b283f-en

Organisation for Economic Co-operation and Development (OECD). (2023). *PISA 2025 Learning in the digital world framework (second draft)*. OECD Publishing. https://www.oecd.org/media/oecdorg/satellitesites/pisa/PISA%202025%20Learning%20in%20the%20Digital%20World%20Assessment%20Framework%20-%20Second%20Draft.pdf

Ostrov, J. M., & Hart, E. J. (2013). Observational methods. In T. D. Little (Ed.), *The Oxford handbook of quantitative methods in psychology* (Vol. 1, pp. 285–303). Oxford University Press.

Plass, J. L., Schwartz, R. N., & Heidig, S. (2012). Interactivity in multimedia learning. In N. M. Seel (Ed.), *Encyclopedia of the sciences of learning* (pp. 1615–1617). Springer US. https://doi.org/10.1007/978-1-4419-1428-6_1848

Redman, E. J. K. H., Chung, G. K. W. K., Feng, T., Parks, C. B., Schenke, K., Michiuye, J. K., Choi, K., Ziyue, R., & Wu, Z. (2020a). *Cat in the Hat Builds That analytics validation study—Final deliverable* (Deliverable to PBS KIDS). University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing.

Redman, E. J. K. H., Chung, G. K. W. K., Griffin, N., & Parks, C. B. (2020b). *Social-emotional learning games analytics validation study design* (Final deliverable to PBS KIDS). University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing.

Redman, E. J. K. H., Chung, G. K. W. K., Schenke, K., Maierhofer, T., Parks, C. B., Chang, S. M., Feng, T., Riveroll, C. S., & Michiuye, J. K. (2018). *Connected learning final report*. (Deliverable to PBS KIDS). University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing.

Redman, E. J. K. H., Chung, G. K. W. K., Feng, T., Schenke, K., Parks, C. B., Michiuye, J. K., Chang, S. M., & Roberts, J. D. (2021). Adaptation evidence from a digital physics game. In E. L. Baker, R. S. Perez, & S. E. Watson (Eds.), *Using cognitive and affective metrics in educational simulations and games: Applications in school and workplace contexts* (pp. 55–81). Routledge. https://doi.org/10.4324/9780429282201

Redman, E. J. K. H., Feng, T., Parks, C. B., Choi, K., & Chung, G. K. W. K. (2023). *Learning-related analytics KPI—KPI final repor*t (Deliverable to PBS KIDS). University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing.

Redman, E. J. K. H., & Kennedy, A. A. U. (2017). *Feature analysis framework for Measure Up* (Deliverable to PBS KIDS). University of California, Los Angeles, National Center for Research on Evaluation, Standards, and Student Testing.

Reese, D. D., Tabachnick, B. G., & Kosko, R. E. (2015). Video game learning dynamics: Actionable measures of multi-dimensional learning trajectories. *British Journal of Educational Technology, 46*(1), 98–122.

Roberts, J. D., Chung, G. K. W. K., & Parks, C. B. (2016). Supporting children's progress through the PBS KIDS learning analytics platform. *Journal of Children and Media, 10*(2), 257–266.

Siegler, R. S. (2007). Microgenetic analyses of learning. In D. Kuhn & R. S. Siegler (Eds.), *Handbook of child psychology: Vol. 2. Cognition, perception, and language* (6th ed., pp. 464–510). Wiley.

Sireci, S., & Benítez, I. (2023). Evidence for test validation: A guide for practitioners. *Psicothema*, *35*(3), 217–226. https://doi.org/10.7334/psicothema2022.477

Skrondal, A., & Rabe-Hesketh, S. (2004). *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models*. Chapman and Hall/CRC.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, *20*(4), 345–354.

Tatsuoka, K. K., Corter, J. E., & Tatsuoka, C. (2004). Patterns of diagnosed mathematical content and process skills in TIMSS-R across a sample of 20 countries. *American Educational Research Journal, 41*(4), 901–926.

Teng, K., & Chung, G. K. W. K. (2025). Measuring children's computational thinking and problem-solving in a block-based programming game. *Education Sciences, 15*(1), 51. https://doi.org/10.3390/educsci15010051

U.S. Department of Education, Institute of Education Sciences, What Works Clearinghouse. (2015, November). *WWC review of the report: The effects of math video games on learning: A randomized evaluation study with innovative impact estimation techniques*. http://whatworks.ed.gov

Vendlinski, T. P., Delacruz, G. C., Buschang, R. E., Chung, G. K. W. K., & Baker, E. L. (2010). *Developing high-quality assessments that align with instructional video games* (CRESST Report 774). National Center for Research on Evaluation, Standards, and Student Testing (CRESST), University of California, Los Angeles. http://files.eric.ed.gov/fulltext/ED512655.pdf.

Webb, N. M. (1983). Predicting learning from student interaction: Defining the interaction variables. *Educational Psychologist, 18*(1), 33–41. https://doi.org/10.1080/00461528309529259

Williams, M. D., & Dodge, B. J. (1993). *Tracking and analyzing learner-computer interaction*. Proceedings of Selected Research and Development Presentations at the Convention of the Association for Communications and Technology. https://eric.ed.gov/?id=ED362212

Young, M. F., Kulikowich, J. M., & Barab, S. A. (1997). The unit of analysis for situated assessment. *Instructional Science*, *25*(2), 133–150. https://doi.org/10.1023/A:1002971532689

Zhu, S., Guo, Q., & Yang, H. H. (2023). Beyond the traditional: A systematic review of digital game-based assessment for students' knowledge, skills, and affections. *Sustainability*, *15*(5), Article 4693. https://doi.org/10.3390/su15054693

Zumbo, B. D., Maddox, B., & Care, N. M. (2023). Process and product in computer-based assessments: Clearing the ground for a holistic validity framework. *European Journal of Psychological Assessment*, *39*(4), 252–262. https://doi.org/10.1027/1015-5759/a000748

Study Group

About the author

Gregory K. W. K. Chung, Ph.D. is the Associate Director for Technology and Research Innovation. Dr. Chung has extensive experience with the use of technology for learning and assessment. He has led projects related to game-based learning or game-based assessments involving pre-school students to adults in formal and informal settings with a focus on STEM topics (e.g., math, physics, engineering, programming) as well as social-emotional learning. His research involves small-scale exploratory studies to multi-district, multi-state RCT. He has conducted instructional technology R&D for IES, NSF, Office of Naval Research, PBS KIDS, Bill and Melinda Gates Foundation, Caplan Foundation for Early Childhood, and numerous other foundations and commercial entities.

Tianying (Teanna) Feng has been appointed Assistant Professor in the Division of Games at the University of Utah. She joins the Division from UCLA, where she is completing her Ph.D. in the Education–Advanced Quantitative Methods program. At UCLA, she serves as a research assistant at the National Center for Research on Evaluation, Standards, and Student Testing (CRESST), SEIS Building, Los Angeles, CA 90095-1522; tfeng0315@ucla.edu. Her primary research interests include technology-based measurement and learning, psychometrics, process modeling, and statistical computing.

Dr. Elizabeth J. K. H. Redman is a Research Scientist specializing in technology and assessment at the National Center for Research in Evaluation, Standards, and Student Testing (CRESST). Her primary research interests include STEM education, educational games, and assessment design. Her recent research focus has been on incorporating assessment capabilities into educational games, including SEL and STEM games. She has experience running observational classroom studies, RCTs and evaluations of educational games.

About the Study Group

The Study Group exists to advance the best of artificial intelligence, assessment, and data practice, technology, and policy; uncover future design needs and opportunities for educational systems; and generate recommendations to better meet the needs of students, families, and educators.

Date of Publication: May 2026

Citation: Chung, G. K. W. K., Feng, T., & Redman, E. J. K. H. (2025). Using learner-system interactions as evidence of student learning and performance: Validity issues, examples, and challenges. In E. M. Tucker, E. Armour-Thomas, & E. W. Gordon (Eds.), Handbook for assessment in the service of learning, Volume I: Foundations for assessment in the service of learning. University of Massachusetts Amherst Libraries.

Author Note:

Gregory K. W. K. Chung, ORCID: https://orcid.org/0000-0003-4380-5661
Tianying Feng, ORCID: https://orcid.org/0000-0003-2215-9234
Elizabeth J. K. H. Redman, ORCID: https://orcid.org/0000-0002-5301-3716

We have no conflicts of interest to disclose. Correspondence concerning this chapter should be addressed to Greg Chung, 300 Charles E. Young Drive North, SE&IS Building, Room 300, Box 951522, Los Angeles, CA 90095–1522.
Email: greg@ucla.edu

The research reported in this chapter was supported by grants from the U.S. Department of Education's Ready to Learn program, the Institute of Education Sciences, and the National Science Foundation. However, research and findings do not necessarily represent the policy of the Department of Education, and you should not assume endorsement by the federal government. [PR/Award No. U295A150003, S368A150011, R305A190433, R305C080015, 2119818].