Applying Gaming Principles to Support Evidence-based Instructional Design

Robert Wray. Ph.D., Angela Woods
Soar Technology, Inc.
Ann Arbor, MI 48015
wray,angela.woods@soartech.com

Heather Priest, Ph.D.
U.S. Army Research Institute
Orlando, FL 32826
heather.priest@us.army.mil

ABSTRACT

There are a number of presumed benefits to using games for training including greater portability, improved training effectiveness, increased student motivation, and overall cost effectiveness. Unfortunately, not all of these presumed benefits have proven to be true; for others, experimental evidence has yet to confirm them. On the other hand, evidence so far suggests that current approaches to game-based training are not a cost effective replacement for direct instruction. We hypothesize the gaps between evidence-based instructional design and game-based technologies can be bridged, with the goal to provide engaging and effective learning via Instructional Games. The paper describes two complementary research and development thrusts:

1. Content-design principles that place relatively greater emphasis on iterative development of skills and capabilities via direct instruction, guided practice, and on-going assessment, while also maintaining central elements of gaming experience such as interactivity and positive feedback.

2. Technologies that organize and constrain student experience to follow proven instructional design methods and fill functional gaps. Examples include a *Lesson Designer* that enables rapid authoring and integrated presentation of instructional content within a game-based practice environment; an *Instructional Game State Manager* software component that integrates instructional design principles and the user-experience managers used in computer games; and contextual feedback delivery that provides learner support and feedback during practice. The integrated toolset gives students and instructors the ability to capture and annotate game sessions for use in presentation (e.g., after action review briefings) and instruction (demonstrations of concepts or skills).

We illustrate the principles and supporting tools with existing examples of an instructional game focused on the US Army Military Decision Making Process. The instructional game prototype was developed using the tool suite. However, the tool suite is general purpose and can be applied to future instructional games in other domains.

ABOUT THE AUTHORS

Robert E. Wray is a Senior Scientist at Soar Technology. He received a Ph.D. in computer science and engineering from the University of Michigan. At Soar Technology, he leads or has led R&D projects for the Air Force and Navy research offices, and the Defense Advanced Research Projects Agency. Dr. Wray's research encompasses many areas of artificial intelligence research including agent-based systems, machine learning, cognitive science, intelligent tutoring systems, and knowledge representation and ontology. He is presently founder and lead scientist for Soar Technology's Dynamic Tailoring and Simulation2Instruction product lines.

Angela J. Woods is a Software Systems Engineer at Soar Technology. She has a background as a game developer; her shipped titles include Call of Duty Roads to Victory (PSP), Pirates of the Caribbean: Dead Man's Chest (PSP) and The Sims2 (PSP). Areas of responsibility as a game programmer included AI programming, game play mechanics, rigid body physics, performance profiling, optimization and technology pipeline R&D. She also has in depth experience with military simulations, and was a key contributor in the development of OneSAF versions 3.0-5.0. At Soar Technology, she is the software lead for teams that create intelligent agents for games and training.

Heather A. Priest is a Research Psychologist at the U.S. Army Research Institute (ARI), Training Technology Unit. She received her Ph.D. in Human Factors Psychology from the University of Central Florida and her M.S in Experimental Psychology from Mississippi State University. At ARI, her research involves intelligent tutoring systems and adaptive training, automated feedback in game-based training, instructional design principles for training technology, and synthetic teammates in game-based training. Along with numerous publications, Dr. Priest also served as a co-editor on the Adaptive Training Special Issue of *Military Psychology (2012).*

Applying Gaming Principles to Support Evidence-based Instructional Design

Robert Wray. Ph.D., Angela Woods
Soar Technology, Inc.
Ann Arbor, MI 48015
wray,angela.woods@soartech.com

Heather Priest, Ph.D.
U.S. Army Research Institute
Orlando, FL 32826
heather.priest@us.army.mil

INTRODUCTION

There continues to be exploding interest in using games and game technology for education and training. This interest is due in large part to a number of *presumed* benefits to using games including greater portability, improved training effectiveness, increased student motivation, and overall cost effectiveness. Unfortunately, not all of these presumed benefits have proven to be true; for others, experimental evidence has yet to confirm them. On the other hand, the evidence so far suggests current approaches to game-based training are not a suitable or cost effective replacement for direct instruction (Clark, 2007; O'Neil, Wainess, & Baker, 2005).

The current generation of training games is not typically grounded in evidence-based instructional design and pedagogical methods. Clark's (2007) review of evidence on the value of games for learning from both military and civilian research states, "There is no evidence in the existing studies that games teach anyone anything that could not be learned some other, less expensive, and more effective way" (p. 57).

Clark also suggests serious games *could* provide effective and motivating instruction for complex knowledge *if they were adequately designed for learning*. Games typically enable practice. However, Merrill (2002) suggests while the task focus enabled by games is central to learning, practice (or "application") is only one of four phases needed to structure task-centered learning effectively. Relating new concepts and skills to prior experiences (activation), demonstration, and integration (enabling students to demonstrate their learning) are also important. Additionally, guidance and corrective feedback during performance of the task is also critical for efficient learning.

This paper describes efforts to bridge gaps between evidence-based instructional design and game-based technology. Bridging the gaps could lead to novel, game-based training solutions that leverage the current advantages of games (portable, simulation-based practice) but also enable the as-yet-unrealized advantages of reducing cost and improving training effectiveness. To attempt to bridge these gaps, we are developing methods and tools to enable training developers and instructors to "bring the classroom into the game." There are two complementary thrusts:

1. Defining content-authoring guidelines and design principles that place relatively *greater emphasis on iterative development of skills and capabilities via direct instruction, demonstration, and guided practice*, while also maintaining elements of gaming experience, especially interactivity and positive feedback, which may be important for many learners.

2. Designing and developing *tools to organize and constrain student experience to follow proven instructional design methods and fill functional gaps*. Instructional guidelines alone are likely insufficient to make a substantial impact on games for learning. To directly support evidence-based instructional design, we are developing tools that support the principles. The tools target non-technical users such as training developers and schoolhouse instructors. The tools are designed to help the instructor harness their domain knowledge, via interaction with game-based practice environments, to create instructional content to supplement practice.

The paper summarizes progress, examples, and challenges in these two thrust areas. We first outline "principles of game-based instruction" we have developed to guide the design and development of the tools. We then introduce several tools we have implemented, designed to support the principles. The tools are components within a larger content-development pipeline for instructional games we call the Game-based Instruction Tool Suite (GBITS). We illustrate the role and benefits of the tools via examples of content developed for an instructional game. The game focuses on Course of Action Analysis (COAA) within the US Army's Military Decision Making Process (MDMP).

GAME-BASED INSTRUCTION

We are seeking to complement instructional design processes with proven user-experience principles of games. We begin by outlining two aspects of games often compelling for sustained gameplay but not typically emphasized in traditional computer-based education/training.

Game-design Principles

What makes games fun? What aspects of games lead to more sustained attention and engagement than is typical of everyday life? Although rigorously-controlled, empirical evidence of the allure of games is not yet firmly established, theoretical analysis and case studies suggest a recurring feature in successful games couples fast-paced, interactive challenges with positive feedback cycles at multiple time scales (Chatfield, 2010; Koster, 2004; McGonigal, 2011).

Interactivity differentiates games from other forms of fictional entertainment, such as books or movie. A fundamental principle of game design is to give the player interesting choices. Good choices require tradeoffs and have long term consequences for game experience. These choices can range from what to say to a virtual character, to which equipment to purchase to how to use skill points to improve a character. The choices are core to interactive user experience, allowing the player to participate in the experience and "own" decisions and their consequences. This participation makes the player part of the story and game environment, which provides a level of immersion beyond most non-interactive media.

Positive Feedback Cycles: Good games deliver frequent rewards. Many games are structured to deliver three levels of rewards at three time scales. Minor rewards happen roughly every minute (kill a monster and get a little loot). Players earn more substantial rewards every 10 to 15 minutes (finish a quest and get a new sword). Still larger rewards are earned every 1 to 3 hours (go up a level and get a new skill). While these examples describe rewards in a role playing game, similar reward cycles are surprisingly prevalent across many different game genres and settings. Rewards at this approximate pacing have proven to keep a player engaged and moving forward in the game, presenting a fun level of challenge in the moment while offering even greater benefits if the challenges are met. Increasingly, via Internet-multiplayer capabilities and social media, another source of reward is social status (among game players) gained thru accomplishment within the game (owning a high score or having been the slayer of some rare monster).

Thus, an engaging game provides the player with specific tasks that require active participation and that provides not only second-to-second- or minute-to-minute positive feedback on task accomplishment, but defines medium- and long-term player goals (completing a level, freeing the princess) against which the player can track his or her progress. As others have also noted, the ways games progressively increase challenge in response to player improvement is directly comparable to the "zone of proximal development" hypothesized to maximize learning efficiency (Vygotsky, 1978).

Other elements, such as virtual immersion, are important in the success of many games; however, a first-person and/or hyper-realistic rendering of game experience does not appear to be a necessary condition for successful player engagement. Obvious examples are real-time strategy games (Star Craft, Civilization, Command and Conquer). These games are highly successful even though they give the player a "god's eye" view of the gameplay. Similarly, so-called "casual games" (Diner Dash, Farmville, etc.) capture player interest with comparatively rudimentary graphics.

A motivating hypothesis is that successfully capturing both interactivity and positive feedback in an instructional game will increase engagement and sustained learner participation in a training experience in comparison to standard computer-based training environments. However, a key open question is how to incorporate interactive and rewarding game-design elements within a learning environment that also makes efficient use of the learner's time. While McGonigal (2011) argues successful games are compelling learning environments, they are typically not focused on maximizing the speed of learning. Indeed efficient use of a game-player's time is generally considered undesirable from a commercial computer-game design perspective (where the goal is to sustain interest in the game as long as possible). Considering ways to use the learner's investment of time more efficiently leads us to best practices in instructional design: instructional design is focused on structuring instruction to deliver efficient learning experiences.

Instructional-Design Principles

In order to understand what's often missing in serious games, we first review one specific approach to instructional design. It provides some concrete requirements for "instructional games."

Guided Experiential Learning (GEL) is an example of a detailed approach to instructional design that

Table 1: The structure of GEL Lessons.

1) **Objectives**: actions, conditions and standards that must be achieved
2) **Reasons** for learning: advantages of learning and risks to the unit of a Soldier's failure to learn and transfer
3) **Overview**: knowledge models, analogies and content outline
4) **Conceptual knowledge**: concepts, processes and principles necessary to learn to perform a task with examples and analogies that support learning)
5) **Demonstration of procedure**: a clear "how to" description for all elements of a task
6) **Part- and whole-task practice** of procedures with corrective feedback
7) **Challenging, competency-based tests** that include reactions (trainee confidence and value for the learning) and learning (memory for conceptual knowledge and whole-task application skills for all procedures).

(Adapted from Clark and Feldon, 2008)

specifies "how to instruct" as well as "what to instruct" (Clark & Feldon, 2008; Clark et al., 2010). It has been used for a variety of business, academic, and military applications. GEL has significant empirical support and GEL-based instruction shows dramatic improvements in learning outcomes compared to other methods (e.g., Early, 2008; Velmahos et al., 2004).

GEL is a specific version of direct instruction. It requires the use of the previously-mentioned instructional principles that Merrill (Merrill, 2002, 2006) identified as the active ingredients of the most effective, evidence-based adult training systems currently in use. Merrill recommended these principles must be included in all courses and lessons regardless of delivery medium or training content, including serious games. As suggested earlier, providing novice-to-intermediate-level trainees with a field-based problem or an immersive situation alone also does not achieve individual or team learning (Kirschner, Sweller, & Clark, 2006).

In a GEL course, all trainees must receive: 1) realistic, field-based problems to solve; 2) analogies and examples that relate trainee's relevant prior knowledge to new learning; 3) clear and complete demonstrations of how to perform key tasks and solve authentic problems; 4) frequent practice opportunities during training to apply what is being learned (by performing tasks and solving problems) while receiving corrective feedback; and 5) application practice that includes "part tasks" (practicing small chunks of larger tasks) and also "whole tasks" (applying as much of what is learned as possible to solve complex problems representing challenges encountered in operational environments). These five training methods are based on Merrill's principles. Games are important (but not sufficient) for presentation of realistic problems, immersive demonstrations, frequent practice, and practicing part-task and whole-task skills.

Design Principles for Instructional Games

Our R&D attempts to integrate these instruction and game design principles. Figure 1 summarizes the gap and goal by comparing the experience of learning in a GEL-like training environment (bottom) to a typical serious game (top). With GEL, students proceed thru an interleaved pattern of instructional overview, demonstration, guided practice, and assessment. Related lessons are grouped together. Progressing to the next level requires demonstrating skill in practice.

During practice, guidance and feedback are delivered when appropriate, resulting in a dynamic inter-leaving that reinforces the knowledge and skills with which the student is struggling. The student spends large amounts of time in practice, but under conditions in which the practice experience targets specific tasks and expectations.

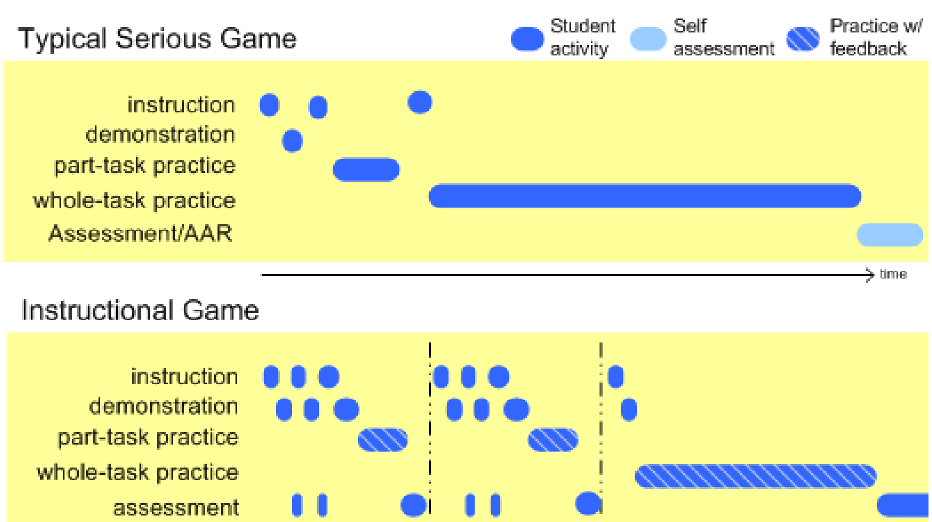

Figure 1: Game-based instruction in comparison to typical serious games.

In contrast, in a common serious games approach, there is often some initial (typically brief) “introduction” to the domain and then the student moves into practice. During practice, specific learning goals may or may not be explicit and there is typically little direct feedback to support learning. Self-assessment may be encouraged via an after-action review (AAR) process, but serious games often lack formal assessment. Thus, there is no assurance the student has learned the material and is ready for the next lesson/scenario. One would expect that for an equivalent investment in training time, the student in GEL environment may have been exposed to fewer total concepts, but one would have confidence that the student had more fully learned any introduced concepts.

For an instructional game, we hypothesize designing for learning requires less emphasis on the game medium for entertainment and engagement and proportionally greater emphasis on iterative development of skills and capabilities via direct instruction, guided practice, and on-going assessment. The resulting principles of game-based instructional design are:

Lessons provide the basic atomic unit of design. All learner experience is organized into these “atoms” of instruction. Each lesson lasts roughly 5-15 minutes. Every lesson includes active (and interactive) experiences for the learner, although reading text and multimedia presentation will also necessarily be part of an individual lesson. Within each lesson, a student should be able to earn reward(s), which can be as simple as completing the lesson or practice.

GEL prescribes a general structure for a lesson (Table 1). Each lesson follows a template (from a family of templates) that defines how information is organized and delivered within the lesson. Consistent organization is useful for authoring but also for the learner. As lessons become familiar, the learner will focus less on the “interface” and more on the content of the lessons. At a high level, GBITS organizes lessons into four components or phases:

- **Overview** (including objectives, reasons for learning, overview, & conceptual knowledge).
- **Demonstration:** Following GEL, our approach focuses on demonstrating “one good way to succeed” rather than detailing (within a single lesson) multiple variations of a procedure.
- **Practice:** Practice includes both opportunities to demonstrate in a non-situated context, as well offering simulation-based (situated) practice. Practice is varied from lesson to lesson and includes performative, explanatory, and reflective (assessment of others) practice. Practice, especially part-task practice, should include explicit guidance and corrective feedback.
- **Testing:** Testing ensures students have mastered the lesson. We have separately developed a situated assessment tool (see Future Work). We plan to integrate this tool with the prototypes we describe in this paper but have not yet done so.

As we discuss further below, we have implemented a Lesson Designer in which an author (instructor or content developer) selects and instantiates these templates to organize instruction at the lesson level.

Modules: Lessons are grouped into modules. A module is a self-contained sequence of lessons that deliver complete instruction over some specific topic. Completion of a module should result in the learner being able to articulate some new conceptual knowledge or to be able to perform some new procedure. Modules should be designed so that each module takes about an hour to complete, with the idea that modules are short enough that they can generally be completed in a single sitting. To support positive feedback cycles, successful completion of a module will offer some reward to the learner and performance within the module may have impact on later modules.

Each module culminates in an intermediate-task exercise. Lessons within a module need not be cumulative (e.g., distinct part tasks) but the final lesson in each module should include an intermediate task exercise that requires the learner complete a more complex task that integrates multiple part-task components within the module. Although the designer should have flexibility to make different choices, in general the “reward” for completing a module will often be tied to successful completion of the intermediate-task exercise.

Campaigns: Modules are presented as a “Campaign” sequence. Overall progress within the learning application is measured and made apparent. Techniques for overall reward will include scoring, relative rankings among learners, the ability to modify (“mod”) the environment and/or achieving unique capabilities within it, badges of distinction/merit, etc. The learner receives campaign-level rewards as they are earned and are evident to learners as they progress thru the campaign. For example, an explicit goal/reward could be to “for completion of Modules 1-3 with > an 85% performance score, you will receive a ‘campaign badge’ that will be displayed next to your name and overall score.”

SUPPORTING GAME-BASED INSTRUCTION

These game-design principles should be adapted within serious game development pipelines for creating new instructional games. In fact, our future plans include integrating the principles within some planned serious games, pending effectiveness outcomes.

However, we believe the best approach to gaining traction in the development of instructional games more generally and immediately will be to provide tools that reflect and support the principles directly, enabling the development or supplementation of practice environments with improved overall instruction. We are developing a Game-based Instruction Tool Suite (GBITS) to support instructors and training content developers to apply these instructional game design principles. The overall scope of the tool suite is beyond the scope of a conference paper. Instead, we illustrate how we are making the principles operational by focusing on three core needs and the implemented capabilities GBITS provides to address those needs:

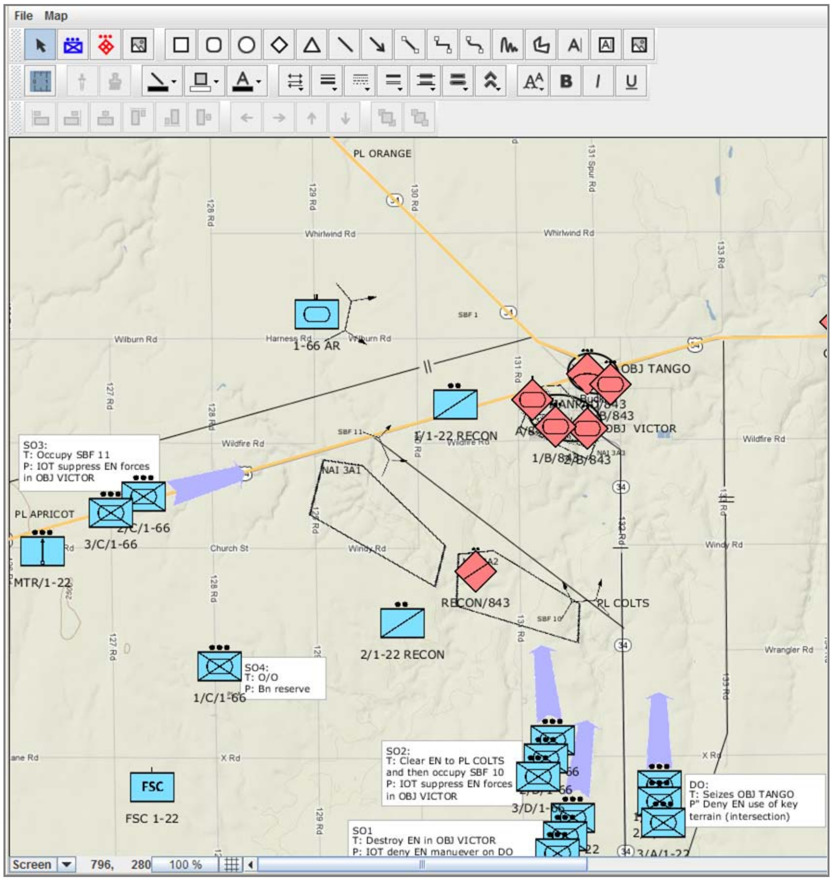

Figure 2: COAA Engine Screenshot from *Introduction to COAA/Wargaming.*

1. *Constructing good lesson content*: For instructors and content developers not deeply versed in instructional design, a **Lesson Designer** guides the content developer in creating content for lessons, including interactive demonstrations.

2. *Delivering instruction within a game environment*: Training content developers may wish to supplement an existing practice environment, but lack the technical (programming) skills to do so. An **Instructional Game State Manager** enables content developers to embed new content, new lessons, and new modules without having to make source code changes or recompile the instructional game.

3. *Guiding learner practice*: Effective practice requires guidance and feedback. A run-time student **Monitor and Feedback** system delivers guidance and feedback based on the specific actions of individual students and uses different strategies for part-task vs. intermediate-task practice exercise.

To illustrate these capabilities concretely, we introduce an instructional game application currently being developed with GBITS. The prototype instructional game is called *Introduction to Course of Action Analysis/Wargaming* or ICW. ICW integrates a game-based practice environment (the COAA Engine) and GBITS run-time components. Figure 2[1] shows a screenshot of a course of action in the COAA Engine. Content for the instructional game is authored with GBITS as well.

ICW's game-based practice environment focuses on Course of Action Analysis (or "Wargaming") within the Army's Military Decision Making Process (MDMP). The COAA task involves step-by-step evaluation of candidate courses of action (COAs) to find gaps, to recognize the need for synchronization

---

[1] All examples and screenshots have been constructed by the authors solely for illustration purposes. They are not examples of curriculum materials used by the US Army.

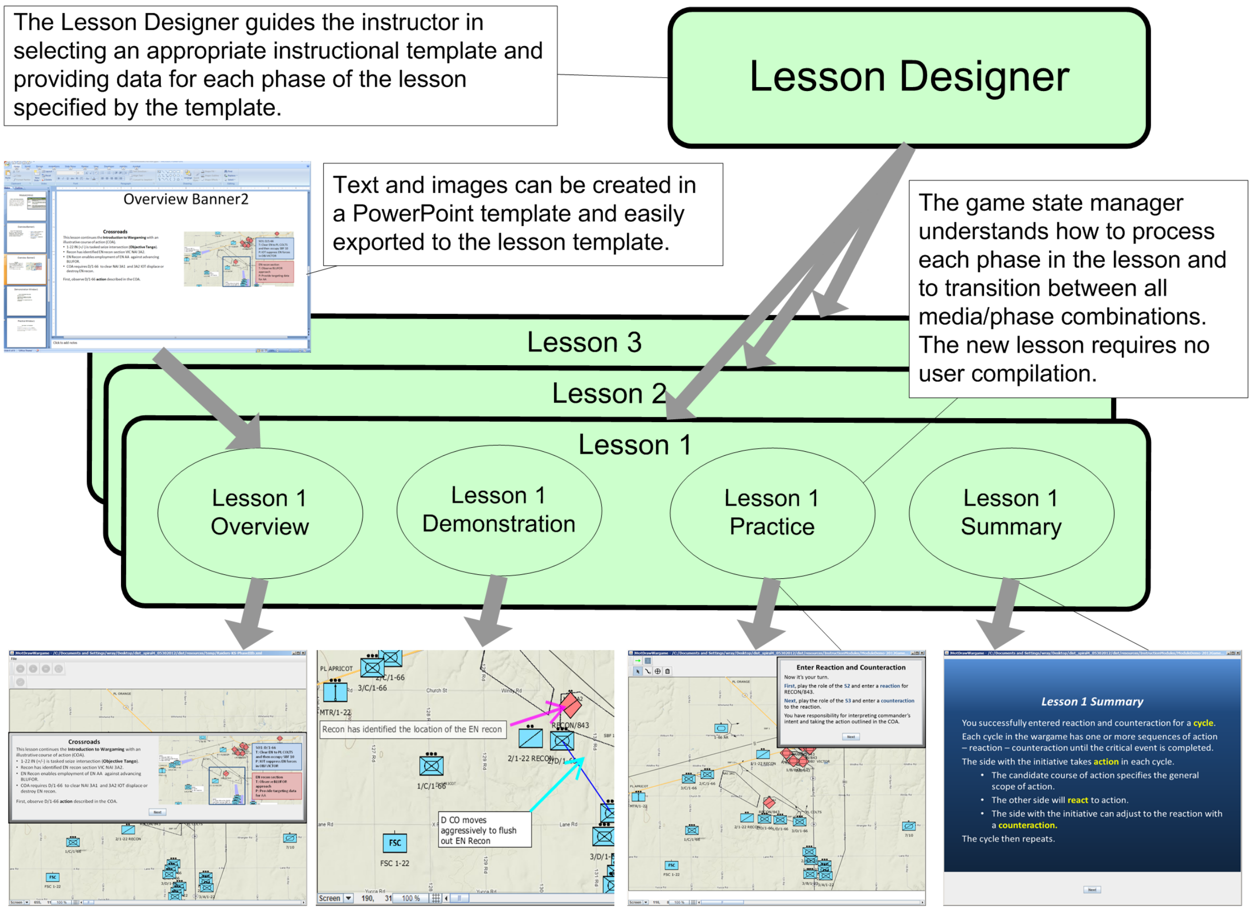

Figure 3: Overview of the GBITS Lesson Designer and Instructional Game State Manager.

(combat power), to coordinate unit actions with those of other units, and to improve understanding of the plan. COA Analysis follows a well-defined procedure documented in Army Field Manuals (e.g., FM 5-0).

At the core of COAA is a wargaming process in which the side with the initiative takes actions defined in a candidate course of action (moving into position, directing fires, etc.), the opposing side reacts to the action(s), and then the initial side responds to the reaction (counteractions). This action-reaction-counteraction process then repeats until the COA is determined to be infeasible or measures of effectiveness of the COA can be assessed.

The COAA Engine supports student practice of course of action analysis. The student can create a rough initial plan ("COA sketch") as shown in the figure. The student then executes the wargaming process, moving, directing fires, and noting the effects of tactical actions, such as combat outcomes and logistical calculations such as fuel usage. Students playing different warfighting function roles within COAA currently play the game together, with one student "driving" the interface. We designed ICW as a single-user instructional game, so this limitation of the COAA Engine presents no obstacle for use in ICW.

We have demonstrated ICW and GBITS at several schoolhouses presenting the US Army Captains Career Course (C3) curriculum, a likely user community for the finished ICW/GBITS tool. The demonstrations and interactions with these groups informed requirements for the ICW instructional game and helped us prioritize tools and features within GBITS. In particular, we have benefited significantly from, and would like to acknowledge, the contribution of personnel at the School for Advanced Leadership and Tactics (SALT). SALT is currently providing an informal evaluation of the tool, which will inform us of the potential for classroom use.

Constructing Effective Lesson Content

The GBITS Lesson Designer provides templates that enable a content developer to structure instruction. Figure 3 illustrates the concept of the Lesson Designer. There are different templates for different kinds of learning (concepts vs. skills). The Lesson Designer guides the content developer by identifying “slots” that authors will need to fill with specific types of content.

Lesson templates define overview, demonstration, practice and feedback, and assessment phases, as outlined previously. Content developers can specify the medium of content delivery for a phase within a lesson in addition to the content itself. For example, a simulation recording can be used for reflective practice or a video recording can be used for a demonstration. The outputs of the lesson designer are an instantiated, executable game-state (see below) and placeholder content for each content element specified during lesson design. The author can then fill individual frames of the lessons using other authoring tools.

Many lesson slots can be filled with recordings of previous games played within the COAA Engine. GBITS builds on a technical capability which records (via logs) all student/user experience in the simulation environment. Simulation2Instruction a content developer to play back a captured log for the purposes of review, formal after-action review, or to create a stand-alone or instructor-guided demonstration for others, etc (Wray & Munro, 2012). GBITS includes S2I components that allow a content developer to add graphical annotations to logs in order to create demonstrations and a run-time control component that interfaces to the game engine to playback recordings. By making it simple to capture and edit recordings from the simulation, content developers can readily express complex domain concepts and skills without having any programming expertise.

The Lesson Designer currently produces a PowerPoint template for text and image content. This template allows the content developer to readily map individual overlays to the specific content frames defined in the Lesson Designer. Although we initially viewed the PPT template as a stopgap capability, feedback from demonstrations and use of GBITS by instructors has suggested they prefer the familiarity of PowerPoint to a new authoring tool and find the process of transferring content from the PowerPoint template to game data comfortably straightforward.

Delivering Instruction within a Game Environment

A game state manager is a software approach used by computer-game developers to manage player states during a game and to effect transitions between them. The game state manager tracks an on-going game's state space (using a stack), where each state is equivalent to a game level or a mini-game. Game states are pushed and popped onto the game state stack; the state at the top of the stack is the one in which the student is actively interacting. As software, commercial game states are organized into a tree structure, where those individual states that can be instantiated are the leaf nodes of the tree. Typically, each new game state is hand-programmed and requires source code modification that requires compilation.

We have developed a game-state manager for instructional delivery with specialized functionality for “game” states associated with the phases of a lesson (overview, demonstration, practice, assessment). Importantly, because we anticipate the authors of instructional games will often not have programming skills, we needed to modify the game state concept both to support instructional modes but also to not require source code level changes and compilation.

We addressed this challenge by creating leaf-nodes templates in the game state hierarchy. The templates can be instantiated with data (lesson content). Each leaf node is parameterized and used to generate many gameplay levels with a similar structure, but unique content. As illustrated in Figure 3, the Lesson Designer outputs instantiations of the lesson template with content data specified by the lesson author. The result is a newly generated game state (the phase bubbles in Lesson 1 in the figure), representing a unique lesson (such as a specific overview state or a unique practice state). Using this approach, a content developer need not re-compile to create a new lesson game state.

At run-time, the game state manager's stack of states delivers structured lessons (where a full lesson is composed of multiple game states). The structured lesson template (such as Overview-Demonstration-Practice-Summary) describes how the game state manager should move through the game states that make up the structured lesson. It also specifies any student-interaction parameters. For example, the demonstration game state uses the Annotated Playback media component to enable the student to review a previously captured recording. However, because the game state is “demonstration,” it also knows to enable demonstration-specific annotations and ensures that the student cannot “skip” the demonstration. As a result, transition to the subsequent game state follows only after completion of the demonstration. There are two advantages to including the student interaction parameters in the lesson template. First, because these details are not embedded within the Annotated Playback media component, it remains general purpose and can be used in other contexts (such as Playback2Game) without modification. Second, the lesson author can simply focus on content for the lesson, rather than how it is executed at run-time. The

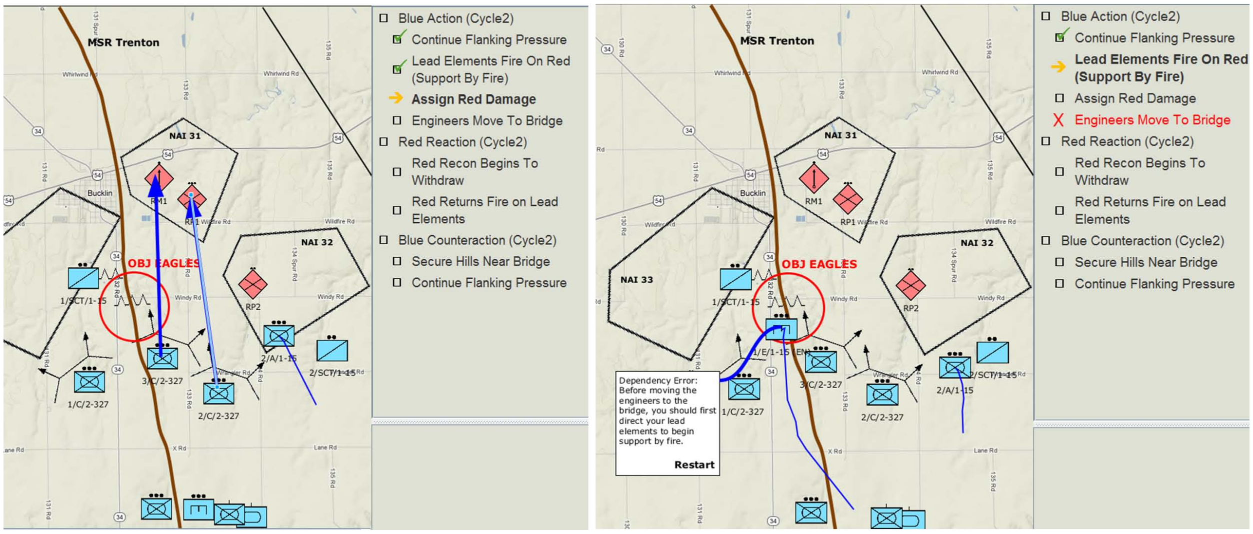

Figure 4: Examples of Guidance and Feedback during Part-task Practice.

lesson author selects the desired lesson template and the instructional game state manager controls structural execution details automatically.

Student Guidance and Feedback

As students practice, they need feedback in response to their actions and they also will need explicit guidance to address observed or perceived learning needs. Because such feedback and guidance are not typically part of entertainment games, these capabilities are novel requirements for an instructional game development toolkit.

GBITS is focusing on delivering feedback and guidance without introducing significant technology requirements or content creation burdens. The goal is to provide guidance during practice and deliver timely feedback, focused on underlying concepts, with positive messages as well as correctives (Shute, 2008).

The current design and implementation delivers guidance and feedback differently based on the type of practice. During initial (part-task) practice, the execution of individual steps during initial learning is scaffolded by an explicit representation of the procedure. In the examples in Figure 4, the procedure outlined on the right is a step-by-step description of the specific steps the student should execute during the current wargaming cycle.

As steps are completed, they are marked with a checkmark, which is simple mechanism of positive feedback to indicate progress and help the student have confidence the procedure is being executed correctly. The current, active step(s) is marked as well, evident in the orange arrow in the diagram.

Because each part-task lesson covers a few (5-7) individual steps, when the system observes an error, the student receives information about the error and practice restarts. In the example on the right side of Figure 4, the Blue forces in the support-by-fire positions should begin suppressive fires on the red units before the engineer platoon moves up toward the obstacle blocking the road. Because the student moved the engineers first, an error message is displayed and the student must restart the practice. Re-starting helps the student to focus on learning the “one good way” for application (as recommended by GEL), without needing to learn to recover from errors simultaneously.

For intermediate- and whole-task practice, the procedure or process is not represented by default, although the student can choose to activate a guidance window comparable to one used in part-task practice (within the context of the game, this action may have repercussions such as limiting the maximum score for the lesson, although this scoring feature is not implemented). Scaffolds such as pro-active hinting provide less explicit guidance. We are exploring the use of graphical highlighting within the game environment to deliver pro-active hinting. For example, in the scenario illustrated on the left side of Figure 4, if the student needed to enter combat results for the red units and had not (after some reasonable delay), the feedback system could place heavy lines around the edges of the red units under fire, suggesting action was needed.

Additionally, in intermediate-task practice, when an error is committed, the student has the option of attempting to recover from the error or re-starting. The scenario author has the option of providing recovery suggestions based on the type of error that has occurred. A summary of the lesson, presented at its conclusion, outlines the specific steps the student took and any recommendations for review or additional study base on the student's execution of the lesson.

We have re-purposed and refined existing student modeling and tailoring technologies for GBITS to realize the action monitoring and error recognition components of this capability. In the native capability, tailoring occurs "behind the scenes," analogous to a film director's actions, and coordinates scaffolding, fading, and challenges for the student (Wray et al., 2009). An active monitoring process supports on-line interpretation of student actions. The approach employs comparatively simple, declaratively-represented student models that define "bounds" on appropriate behavior (Mitrovic & Ohlsson, 1999). A student "error" is recognized when one of these boundary constraints is violated. The original language employs a graphical representation appropriate for non-programmers.

We have extended the original modeling language to enable the author to define the steps in a procedure and to define potential errors in the procedure. This declarative modeling language has a notion of three levels of abstraction: domain constraints (e.g., fratricide is never appropriate), scenario constraints (e.g., movement over a bridge is not allowed unless the bridge is clear), and practice constraints (Unit 3/C/2-327 should be located near SBF11 by the end of this cycle). In contrast to most expert modeling approaches, this approach hypothesizes it will be simpler and faster for an author to express constraints about a particular scenario or practice experience than to create general constraints applicable across the domain.

Constraints are parsed into rules that feed a rule-based system that monitors student action. The rule-based system then compares student action to the constraints. When a constraint is violated, the monitor constructs an error message to present to the student. To-date, these messages are pre-authored, but we expect over time the feedback system will be able to (at least partially) synthesize error messages. The monitor also determines, based on the practice mode (part-task or intermediate-task) what system action is needed. The game state manager simplifies the "return to start" action because, at the implementation level, this simply requires "popping" the current practice state and "pushing" a copy of the initialized practice state.

FUTURE WORK

Having successfully implemented GBITS and the ICW instructional game prototype using GBITS, we are currently planning several parallel lines of future research and development:

Effectiveness Evaluation: While the experimental approach is still under development, ARI plans to conduct user testing and effectiveness evaluations with Soldiers. Specifically, the GBITS/ICW tool will be compared to a control (e.g., upfront instruction followed by ICW practice and AAR/post-training feedback) to assess learning and training effectiveness/efficiency. Potential user populations will be targeted (students at SALT other C3 courses, USMA, ROTC). In addition to learning, related constructs like interactivity, immersion, engagement, and transfer may be evaluated.

Bite-sized games: One capability made feasible by the current tools within the MDMP domain is the notion of students and instructors developing individual lessons with practice scenarios, targeted for a single topic or very short duration. For example, a 1-2 hour "self study" or "homework" lesson could easily be constructed. These "bite-sized" games would not include the GBITS module and campaign features. Importantly, students would have available GBITS capabilities, including the ability to capture and show student work via annotated demonstrations.

Model and Feedback Authoring Tool: All of the elements in the existing GBITS instantiation are authorable (data driven); however, we have not yet built and evaluated an authoring tool for the guidance and feedback system (authors today would create models directly in XML). We are currently starting a new effort that will create a graphical editor for the student and expert models.

Integration of Assessment Authoring: GBITS recording and playback provides not only the ability to create demonstrations for instruction, but also the ability to capture decision points within a scenario which can then be used for assessment (and reflective practice). We have developed an assessment authoring tool for another domain that enables an instructor to construct a question (or series of questions) that are presented in conjunction with captured recordings (Wray & Munro, 2012). We plan to integrate this assessment capability into GBITS and extend the game state manager to load a recording, and to deliver and score a "test" based on the authored content.

Bridging the gulf between effective games and evidence-based instructional design represents a reasonable path toward portable, compelling, simulation-based instruction and practice for improved training outcomes. In this paper, we have described our rationales and hypotheses for an approach to bridge the gap and outlined an implemented capability for authoring content for instructional games that offers a concrete first step toward the vision.

ACKNOWLEDGEMENTS

This work is supported by the Army Research Institute, contract W91WAW-10-C-0044. The view, opinions, and/or findings contained in the report (paper) are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision. We acknowledge and thank Dr. Stephen Goldberg, Dr. Paula Durlach, Dr. Richard Clark, COL (R) James Zanol, LTC (R) Mark Pires, Keith Knudsen, Dr. Jeremiah Folsom-Kovarik, Laura Hamel, Jonathan Carbone, and Joshua Haley, who have each contributed to the conceptual and/or software development of the tool suite and/or the content of the instructional game.

REFERENCES

Chatfield, T. (2010). *Fun Inc.: Why Gaming Will Dominate the Twenty-First Century*: Pegasus.

Clark, R. E. (2007). Learning from serious games? Arguments, evidence and research suggestions. *Educational Technology, May-June*, 56-59.

Clark, R. E., & Feldon, D. (2008). *GEL, Adaptable Expertise and Transfer of Training*. Pasadena: Center for Cognitive Technology, USC.

Clark, R. E., Yates, K., Early, S., & Moulton, K. (2010). An Analysis of the Failure of Electronic Media and Discovery-based learning: Evidence for the performance benefits of Guided Training Methods. In K. H. Silber & R. Foshay (Eds.), *Handbook of Training and Improving Workplace Performance* (Vol. I: Instructional Design and Training Delivery, pp. 263-329). New York: Wiley and Sons.

Early, D. (2008). *The effect of guidance on learning in a web-based, media-enriched learning environment*. Unpublished Doctoral Dissertation, University of Southern California, Pasadena.

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist, 41*(2), 75-86.

Koster, R. (2004). *A Theory of Fun for Game Design*: Paraglyph Press.

McGonigal, J. (2011). *Reality Is Broken: Why Games Make Us Better and How They Can Change the World*: Penguin Press.

Merrill, M. D. (2002). First principles of instruction. *Educational Technology Research and Development, 50*(3), 43-59.

Merrill, M. D. (2006). Hypothesized Performance on Complex Tasks as a Function of Scaled Instructional Strategies. In J. Elen & R. E. Clark (Eds.), *Handling Complexity in Learning Environments: Research and Theory* (pp. 265-282). Oxford: Elsevier Science Limited.

Mitrovic, A., & Ohlsson, S. (1999). Evaluation of a constraint-based tutor for a database language. *International Journal of Artificial Intelligence in Education, 10*(3-4), 238-250.

O'Neil, H. F., Wainess, R., & Baker, E. (2005). Classification of learning outcomes; Evidence from the games literature. *The Curriculum Journal, 16*(4), 455-474.

Shute, V. J. (2008). Focus on Formative Feedback. *Review of Educational Research, 78*(1), 153-189.

Velmahos, G. C., Toutouzas, K. G., Sillin, L. F., Chan, L., Clark, R. E., Theodorou, D., et al. (2004). Cognitive task analysis for teaching technical skills in an inanimate surgical skills laboratory. *The American Journal of Surgery, 18*, 114-119.

Vygotsky, L. S. (1978). *Mind and society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Wray, R. E., Lane, H. C., Stensrud, B., Core, M., Hamel, L., & Forbell, E. (2009). *Pedagogical experience manipulation for cultural learning*. Paper presented at the Workshop on Culturally-Aware Tutoring Systems at the AI in Education Conference

Wray, R. E., & Munro, A. (2012). *Simulation2Instruction: Using Simulation In All Phases of Instruction*. Paper to be presented at the 2012 Interservice/Industry Training, Simulation, and Education Conference.