Research and Practice in Technology Enhanced Learning
Vol. 6, No. 3 (2011) 161 – 185

SUBSTANTIVE VALIDITY OF A SIMULATION-BASED GAME

KRISTEN E. DICERBO[†]

Independent Researcher,11228 W. Roanoke Ave.
Avondale, AZ 85392 USA
kdicerbo@cox.net

DENNIS C. FREZZO

Cisco,325 E. Tasman Dr.
San Jose, CA 95134, USA
dfrezzo@cisco.com

TONY DENG

Cisco,325 E. Tasman Dr.
San Jose, CA 95134, USA
tdeng@cisco.com

Games are seen as attractive potential learning tools because of their ability to engage players and immerse them in situations that invite application of content and skills. However, it is not always clear whether educational games encourage players to utilize particular cognitive processes, access appropriate prior knowledge concepts, and/or apply given procedural skills as intended. Research of substantive validity examines evidence of the cognitive processes students use to complete tasks. This paper examines evidence for the substantive validity of a simulation-based game using recorded sessions of eleven pairs of students (six beginning and five advanced) playing the game. Evidence for the use of troubleshooting skills, prior knowledge, and procedural skills was analyzed. Cognitive processes related to troubleshooting were revealed and differences in the processes of beginning and advanced students were suggested by the data. These findings are discussed in the larger context making inferences about student performance using educational games.

Keywords: Educational games, validity, problem solving.

1. Research Overview

Educational gaming is no longer a new innovation, but the search for engaging, motivating games that are valuable in the classroom continues. Games are seen as attractive potential learning tools because they engage and immerse players in ways that traditional school does not, providing the context needed to encourage application of content (Gee, 2003; Shaffer, 2006; Squire, 2006). Authors of well-designed games claim they can provide problems, tools, experiences, perspectives, and consequences that enable learners to develop rich content understanding (Barab *et al.*, 2009). Numerous

[†]Kristen DiCerbo, 11228 W. Roanoke Ave., Avondale, AZ 85392 USA

examples exist of educational games designed to teach specific content currently implemented in classrooms (see Clark *et al.* (2009) for a review).

However, it is not always clear exactly what skills games elicit from players. While the capability of games to motivate is fairly well documented (Cordova & Lepper, 1996), their ability to encourage players to utilize particular cognitive processes, access particular prior knowledge concepts, and/or apply given procedural skills to solve problems is less clear. In the world of assessment, this question of whether a task is completed using the processes the authors intended is the essence of substantive validity. For example, a student who completes a math game by using algebraic processes to find the answers to the problems is using the intended processes while the student who uses brute force trial and error to find the solution is not. If the students using brute force are lucky, or if there are game clues that hint at the right answers, they may end up with game outcomes similar to those who have used math processes to solve the problems. When we make inferences about students' abilities from the outcomes of the game, we need to ensure we understand all the processes that can lead to those outcomes.

This paper uses a validity framework to examine the evidence that a computer networking game requires troubleshooting skills and domain knowledge that it is designed to elicit. A think-aloud method was judged to be the best way to capture players' thoughts while playing, and results from these efforts were analyzed for evidence of the use of problem solving steps, domain-specific knowledge, and procedural skills. This paper contributes to the literature because it 1) analyzes an in-production, educational game that provides practice with job-related skills, 2) examines the use of skills and knowledge in game play rather than just motivation, and 3) suggests the lens of substantive validity as a means to investigate whether success in a game requires use of targeted knowledge and skills.

2. Literature Review

2.1. *Problem solving in a technical domain*

At the same time that games are coming into favor, there is also increasing interest in teaching and assessing problem solving. Employers rate problem solving skills among the top five most important applied skills for employees at all levels (The Conference Board, 2006). There is great interest in the teaching and assessment of problem solving as an exemplar of 21st century skills (Partnership for 21st Century Skills, 2009). One of the earliest and simplest attempts to define problem solving was Newell and Simon's (1972) list of three steps: orientate, solve, evaluate. While there have been many other definitions, most increasing the number of steps involved, they seem to be able to be reduced to these three steps. One commonly-agreed upon change in the definition of problem solving is that it is now usually described as a cycle (Bransford & Stein, 1993; Hayes, 1989).

Troubleshooting is a common problem solving activity defined as finding a fault in a system and making the necessary changes to restore the system to functioning (Jonassen

& Hung, 2006). Although commonly associated with solving technical problems, troubleshooting can be applied to any situation in which the problem solver works from a broken system to restore functioning, including areas like customer service and business consulting. Experienced troubleshooters recognize the pattern of symptoms associated with different problems, which enables them to rapidly generate and implement solutions (Besnard & Bastien-Toniazzo, 1999).

Jonassen and Hung (2006) propose a cognitive model of troubleshooting that includes:

- Constructing the problem space, or mental representation of the task environment
- Identifying fault symptoms
- Diagnosing faults
- Generating and verifying solutions
- Remembering experience

Their evidence suggests that more experienced troubleshooters develop historical knowledge of problems solved, which they rely on at each stage. In addition, problem solving and troubleshooting are not domain-free activities; deep content knowledge is required to be successful (Willingham, 2007). Games may provide one mechanism by which to assess students' troubleshooting skills, along with their content knowledge in the context of real world problems.

2.2. *Making inferences about students from game play*

Most games, through their scoring mechanisms and in-game events, provide feedback about game play in a way that players easily digest. They also have the potential to provide feedback to students about their knowledge, skills, and abilities, a task traditionally associated with assessment. In fact, games in general, and simulation-based games in particular, contain many parallels to assessment (Behrens *et al.*, 2007). Both games and assessments have the purpose of describing knowledge and skills in a quantifiable manner. Rules define what information is available and constraints around solution paths. The Evidence-Centered Design four process model (ECD; Mislevy *et al.*, 2003) that describes activity selection, presentation, response processing, and evidence accumulation in assessment can also be applied to simulation game scenarios. Both assessment and game authors desire to create models of student (player) behavior and knowledge, and often use similar tools (e.g. Bayesian inference networks) to do so.

A major advantage of game environments as a way to gather information about student abilities is that they provide context for the activities. Students playing "epistemic games" tackle challenges in simulated environments that reflect the actual challenges they would confront in the domain, and can be encouraged through game design to draw on the same kinds of language, tools, and interactions that professionals use (Shaffer, 2006). The authentic context engages students in the task, likely providing an increase in students' motivation to perform well. Research (Schmit & Ryan, 1992; Sundre & Wise, 2003) has suggested that when students' motivation increases, so does the validity of the test. That is, as students become more motivated, their scores become a more accurate reflection of their skills and abilities.

Games can also move us away from having set times and activities called "tests" that interrupt the flow of instruction and learning. Rather, games are a part of the everyday environment of students and playing them is a more natural activity than the artificial testing environment. Shute, Ventura, Bauer, and Zapata-Rivera (2009) discuss the potential for embedded formative assessment within games. That is, they advocate for the use of unobtrusive measures of performance gathered while students maintain flow in the game to provide direct feedback on personal progress and/or modify the learning environment for the player. They introduce the term "stealth assessment" (or what we call ubiquitous, unobtrusive assessment; Behrens *et al*., 2007) to describe embedded assessments so closely tied in to the environment that they are invisible.

This argument for stealth or unobtrusive assessment does not imply that games should be viewed solely as a means of assessment. Rather, any good learning environment needs to be able to gather information about students' levels of knowledge and skill and use that information to provide feedback and ultimately improve subsequent learning (Park & Lee, 2003; Shute *et al.*, 2000). By harnessing this information, we can make inferences about students. However, we need to ensure that the learning environment, in this case the game, is accurately gathering information about the correct skills.

2.3. *Validity*

Before games can be used to make inferences about students' knowledge, skills, and abilities, we must establish that they require the use of these constructs and accurately identify evidence about them. This is essentially a validity question, asking whether we can gather relevant information from a game to make inferences about skills of interest. Messick (1995) proposes six general standards for evidence of validity: 1) content relevance and representativeness; 2) substantive theories and process models; 3) structural evidence that responses are internally consistent across different parts of the test; 4) external evidence that scores relate to other measures or background variables; 5) generalizable, both within and across populations, settings, and time; and 6) consequential, considering the applications, both intended and unintended of scores and their interpretations.

This paper will focus particularly on the substantive aspect of validity. Shepard (1993) argues that it is important to determine which validity questions are essential to support the use of a test, so research in those areas can be prioritized. Substantive validity is particularly important for games to be used for learning because responses in a game are different than traditional assessment responses, so the processes by which they are made require investigation. Questions about response process ask whether examinees used methods related to the construct of interest to complete a task. That is, do the processes used by examinees to complete the task align to what we want to measure? On a multiple choice exam, we want to know that students are using the correct process to arrive at an answer, rather than using extraneous cues in the responses, for example, to select the correct answer. Hickey *et al.* (2000) examined substantive validity of an assessment for

an online system for genetics learning by observing students thinking aloud while completing the problems. They found evidence of students using cues from the question and from previous questions to answer more difficult items, demonstrating that students got correct answers without requisite knowledge, which they call construct-irrelevant easiness. In a computer networking game, we want to know, for example, whether students used game cues that were irrelevant to their networking knowledge to solve tasks.

Validity is about determining whether the interpretations, decisions, and actions made based on data are justified (Moss *et al.*, 2006). The decisions about what knowledge and skills are the ones of interest are based on the interpretations and decisions that people desire to make from the data. If educators want to be able to make decisions about whether students have mastered single digit addition based on their performance, then we want to ensure that students use single digit addition to solve problems in the game. The end goal is to be able to say that students arrived at the correct answer by using the knowledge, skills, and abilities/processes we were interested in assessing. Therefore, we must use research methodologies that allow us to observe both players' actions and their mental processes.

Think-alouds, in which participants say out loud what they are thinking while solving a problem, are a common method by which to gain access to participants' thoughts and ideas (van Someren *et al.*, 1994). Ericsson and Fox (2011) note that the goal of think-alouds is not introspection, but focusing on a challenging task and verbalizing thoughts those come into attention. In a meta-analysis, they found that think-alouds did not result in changes in performance. They have been used successfully, and shown to be more effective than other methods, for observing cognitive processes in, for example, mathematics (Blackwell *et al.*, 1985), second language acquisition (Leow & Morgan-Short, 2004), and usability testing of new technology (Virzi *et al.*, 1993).

3. Method

3.1. *Participants*

Eleven pairs of students (22 students total) participated in the study. Students were recruited through their instructors, who were contacted based on their previous participation in research or attendance at instructor professional development sessions about the game. Instructors recommended pairs of students, so each pair was in the same class and knew their partner previously.

All of the students were taking classes associated with the Cisco Networking Academy (see http://cisco.com/go/netacad/), a public-private partnership between Cisco and over 9,000 educational institutions in over 160 countries. Cisco provides partnering schools with free on-line curriculum and on-line assessments to support local school instructors in teaching ICT skills in areas related to PC repair and maintenance, as well as computer and data network design, configuration, and maintenance in alignment with entry-level industry certifications.

There are two curricula that teach basic networking skills and prepare students for the Cisco Certified Network Associate (CCNA) exam. The two curricula, labeled Discovery and Exploration, each consist of a four course sequence. In Discovery, students are taught with a spiraling method in which each course builds on the content of the course before it and emphasizes early hands-on practice. In the Exploration sequence, course content is divided based on technology (e.g. there is a routing course and a switching course) and emphasizes the theoretical grounding of what is taught. In this sample, six student pairs were in the first course of either the Discovery or Exploration curriculum. These students were referred to as "beginning" students. Five student pairs were in the second course or higher of the curricula, and were expected to have mastered the content of the portion of the game examined here, so they were referred to as "advanced" students. Their demographic characteristics are summarized in Table 1. There were variations in country, education level, and gender of the pairs. While level of class (beginning vs. advanced) was an experimental variable, none of the other variables was used in the analysis of this study. Rather, an attempt was made to sample a broad cross-section of students in order to increase the generalizability of the results.

Pairs 1 and 2 were in the same Discovery class, pairs 8 and 9 were in the same Exploration course 1, and pairs 7 and 10 were in the same Exploration course 4. Instructors indicated the participating students ranged from average to excellent in their computer networking ability. Members of pairs 1, 2, 6, and 11 indicated they spent more than five hours per week playing digital games. Members of pair 4 indicated they rarely played digital games, and members of other pairs ranged between these two extremes. All students were asked to play the game used in this study through the first three tasks (contracts) prior to their session to familiarize themselves with game play.

Table 1. Sample characteristics.

ID	Education level of institution	Course[a]	Country	Gender
P1	Continuing education	D1	US	Male/Male
P2	Continuing education	D1	US	Male/Male
P3	High school	D3	US	Male/Male
P4	High school	D1	US	Male/Male
P5	University	E4	Italy	Female/Male
P6	University	E1	UK	Male/Male
P7	University	E4	US	Female/Male
P8	University	E1	US	Male/Male
P9	University	E1	US	Male/Male
P10	University	E4	US	Male/Male
P11	Community college	E2	US	Male/Male

[a]D1 = Discovery course 1, D3 = Discovery course 3, E1 = Exploration course 1, E2 = Exploaration course 2 (Routing), E4 = Exploration course 4.

3.2. *Materials*

The game used in the study is a simulation-based game using networking and entrepreneurial skills called Aspire. Aspire is closest in genre to strategic simulation and quest games. The main idea of Aspire is that students are entrepreneurs, starting their own small networking companies, and must make both business and technical decisions in the game. Aspire consists of a 2 1/2-D interface that allows navigation, interaction with characters in the game, decision making and interaction (sometimes in the form of multiple choice questions) and complex scenarios that combine numerous networking task requirements. Players are offered up to 24 contracts or technical networking challenges to complete in different venues across a city (see Figure 1). Elements that make this a game rather than a simulation include: the overarching story of building a business, a points scheme in which players earn points on three business and three technical dimensions, supplemental badges which players earn by completing additional tasks, and a controlled and increasing level of challenge. The game is a stand-alone desktop application intended for individuals or pairs; local and global leaderboards are in development. The game is currently aimed at first and second semester students in the CCNA Discovery courses.

The simulation engine behind the game is called Packet Tracer (PT; Frezzo *et al.*, 2010). PT is a domain specific data network simulator used in Networking Academy curricula and performance-based assessments that provides instructional direction, practical experience and assessment-based feedback throughout the courses. PT is a comprehensive simulation, visualization, collaboration, and micro-world authoring tool for teaching networking concepts distributed free to hundreds of thousands of

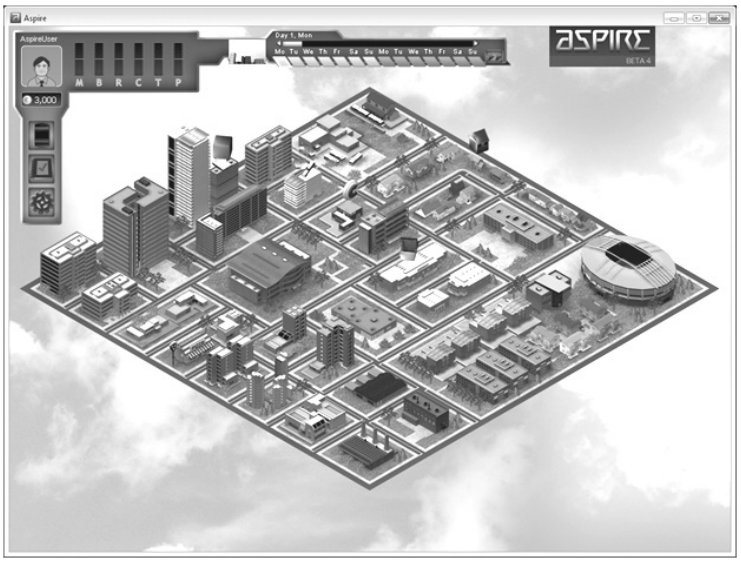

Figure 1. Screenshot of city view in Aspire game.

Networking Academy students. The Aspire game interface is integrated with the Packet Tracer software (as seen at the bottom and right in Figure 2) which renders and simulates the computer and networking devices and systems and provides the ECD-based scoring architecture. Of particular relevance to the notion of substantive validity in this paper is that the Packet Tracer microworld supports a wide variety of networking devices, protocols, and interactions, giving the student ample opportunity for misconcepetions, breakdowns, and sub-optimal troubleshooting in addition to preferred approaches.

3.3. *Procedure*

This study used a paired think-aloud technique (van Someren *et al.*, 1994) to elicit students' thoughts and ideas as they played the Aspire game. Each pair played together on one computer and was instructed to discuss their game actions out loud with each other as they played. The think-aloud procedure was chosen because it provides a means by which to observe otherwise invisible players' cognition without influencing them in the act of observation (Ericsson & Fox, 2011). Solely observing game actions or gathering data from post-tests would not have been sufficient to access the thought processes of the players.

For this study, recording for 10 of the 11 pairs was done over the Internet using Cisco Webex to capture the students' screens and their verbalizations. The interactions of pair

Figure 2. Screenshot of players using ping command and protocol data units for testing.

11 were recorded via ScreenFlow which captured their screen, verbalizations, and video of the players.

3.3.1. *Game play procedure*

Students were given a game file with the first contract played through so they all started at the same place with the game choices made (however, due to technical difficulties, pairs 4, 5, 7 and 8 started from the beginning of the game). Each pair played for 90 minutes.

This paper focuses on an analysis of one contract. The contract was offered by a character named Dr. Evans and involved work in a medical office. The medical office had a network with four computers connected to a Linksys router. Players were told that three of the computers cannot connect to the Internet, while one can, and that their task was to get them all connected. Players were then given instructions and hints for diagnosing the problem, fixing the connections, and testing the solution, as follows:

> "Correct the settings on the non-working PCs so that they can connect to the hospital web server.
>
> Hint: Use the PING command from the command line on each PC to determine which PC is able to successfully connect to the web server at 192.0.2.254. It may be necessary to PING multiple times.
>
> Hint: Investigate the configuration settings on the one PC that can connect to the web server.
>
> Hint: Correct the settings on the other PCs. All PCs should be on the same IP network as their default gateway – the LAN interface of the Linksys router.
>
> Verify that all PCs can connect to each other and can reach the web server at 192.0.2.254 by firing the pre-defined PDU connectivity tests."

A general outline of steps to be taken would be: 1) read the instructions, 2) determine which computer is the working one, 3) compare the configuration of this computer to the other computers to determine what is incorrect on the other computers, 4) reconfigure the other computers, and 5) test the solution. In reality, there are a number of correct ways players could do this, for example they could compare, fix, and test one PC then move on to the next or they could compare and fix each computer and test them all at the end. An incorrect method might involve implementing solutions prior to determining which the working computer is or changing settings on the web server rather than the PCs.

There are two solution paths that could be followed. The computers are all initially set with what are called static Internet Protocol (IP) addresses. This means they are manually entered by the person configuring the computer. One solution path involves determining what is wrong with these addresses and manually correcting them. A second solution path involves changing the settings so the PC requests an address automatically from the router. The instructions suggest the first path, but the second path works as well and is somewhat easier and faster. However, if this path is used, the player should not change the already working PC to the automatic setting. If he/she does, the end tests will not work. It is designed this way because in the real world technicians often set one

computer to a static address for a specific reason and this should not be changed to an automatic setting without good reason. Someone who does make this change is possibly applying the solution indiscriminately without understanding the full consequences of their actions.

3.4. *Coding*

The game actions for each recording were transcribed and treated as separate events. The recordings were played along with the transcriptions and codes assigned based on the dialog and game moves made. The following items were coded: problem identification, solution generation/implementation, solution evaluation, prior knowledge: subnets, prior knowledge: DHCP, procedural knowledge: ping, procedural knowledge: PDU, and expert intervention. Each of these will be explained below.

3.4.1. *Problem solving steps*

First, given that the game was intended to require technical problem solving skills, we needed to select a coding scheme for the steps in which we were interested. After reviewing various descriptions of problem solving, the parsimonious three step process described by Newell & Simon (1972) (orientate, solve, evaluate) was used as an initial coding scheme. To remain consistent with more recent terminology, the three steps were called problem identification, solution generation/implementation, and solution evaluation. Initially, solution generation and implementation were viewed as separate. However, after viewing the videos it was determined that it was exceedingly difficult to separate the two based on the actions and discussions of the pairs. Therefore, most instances were being coded as both and the codes were subsequently combined.

Problem identification was coded in all cases where players were attempting to define and understand the problem. Players identifying which computer worked and then comparing the configuration of that computer to others was coded as engaging in problem identification. In addition, players' attempts to diagnose problems by looking at other elements of the network (which were in fact not broken) were also coded as problem identification. That is, the players did not necessarily have to be doing the correct or prescribed diagnostic activities, as long as it was clear that they were attempting to find the problems in the network. Again, it should be noted that Packet Tracer manifests itself in the Aspire game as providing open-ended problem identification, diagnosing, and solution environment.

Solution generation/implementation was coded whenever 1) students discussed possible solutions to a problem or 2) students were actively changing elements of the network in an attempt to solve the problem. In the ideal solutions, this would be changing the IP addresses and subnet masks of the non-working PCs. However, this was also coded when students were making other changes to solve what they perceived the problem to be.

Finally, solution evaluation was coded when students were testing whether their solution had fixed the problem or discussing the results of that test. There were three primary activities coded in this section. First, students could use the ping command

(sending an "echo request" message hoping for an "echo reply" from the target) on the computer to test the connection (see Figure 2). Students could use the Command Line Interface (CLI) of the PC to enter the command. By observing whether they were able to get a reply from the target device, they can tell whether they have implemented a successful solution. Second, they can use the PT interface to send test packets (as required in the activity). These packets are used by Packet Tracer similar to the ping command (see discussion of PDU below). Third, students can use the game's checkmark system to determine if the game judged them to have completed a section of the task. When students successfully complete a task (as judged by the comparison of their network to an answer network), the task description changes color and is checked off on their task list. This evaluation is a fairly simple question of whether the solution worked. As Jonassen (2000) notes, troubleshooting problems usually have easily interpreted success criteria. However, this does not diminish the fact that the solution must be tested and a judgment made about whether it is correct.

3.4.2. *Knowledge, skills, and abilities*

Given that the interest was in validating not just that students were engaged in generic problem solving, but using networking-related knowledge and skills to play the game, a specific set of these skills thought to be essential for completing the task was identified and coded, as identified below. Figure 3 presents a concept map of how these concepts relate to each other and the process of correctly configuring a network.

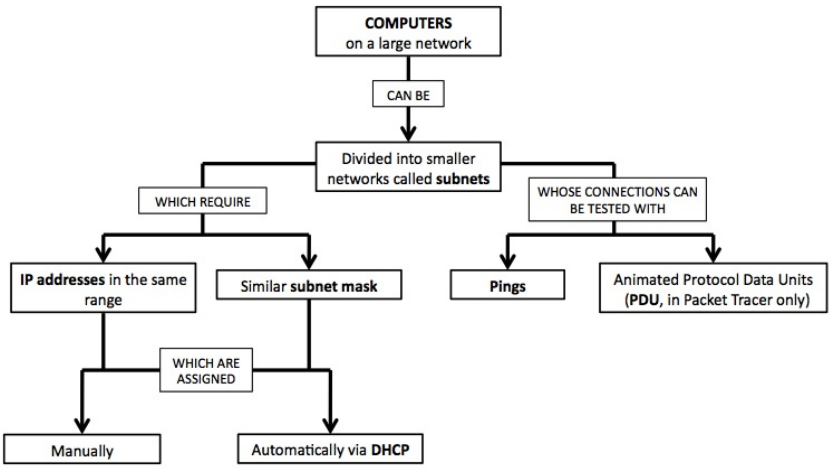

Figure 3. Concept map for IP addressing and testing in the domain of computer networking.

3.4.2.1. Prior knowledge: Subnets

In the Internet Protocol, smaller sections of the overall network are given hierarchically organized identifying addresses, not unlike postal or telephone area codes; to participate in the overall network, the address ranges of the component networks – subnets – must follow strict rules. Knowledge of this was coded any time students' statements or actions indicated that they understood: 1) the computers all had to be on the same subnet, 2) the subnet masks (a number that is used to divide the IP address into a network address and an individual computer address) all had to be the same to make this happen, and 3) the IP addresses all had to be in the subnet range.

3.4.2.2. Prior knowledge: DHCP

As mentioned above, an alternate solution path to the problem involves setting the computers to get their addresses dynamically. This is called a Dynamic Host Control Protocol (DHCP). Given that this eliminates the need for students to manually determine the "legal" addresses that must be used to allow the computers to connect to each other, this was coded as an alternative when pairs changed the computer settings to DHCP or discussed doing so.

3.4.2.3. Procedural skills: Ping

This was coded when students correctly used the ping command to test connectivity in the network. This command is issued using the command prompt on the PC and can be used either at the beginning of the contract to find problems or at the end to test solutions. The initial fault symptom (to use Jonassen and Hung's (2006) terminology) in this contract was that issued ping commands on three of the four PCs will not receive "affirmative" responses.

3.4.2.4. Procedural skills: PDU

PDU stands for protocol data unit. In the case of Aspire, it is a one-time echo request message (to which an echo reply is anticipated), it can be thought of as simulation equivalent – depicted by a little envelope representing the echo request or echo reply data packet – that is one part of a ping command issued from the command line. As such, it is a sort of graphical ping, a way to help visualize the flow of the test packets. After watching the videos, it was determined that some students used the predefined PDUs to test connectivity throughout the contract. This code was added to compare this use with the use of the ping command issued from the virtual command line interface. Use of the graphical ping (PDU) may imply more background and knowledge of Packet Tracer since it is a skill specific to PT and not available on the real equipment labs that are also part of the students' learning environments.

3.4.2.5. Expert

Finally, expert help was coded as present if help was needed from a teacher, researcher, or other expert to complete the task.

3.5. *Interrater agreement*

A primary researcher coded all of the videos. A secondary researcher was then trained on the codes via discussion of the codes and examples and coding of sample videos. The secondary researcher coded five of the 11 videos. The interrater reliability between the two coders after this initial coding was kappa = 0.77. Although benchmarks for kappa values are somewhat arbitrary, Fleiss *et al.* (2003) suggests values over 0.75 indicate excellent agreement and Landis and Koch (1977) suggest values from 0.61 to 0.80 indicate substantial agreement. For this study, each instance of disagreement was discussed and agreement reached on the code. The videos for the remaining pairs were then revisited by the primary researcher to ensure consistency in the coding.

4. Results

4.1. *Qualitative description of play*

Before looking at the specific codes and coding of game play, it is useful to provide context for those codes through a general summary of the game play of each pair (see Table 2). In general, the interactions between the members of each pair could be characterized as collaborative. There was no conflict observed between players, even in pairs that spent more than 20 minutes struggling to identify the problem and correct solutions. Rather in approaching the contract, in most pairs, one member read the initial invitation out loud. Then, one member of the pair would suggest an initial action, which would then be implemented. In cases where players disagreed on a course of action, one member was usually quick to give in and the alternate course of action was attempted. After reading the initial introduction, some pairs began immediately exploring the equipment while others read the more detailed instructions and hints. As suggested in Table 2, those pairs that solved the problem quickly (for example, pairs 3 and 5) had less interaction between the players, as they both agreed on a solution and its implementation. The pairs that struggled more (for example, pairs 1 and 2) required more interaction as they identified many potential problems and solutions. Players used and discussed feedback about their solutions obtained from the game to make decisions about subsequent actions. For example, pair 6 began changing subnet masks to solve the problem, but when they got to PC0, they found that the subnet mask was already correct. The game prompted a conversation among the players about what else might be wrong, and an exploration of the devices in the game to examine the address ranges. This interplay of the game and players was observed across pairs.

Table 2. Brief description of game play by pair.

ID	Course	Strategy
P1	D1	This pair spent the most time of any pair trying to figure out the problem. They finally came to DHCP as a solution and implemented it. The both contributed to brainstorming about the problem and solutions but neither demonstrated the required content knowledge.
P2	D1	Pair 2 made changes one step at a time as they realized their previous step did not fix the problem. They had difficulty in the end because they were not familiar with PDUs in PT. One member of this pair generated most of the solution ideas, while the other member implemented them in the game.
P3	D3	Pair 3 quickly found that PC3 worked and changed IP addresses and subnet masks on other PCs to match. The player who operated the computer talked out loud while troubleshooting. The other player mostly confirmed the actions of the first player.
P4	D1	Pair 4 quickly found that PC3 worked and changed PC1 and PC2 subnet masks, then went to change IP addresses but changed the address on PC3 by mistake. They required researcher to intervene. Both members of the pair suggested potential problems and solutions during play.
P5	E4	Pair 5 spent a short time identifying which PC works, but quickly changed all computers to receive addresses dynamically via DHCP. This pair spoke very little with the player who was “driving” making nearly all the decisions.
P6	E1	Pair 6 quickly found that PC3 worked and changed IP addresses and subnet masks on others to match. PDUs from PC0 and PC3 failed on first attempt. Rather than firing again, they spent a lot of time looking at the configurations trying to figure out what was wrong until they happened to refire the PDUs. Both members of this pair actively engaged in problem solving, offering suggestions and solutions.
P7	E4	Pair 7 quickly found PC3 worked and changed IP addresses and subnet masks on others to match.
P8	E1	Pair 8 issued ping commands from PC0 and PC1, found they didn’t work, and immediately changed to DHCP. They changed PC2 and PC3 to DHCP without pinging first. They spent time trying to figure out why the end tests wouldn’t work and needed an expert to help get PC3's address back.
P9	E1	Pair 9 changed all the PCs to DHCP without doing any investigation of which ones were working. They also changed the WebServer to DHCP. They required expert intervention to get addresses back. In the end still ended up with all but PC3 using DHCP.
P10	E4	Pair 10 changed all the PCs to DHCP with no checking to see which PCs were working; they needed expert help to get back the PC3 configuration.
P11	E2	Pair 11 quickly found PC3 worked and changed IP addresses and subnet masks on others to match.

4.2. *Problem solving*

Figure 4 presents each pair's use of the three problem solving steps over time. Time passing (in minutes and seconds) is represented from left to right. Each occurance of problem identification, solution generation/implementation, and solution evaluation is graphed over the time at which it occurred. The pairs are arranged so that the pairs in beginning classes are at the top and those in advanced classes are at the bottom.

First, we can see that those in beginning classes took much longer to solve the problem than those in advanced classes. The time spent on the activity for those in the beginning classes ranged from 14 minutes 5 seconds to 25 minutes 50 seconds. In contrast, the advanced students took from 4 minutes 43 seconds to 7 minutes 58 seconds to complete the activity.

Second, eight of the 11 pairs engaged in problem identification for at least the first two minutes of the task. Activity tended to occur in cycles of problem identification, solution generation/implementation, and solution evaluation. Those pairs in beginning classes have more cycles on average than those in advanced classes. Analysis of the transcripts suggests that this is because their initial problem identification or solution implementation was incorrect. For example, when we look at the graph for pair 4, we see an initial cycle of problem identification lasting more than four minutes, followed by solution generation, more problem identification, solution generation/implementation, and solution evaluation. We then see them go through five more cycles from problem identification through solution evaluation. Examination of the transcript reveals that pair 4 initially identified the problem that the IP addresses were not all in the subnet range, but changed them to the wrong range. When their evaluation indicated the solution was incorrect, they went back to problem identification activities and identified that the problem was that the pre-defined PDUs were configured incorrectly and sought to change them. They then re-identified the problem that the IP addresses were in the wrong range, but attempted to solve that by changing the router rather than the PCs. They required expert intervention to get the configuration for PC3 back to its original state and were then able to configure the other PCs.

Finally, it is noteworthy that almost every activity that players did in the contract was coded as one of the three steps. In other words, nearly every game move the players made in this contract was either an attempt to identify, solve, or evaluate the solution to the problem (as opposed to exploring the office, engaging in off-task conversation, etc.). The few exceptions can be seen where there are breaks in the lines, such as in pair 2 around the 17:00 mark.

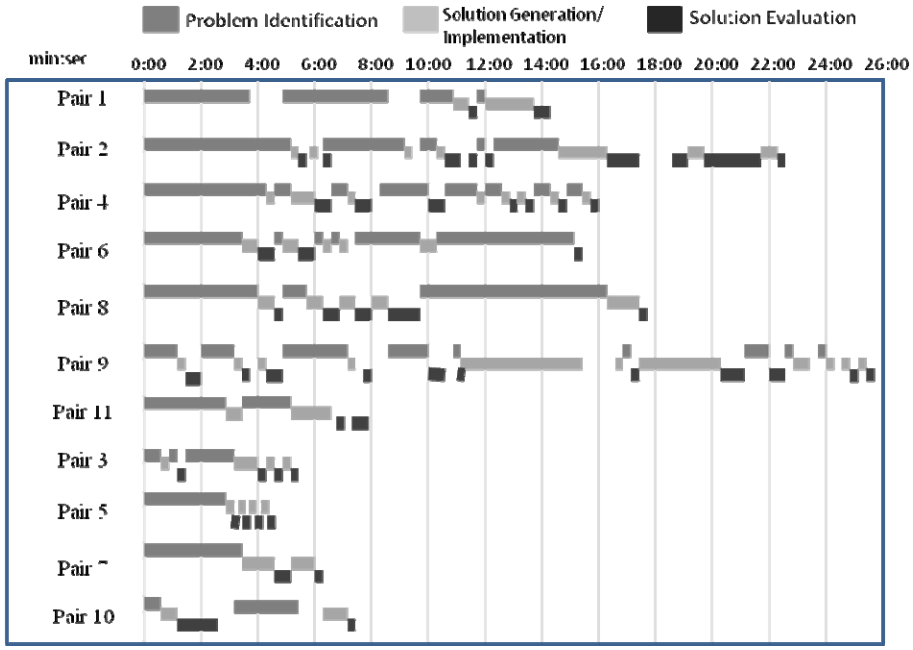

Figure 4. Time series of problem solving steps for each pair.

4.3. *Knowledge, skills, and abilities*

4.3.1. *Prior knowledge: Subnets*

Knowledge of the concepts of subnets is important for completing the task (if a DHCP solution is not used). Figure 5 presents a time series display of the use of prior knowledge presented on top of the problem solving steps as presented in Figure 4. The timing of the first indication of subnetting knowledge is displayed with the diamonds. We see seven of the 11 pairs have diamonds at some point in their graph. For example, we see that pair 6 (the fourth pair in the graph) exhibited knowledge of subnetting just after four minutes into the contract. For this pair, the code was assigned based on one player noting that the subnet mask should not have to change, but should be the same for all of the PCs. Generally, the first indication of subnetting knowledge is followed within a few seconds by solution generation/implementation. This is the case with pair 6, as the graph shows solution implementation occurring immediately after the diamond shape. This was coded when the pair opened PC2 and changed its subnet mask to match that on PC3. Looking across all of the pairs, we see evidence that the game elicited and encouraged application of subnetting knowledge.

Looking again at differences between beginning students at the top of Figure 5 and advanced students at the bottom, all of the pairs in advanced classes who used subnetting

knowledge did so for the first time in less than four minutes, while only one of the five pairs in beginning classes demonstrated it this quickly. The diamonds for pairs 3, 7, and 11 are all placed to the left of the four minute line for these advanced students. Only pair 4 among the beginning students displayed subnetting knowledge this quickly (they then, as described above, used it incorrectly).

4.3.2. *Prior knowledge: DHCP*

There remain four pairs who completed the contract without displaying subnetting knowledge. However, there is also a second method, involving knowledge of DHCP, that can be used to complete the problem. Therefore, we next sought to understand whether pairs used this second, expected, concept, or other, unexpected, knowledge to fix the problems in the network. The ovals in Figure 5 represent the time at which players demonstrated knowledge of DHCP. Examination of the graph reveals that seven of the 11 pairs displayed knowledge of DHCP, and some pairs invoked knowledge of both subnetting and DHCP. Pair 5 used just DHCP knowledge to solve the problem, pair 3 invoked DHCP initially and then also used subnetting knowledge, and pair 1 moved from first using knowledge of subnetting to DHCP. Three pairs of players (3, 9, and 10) jumped to the use of DHCP within the first minute of the contract, without doing the testing to see which PC was working.

Pairs 2 and 3 tried DHCP, but did not ultimately use it to solve the problem. Pair 3

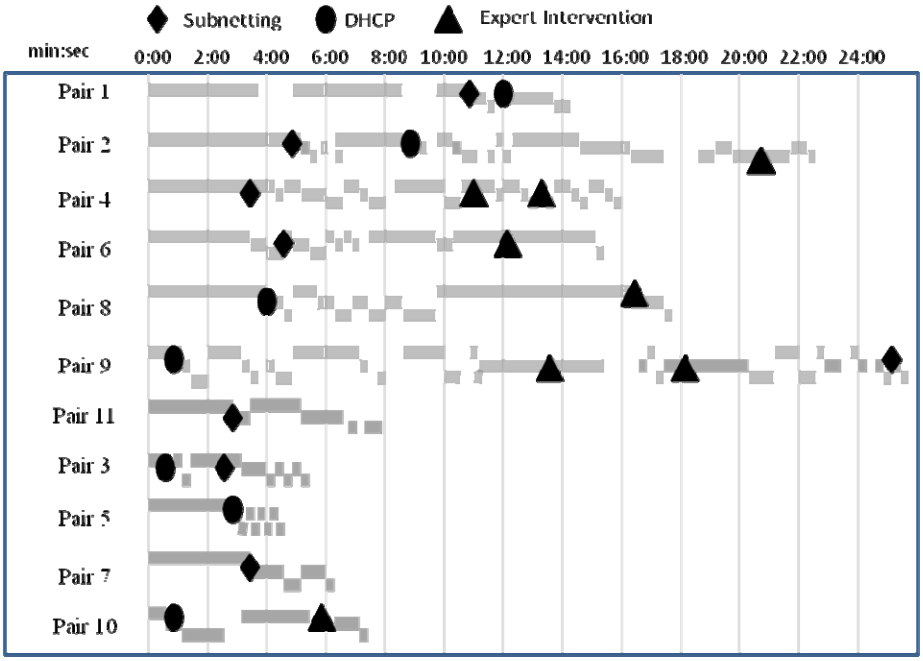

Figure 5. Time series of procedural knowledge demonstration for each pair.

went first to switch the computers to DHCP, but when they changed the first computer, the DHCP request failed, so they abandoned that strategy. Pair 2 temporarily switched some computers to DHCP, but ended up switching them back to static and configuring them manually.

As with subnetting, instances of DHCP are closely tied temporally with solution generation and implementation. Unlike subnetting, the first coding of DHCP was often the action of changing the setting on a PC. There was little discussion of the use of DHCP among the pairs, just implementation of it. For example, neither pair 1 nor pair 5 had any dialog at all about DHCP. One of the members of pair 9 just said, "DHCP" and they made the change.

4.3.3. *Procedural skills: Ping*

The contract instructions told players to use the ping command to determine which PC was working initially. Figure 6 displays the use of the ping command in this role of a symptom identifier. If the players start at PC0 and work sequentially, they need four pings to get to PC3 (the working PC). Four pairs (1, 2, 7 and 11) went through this sequence. Four pairs (3, 4, 9, and 10) did not issue pings prior to implementing solutions. Pairs 5 and 6 started with PC3 and immediately saw that it was working. These two combined with the previous four pairs who went through the whole sequence resulting in six pairs that would be considered to correctly use the ping command to identify symptoms; three of these were in beginning classes and three in advanced. Pair 8 implemented a solution without identifying a working PC.

4.3.4. *Procedural skills: PDUs*

Figure 6 also displays instances of the use of PDUs for symptom identification. Two pairs of students (4 and 10) used PDUs instead of the ping command for initial problem identification. Unfortunately the PDUs were not set up to test the connection from each PC to the web server so this strategy was not effective in the identification of the working and non-working PCs.

All of the pairs used the predefined PDUs to test their final configuration (as was required to complete the contract). However, there were two examples of problems associated with PDUs that are informative. First, pair 2 was unfamiliar with the term PDU and did not know what the instructions were referring to when telling them to fire them. They required intervention from the researcher to figure out what was required. Second, some PDUs of pair 6 failed when first fired. The students then spent the next 6.5 minutes trying to find the error in the network (when in fact the configuration was correct but needed the PDU to be resent).

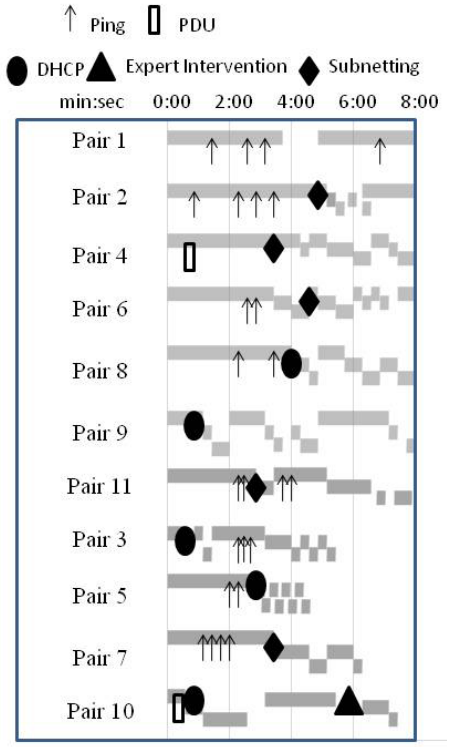

Figure 6. Use of ping and PDU commands for initial symptom identification.

4.3.5. *Expert*

Figure 5 marks instances of requiring expert intervention with a triangle. Five of the six pairs in beginning classes required expert intervention at some point to complete the contract while only one of the five pairs in advanced classes did. Pairs 2 and 6 needed help to figure out what firing PDUs meant and that they might need to be fired multiple times. Pair 4 needed intervention because they changed the IP address of the working PC, set the other PCs to match the new information, and could not figure out why things were not working. Additionally, they needed a prompt to "think of subnets" in order to realize they needed to change the client portion of some of the addresses. Pairs 8, 9, and 10 (the one pair from the advanced class) needed help resetting the working PC's addresses after changing them to DHCP.

5. Discussion

This paper provides an intersection of three areas: games, troubleshooting, and validity theory. It examines the problem solving processes, knowledge and skills used by players

completing a troubleshooting task in a simulation-based game in order to evaluate its substantive validity. Specifically, the aim is to demonstrate that problem solving skills, prior knowledge about networks, and procedural skills involved in configuring networks are used to complete complex game tasks. The game is designed to allow students the opportunity to practice specific skills and to encourage consideration of particular content. If we observe students using these target skills and knowledge, we have provided evidence of substantive validity, and can argue that the game provides an opportunity to practice these skills. Although we would not expect students to be off task while being observed and recorded, there was a potential that students would find ways to complete the tasks without using the targeted problem solving or networking skills; they might have used different skills than intended, have used inadvertent clues from the game to solve the problem, or gotten far off task in the richness of the game's simulated microworld. Second, seeing general variation in response processes indicates that the game is able to capture potentially meaningful differences in cognitive processes. Finally, observing variation in response processes that covaries with the sequence of classes provides further suggestion that performance in the game is related to networking skill.

Examining the performance of 11 pairs of students completing one contract in the game, it was observed that students engaged in cycles of problem solving steps: problem identification, solution generation/implementation, and solution evaluation. As might be expected, problem identification stages were generally longer than solution generation/implementation or solution evaluation stages and students in beginning classes had more cycles due to errors in problem identification and solution implementation. Observing players engaging in activities that indicate problem solving activity confirms us that the game does provide opportunity for students to practice these thinking skills.

From Jonassen and Hung's (2006) troubleshooting perspective, players needed initially to, "seek out and recognize faulty components" (p. 18). In the game, it was important to identify not only what was faulty, but also what was working. Ten of the 11 pairs engaged in some form of this process, with varying degrees of success. Pairs that were not successful at identifying the working computer generally tried to "fix" that one too, had difficulty solving the problem, and required expert intervention. Similarly, players who used less effective methods for finding the faulty PCs spent longer on the task and/or reached the wrong conclusions.

The next troubleshooting process requires diagnosing faults, requiring players to "identify discrepancies between existing states and normal states and interpret those discrepancies in terms of their conceptual model of the system components in order to generate plausible hypotheses" (Jonassen & Hung, 2006, p. 19). We know that experts are generally quicker to arrive at a diagnosis because they are able to compare problems to an internal database of other, similar problems they have encountered. In this sample, we saw students in more advanced classes were quicker to bring the appropriate subnetting and/or DHCP knowledge to the problem. Students in beginning classes on the other hand, were more likely to identify incorrect problems, such as the pair who thought the fault test itself was incorrect, and require more cycles of problem identification.

The next process in troubleshooting requires generation and implementation of solutions for repairing the broken system (Jonassen & Hung, 2006). The examination of the use of DHCP provides an interesting view about solution selection implementation. In general, the DHCP solution is easier and faster, things that research suggests would make the solution attractive to experts. However implementation of DHCP indiscriminately leads to the most severe roadblocks in the game resulting from making changes in the equipment that they could not undo without assistance. This highlights the fact that the game requires thoughtful solution implementation.

Finally, players have to evaluate their solutions. The game requires the use of the PDU mechanism to do this. PDUs in the game are an abstraction of a real network feature specific to Packet Tracer. Therefore, knowledge of the use of PT is required to understand and carry out instructions involving these PDUs. This raises a potential tension in the game between learning game-playing skills versus learning networking skills. Other researchers (Barab *et al.*, 2009) found evidence of a similar tension between learning how to use a simulation tool and learning the content itself.

In conclusion, these analyses demonstrate that the knowledge, skills, and abilities hypothesized to be involved in the successful completion of this game contract are in fact displayed by this sample of players. The players engaged in problem identification, solution generation/implementation, and solution evaluation. They applied prior knowledge regarding subnetting and DHCP to the task, used the ping command, and fired PDUs. This evidence of substantive validity suggests that the game provides an opportunity for students to practice and solidify computer networking-related skills.

5.1. *Cautions*

This study is based only on 11 pairs of students. It was designed to use descriptive methods to find patterns of processes used to solve problems, which it has. However, conclusions about patterns like the differences between beginning and advanced students need to be verified on larger samples by using a variety of techniques. While the use of students from different class levels allowed for comparison, it also further decreased the sample of beginning students, the target of the game, and potentially diluted conclusions that might be reached based on them. In addition the students who participated in the study were all described by their teachers as being at least at an average level of ability, so it is not clear how lower functioning students would play the game.

In qualitative research issues of credibility, trustworthiness, and bias of research come to the forefront because the interpretation of the data relies on the judgment of the researcher. The largest source of researcher bias in this study is likely that all of the members of the research team are employed by the Cisco Networking Academy and most were members of the team that developed the game being studied. As such, it might be assumed that we have a vested interest in finding evidence of validity of inferences about students. Although clarification of bias can be seen as one strategy for validation of qualitative results (Creswell, 2003), we also attempted to present negative or discrepant information, used a prolonged time over which to gather data (five months), and engaged

in peer debriefing, particularly regarding application of codes. We believe the use of a qualitative approach for suggesting evidence of substantive validity also will point the way to quantitative parameters of interest (which are easily obtained given the assessment architecture used to determine user's progress through the game).

5.2. *Future research*

This study focused on gathering evidence regarding response processes used as evidence for substantive validity. Within the results, a clear pattern in the amount of time taken to solve the problems for students at different class levels was suggested. This suggests the need for studies gathering evidence of validity based on external measures. We might expect students who are in more advanced classes or have higher exam scores to be able to complete more contracts and complete contracts more quickly with fewer errors, for example. These larger scale quantitative studies would further establish evidence for inferences about students' problem solving and networking ability based on game play.

Second, interesting questions about the need for assistance were raised by this study. First, there does appear to be a role for someone familiar with the game to assist students, given that five of the six students in beginning classes required help. It should be noted that the teachers of pairs 1, 2, and 6 were present and occasionally intervened in the game play. However, they were not familiar with the game and their interventions did not lead to solutions. So, classroom instructors need to be familiar with the game in order to provide more efficient guidance. In addition, it appears there might be a role for game-based help that is dependent on time. In all three cases of expert intervention, there were long periods of "fumbling" or clicking through devices without doing anything meaningful, prior to help being given. It was clear to the human observer that help was required, although the specifications for computer cues would need further examination.

5.3. *Conclusion*

Gee (2009) suggests that rather than giving tests after learning activities, the completion of the activity itself should signal that a student has mastered the content. As an example, he argues that someone who plays Halo and finishes does not then need to take a test to show s/he is proficient at Halo. Similarly, completion of an educational game should signal that someone is proficient in the skills of that game. However, given the complexity of game environments, before these conclusions can be drawn, we need to determine whether there are construct-irrelevant processes and clues that players use to successfully complete tasks.

Gathering evidence in support of substantive validity is a way to formalize and document the investigation of these processes. In this paper, we have demonstrated that completing the troubleshooting contract in this game required the use of troubleshooting skills and domain specific knowledge. We argue that in an educational gaming environment the issue of substantive validity is an essential one for both guiding the designing process and establishing inferences about students. Ultimately, understanding

the response processes used to solve the game problem allows us to be more confident in the inferences we make about students who play.

Acknowledgments

The authors wish to thank Pamela Vinco for her assistance with graphics and review, all of the students and instructors who participated in the study, and the two anonymous reviewers whose comments improved the paper.

References

Barab, S. A., Barnett, M., Yamagata-Lynch, L., Squire, K., & Keating, T. (2002). Using activity theory to understand the systemic tensions characterizing a technology-rich introductory astronomy course. *Mind, Culture, and Activity, 9*(2), 76–107.

Barab, S. A., Gresalfi, M. S., & Arici, A. (2009). Transformational play: Why educators should care about games. *Educational Leadership, 67*, 76–80.

Behrens, J. T., Frezzo, D. C., Mislevy, R. J., Kroopnick, M., & Wise, D. (2007). Structural, functional , and semiotic symmetries in simulation-based games and assessments. In E. L. Baker, J. Dickieson, W. Wulfeck, & H. F. O'Neil (Eds.), *Assessment of problem solving using simulations* (pp. 59–80). New York: Erlbaum.

Besnard, D., & Bastien-Toniazzo, M. (1999). Expert error in troubleshooting: An exploratory study in electronics. *International Journal of Human-Computer Interaction*, 50, 391–405.

Blackwell, R. T., Galassi, J. P., Galassi, M. D., & Watson, T. E. (1985). Are cognitive assessment methods equal? A comparison of think aloud and thought listing. *Cognitive Therapy and Research, 9,* 399–413.

Bransford, J. D., & Stein, B. S. (1993). *The IDEAL problem solver* (2nd ed.). New York: Freeman.

Clark, D., Nelson, B., Sengupta, P., & D'Angelo, C. (2009). *Rethinking science learning through digial games and simulations: Genres, examples, and evidence.* Paper presented at National Academies of Sciences Learning Science: Computer Games, Simulations, and Education conference, Washington, DC. Retrieved from http://www7.nationalacademies.org/bose/Clark_Gaming_CommissionedPaper.pdf

Cordova, D. I., & Lepper, M. R. (1996). Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of Educational Psychology, 88*, 715–730.

Creswell, J. W. (2003). *Research design.* Thousand Oaks, CA: Sage Publications.

Ericsson, K. A., & Fox, M. C. (2011). Thinking aloud is not a form of introspection but a qualitatively different methodology. *Psychological Bulletin, 137*, 351–354.

Fleiss J. L., Levin B., & Paik M. C. (2003). *Statistical methods for rates and proportions* (3rd ed.). Hoboken: John Wiley & Sons.

Frezzo, D. C. (2009). *Using activity theory to understand the role of a simulation-based interactive learning environment in a computer networking course* (Doctoral dissertation). Retrieved from ProQuest http://gradworks.umi.com/33/74/3374268.html

Frezzo, D. C., Behrens, J. T., & Mislevy, R. J. (2010). Design patterns for learning and assessment: Facilitating the introduction of a complex simulation-based learning environment into a community of instructors. *Journal of Science Education and Technology, 19*, 105–114.

Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York: Palgrave/ Macmillan.

Gee, J. P. (2009). *Games as their test. Spotlight on Digital Media*. Retrieved from http://spotlight.macfound.org/blog/entry/james-paul-gee-games-as-their-test/

Hayes, J. R. (1989). *The complete problem solver* (2nd ed.). Hillsdale, NJ: Erlbaum.

Hickey, D. T., Wolfe, E. W., & Kindfield, A. C. H. (2000). Assessing learning in a technology-supported genetics environment: Evidential and systemic validity issues. *Educational Assessment, 6*(3), 155–196.

Jonassen, D. H. (2000). Toward a design theory of problem solving. *Educational Technology: Research and Development, 48*(4), 63–85.

Jonassen, D. H., & Hung, W. (2006). Learning to troubleshoot: A new theory-based design architecture. *Educational Psychology Review, 18*, 77–114.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics, 33*, 159–174.

Leow, R. P., & Morgan-Short, K. (2004). To think aloud or not to think aloud: The issue of reactivity in SLA research methodology. *Studies in Second Language Acquisition, 26*, 35–57.

Messick, S. (1995). Validity of psychological assessment: Validation of inferences from a persons' responses and performances as scientific inquiry into score meaning. *American Psychologist, 50*, 741–749.

Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (2003). On the structure of educational assessment. *Educational Measurement: Issues and Practice, 25*(4), 6–20.

Moss, P. A., Girard, B., & Haniford, L. (2006). Validity in educational assessment. *Review of Research in Education, 30*, 109–162.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.

Park, O., & Lee, J. (2003). Adaptive instructional systems. In D. H. Jonassen (Ed.), *Handbook of research for educational communications and technology* (pp. 651–685). Mahwah, NJ: Lawrence Erlbaum.

Partnership for 21st Century Skills. (2009). *P21 framework definitions*. Retrieved from http://www.p21.org/documents/P21_Framework_Definitions.pdf

Schmit, M. J., & Ryan, A. (1992). Test-taking dispositions: A missing link? *Journal of Applied Psychology, 77*, 629–637.

Shaffer, D. W. (2006). *How computer games help children learn*. New York: Palgrave Macmillan.

Shepard, L. A. (1993). Evaluating test validity. *Review of Research in Education, 19*, 404–450.

Shute, V. J., Lajoie, S. P., & Gluck, K. A. (2000). Individualized and group approaches to training. In S. Tobias & J. D. Fletcher (Eds.), *Training and retraining: A handbook for business, industry, government, and the military* (pp. 171–207). New York: Macmillan.

Shute, V. J., Ventura, M., Bauer, M. I., & Zapata-Rivera, D. (2009). Melding the power of serious games and embedded assessment to monitor and foster learning: Flow and grow. In U. Ritterfeld, M. Cody, & P. Vorderer (Eds.), *Serious games: Mechanisms and effects* (pp. 295–321). Mahwah, NJ: Routledge, Taylor and Francis.

Squire, K. (2006). From content to context: Videogames as designed experience. *Educational Researcher, 35*(8), 19–29.

Sundre, D. L., & Wise, S. L. (2003, April). *'Motivation filtering': An exploration of the impact of low examinee motivation on the psychometric quality of tests*. Paper presented at the annual meeting of the National Council on Measurement in Education, Chicago, IL.

The Conference Board. (2006). *Are they really ready to work?: Employers perspectives on the basic knowledge and applied skills of new entrants ot the 21st century U.S. workforce.* Retrieved from http://www.p21.org/documents/FINAL_REPORT_PDF09-29-06.pdf

Van Someren, M. W., Barnard, Y. F., & Sandberg, J. A. C. (1994). *The think-aloud method: A practical guide to modelling cognitive processes.* London: Academic Press.

Virzi, R. A., Sorce, J. F., & Herbert, L. B. (1993). A comparison of three usability evaluation methods: Heuristic, think-aloud, and performance testing. *Human Factors and Ergonomics Society Annual Meeting Proceedings*, pp. 303–315.

Willingham, D. T. (2007). Critical thinking: Why is it so hard to teach? *American Educator, 31*(2), 8–19.