The Importance of Narrative as an Affective Instructional Strategy

Mark O. Riedl[1] and R. Michael Young[2]
[1] School of Interactive Computing, Georgia Institute of Technology; [2] Department of Computer Science, North Carolina State University

INTRODUCTION

Virtual simulations, computer games, and other game-like virtual worlds are progressively being adopted for "serious" purposes, such as training and education. Computer-based training systems and games share many similarities: both involve a progression of skill-based activities of increasing complexity and difficulty. The learner is expected, over the duration of the game, to master a set of skills. Skill mastery is one of the fundamental principles behind the success of computer games; indeed, this mastery of the game is a fundamental aspect of having fun in computer games (Koster, 2005). It is easy to see the appeal of games and game-like virtual environments for the purposes of training and education. The skill-based progression typically used in computer games and the desire for players to achieve mastery of particular game-relevant skills can be mapped to educational outcomes and pedagogy.

In computer games, it is not always enough to have a progression of skill-based activities. Many game genres use fictional context to reinforce the immersion within the game world and to motivate the skill-based activities. These fictional contexts answer the question "why am I, as the player, engaging in a particular activity?" The fictional context may further induce an affective response from the player: dramatic tension over how events are unfolding, strong positive or negative feelings toward virtual characters, or suspense over what might happen next. In many games skill-based activities are often structured through narrative, mission, quest, or scenario. These narratives, missions, quests, and scenarios can manifest as backstory or as non-skill-based interactive game play such as moving through a virtual environment and interacting with virtual, non-player characters. Without narrative context, educational and training games may be perceived as a progression of drills without a purpose other than mastery itself. The use of fictional context is one possible way to more fully engage learners and motivate them to partake in skill-based progressions. In this chapter, we use the term *narrative* to mean a predetermined, temporally ordered set of actions or events. To that end, missions, quests, and scenarios are forms of narrative because they involve a temporally ordered sequence of events.

Unfortunately, the similarities between computer games and game-like learning environments is sometimes only skin-deep when it comes to the use of narrative contexts to motivate and engage. In particular, it is often sufficient for entertainment-based computer games to rely on linear narrative sequences and linear skill-progressions. Typically, important design consideration in entertainment-based games that all players have the same experience. However, intelligent educational and training technologies require the ability to adapt to deliver the right educational content to the right learner at the right time. That is, variability in skill ability and rate of mastery must be accounted for because a skill-progression that is too slow or too fast may result in the learner abandonment. Research in *Intelligent Tutoring Systems* (ITS) has made significant gains in understanding how to model learner abilities and deliver the right problem to individual learners in the right sequence. Intelligent tutoring systems can be broadly described as implementing two nested processes (VanLehn, 2006). The inner process is one of recognition of learner difficulties when solving a problem and selection of the most effective means of feedback and remediation. The outer process is one of selecting the best problem for the learner to work on next. The next problem the learner works on may come from a library of problems or may be automatically generated based on the learner's immediate needs and abilities. If an educational or training game must dynamically select skill-based events in order to maximize player learning potential, then a single, linear narrative may no longer be sufficient and artificial intelligence—in the form of automated story generation—can be brought to bear to construct new narratives that motivate and contextualize the learner-customized skill progression.

In this chapter, we describe two systems that use computational models of narrative generation to create game play experiences that directly support the learning process. One system, Annie, uses a model of a game's task domain to track players' knowledge and to alter a player's challenges as they demonstrate mastery of or misconceptions around a particular skill. Another system, Game Tailor, automatically generates a sequence of skill challenges through which a player will progress and then creates a customized storyline in which the challenges are naturally embedded. These two systems demonstrate the effective role that explicit narrative models can play in the generation of tailored learning experiences within games.

BACKGROUND AND RELATED RESEARCH

A *narrative* is a predetermined, temporally ordered set of actions or events. Actions can be executed immediately in the virtual environment, whereas an event is a discrete period of time in which the actions of a number of characters are thematically or semantically related. An example of an event in a computer game is "player fights boss opponent" in a role-playing game or "user deletes a malicious virus program from memory" in a game-based virtual simulation of a computer. Each event may consist of a number of actions such as combat attacks, discourse acts, manipulation of file permissions, etc. The simplest narrative is a single, linear sequence. However, temporal ordering can support more sophisticated narrative structures, such as parallel actions and events. In this section, we provide an overview of the ways that story is treated in computer games, discuss artificial intelligence techniques for generating and adapting narrative structures, and compare narrative adaptation with intelligent tutoring systems.

Narrative in Computer Games

In computer games, narrative is used to motivate player behavior and establish the context for why the player is to perform certain activities. The narrative acts as an explanation, or context, for the activities the player is about to perform. The events making up a mission, quest, or scenario may be categorized as *skill-based* or *non-skill-based*. Skill-based events are periods of game play that require the player to attempt to perform a skill that is valued by the game designer or instructor. For entertainment-based computer games, skills may include finding and collecting items, solving puzzles, navigating through mazes, combat with opponents, etc. Non-skill-based events are periods of game play that do not require a skill valued by the game designer or instructor. Additionally, the game may involve periods of non-interactivity where the player is watching a cut-scene or in which the player's avatar is temporarily controlled by a script. Game designers use non-skill-based periods of play and non-interactive periods to advance a story, to create context for the next period of skill-based play, and to motivate the player to achieve certain goals.

One aspiration of game design is to encourage players to move along an intentionally circuitous route to incorporate experiences that positively affect the player's enjoyment. Game designers refer to the circuitous route as the *golden path* and the most direct route as the *spine* (Bateman, 2007). Most games have a linear spine, providing little or no variation in which events are involved in completing the game. In contrast, the golden path contains additional, non-mandatory game elements that enhance other aspects of the player's experience. Perhaps equally important, however, the golden path enhances the player's sense of agency over the events that occur during game play, and helps disguise the essentially static structure of the underlying spine.

One can visualize the structure of many computer games as a "string of pearls" where the pearls are periods interactive game play (often referred to as levels or maps) and the string that holds all of the pearls together and space the pearls out is non-interactive narrative. From an implementation perspective, pearls are often implemented as sandbox environments—bounded simulations where possible actions are dictated by the underlying rules and physics of the game. Pearls may be skill-based or non-skill-based. The player engaged with a pearl until he or she triggers the conditions necessary to exit the pearl. This may involve reaching the end of a level, performing a particular action, or successfully demonstrating one or more skills. The narrative string that follows the most recently played pearl then sets up the context for the next successive pearl. Typically, successive pearls will require the player to demonstrate skills under more challenging circumstances. We refer to this as *skills progression*.

Story Generation, Interactive Narrative, and Intelligent Tutoring

Computer games for education and training differ from entertainment-based games in that game play by learners may be mandatory if the game is part of an educational curriculum or part of a training regimen, or required for skill or knowledge assessment. When this is the case, a string of pearls design will be insufficient because learners with different abilities or rates of mastery may require that learners experience different skill progressions. Variability in learner ability and rate of mastery may mean a fixed skills progression may be too difficult or too easy to promote effective learning (Vygotsky, 1978).

Artificial intelligence can be used to model learner skills and mastery rate and to produce tailored skills progressions that directly address learners' needs and abilities. This idea is not new; intelligent tutoring system researchers have sought to tailor learning environments to individuals. vanLehn characterizes an ITS as a process involving two nested loops. The outer loop performs problem generation, creating or selecting the next problem based on information about the learner, including traits, learning goals and needs. The inner loop closely monitors every action the learner takes while performing the given task and uses this information to update a model of the learner and provide directed feedback. Zook *et al.* (2012) note that if an intelligent system can produce new narrative structures, then it may serve the purpose of problem generation while simultaneously contextualizing the learner's behaviors through narrative. Any performance-based feedback operating during skill-based events of the narrative can be thought of as equivalent to inner-loop remediation.

Automated story generation is the problem of automatically selecting a temporally ordered set of events that meet a set of criteria and can be told as a story. For story generation, there are two problems that one must address. The first is to computationally model narrative structure. The consensus among psychologists and computer scientists alike is that a narrative can be modeled as a semantic network of concepts (Trabasso, Secco, and van den Broek, 1984; Graesser, Lang, and Roberts, 1991, Young, 1999). Nearly all cognitively inspired representations of narrative rely on causal connections between story events. The second problem is to computationally model the narrative creation process and develop algorithms that implement the model. Approaches to automated story generation include simulation, planning, case-based reasoning, and natural language processing. The simulation approach (Meehan, 1976; Aylett *et al.*, 2005; Cavazza, Charles, and Mead, 2002) situates autonomous virtual agents in an environment and records their actions. One of the critiques of simulation approach is that coherent narrative sequences may not necessarily always emerge. To solve issues of narrative coherence, planning—the search for a sound and complete sequence of actions that achieves a goal situation—techniques have been developed that observe global structural patterns (Lebowitz, 1987, Porteous and Cavazza, 2009; Riedl and Young, 2010), employing cognitive models (Riedl and Young, 2010) or specialized heuristics and constraints (Porteous and Cavazza, 2009; Riedl, 2009). Case-based reasoning approaches to story generation reuse existing stories in new contexts (Turner 1994; Pérez y Pérez and Sharples, 2001; Gervás *et al.*, 2005; Riedl, 2010). The natural language processing (NLP) approach to story generation is to mine word tuples or sentences from blogs (Swanson and Gordon, 2008) or text corpora (McIntyre and Lapata, 2009).

Interactive narrative (also *interactive storytelling* or *interactive drama*) is a form of digital entertainment in which users create or influence a dramatic storyline through actions, either by assuming the role of a character in a fictional virtual world or by issuing commands to autonomous, virtual non-player characters. The simplest interactive narratives, such as Choose-Your-Own-Adventure books and hypermedia, do not require artificial intelligence. A branching story graph is a directed graph where nodes are events and arcs are annotated with actions that the player can choose that lead to different narrative continuations. Branching story graphs can be manually authored, or procedurally generated by a story generator. Riedl and Bulitko (2013) provide an overview of AI approaches to interactive narrative. Mott *et al*. (1999) observe that narrative can be a useful tool for framing educational problem-solving activities. Interactive narrative has been explored as means of guiding humans through educational experiences and training scenarios (*c.f.*, Rowe *et al.*, 2011; Riedl *et al.*, 2008; Magerko, Stensrud, and Holt, 2006; Johnson and Valente, 2009; Marsella, Johnson, and LaBore, 2000; Aylett *et al.*, 2005; Thomas and Young, 2010).

DISCUSSION

In the following sections, we describe two ways in which interactive narrative and story generation can support learners through remediation of misconceptions, generation of skills progressions, and contextualization of activity in the virtual world. First, we describe a system called Annie; Annie detects and addresses misconceptions about procedural knowledge. Because many computer games provide sandbox-style, exploratory environments for learning, players and learners typically have wide latitude to select actions and compose plans to achieve their in-game objectives. By design, these games provide many possible ways for a learner to navigate the task space. This presents an intelligent tutor with a challenge that Annie is intended to address: a game-based ITS must track the learner's plan and take action on-the-fly to remediate any misconceptions. Second, we describe Game Tailor, a system that addresses problem generation for serious games. Game Tailor determines the next skill in a skill progression that a learner should practice. Unlike tutoring systems that select the next new problem from a library, Game Tailor generates an entire skills progression at once and then generates a storyline that motivates all the problems the learner will work on in the sandboxes.

Annie: Leveraging Plan-Based Models of Narrative to Detect and Address Misconceptions

Exploratory environments provide students with freedom to choose different courses of action. This complicates the tutor's ability to know what the student it trying to do, which introduces uncertainty in knowing whether or not a student has a misconception about the domain. When the tutor decides a misconception exists, it is difficult to know when is the right time to provide support to remediate that misconception, as the student may have changed focus to a different task. As Van Joolingen, De Jong, and Dimitrakopoulou (2007) note, it is difficult to balance guidance with student exploration.

In our previous work on the Annie system (Thomas and Young, 2010; Thomas and Young, 2011), we have addressed these problems by leveraging a well-understood computational model of actions and the causal relationships between them used in automated planning. The style of action descriptions invented for the STRIPS system (Fikes and Nilsson, 1971) has continued to form the basis of much subsequent research in automated planning. Building on several distinct approaches to integrating automated planning with game domains (Mott and Lester, 2006; Mateas and Stern 2005; Cavazza, Charles and Mead, 2002; Riedl, Saretto and Young, 2003), the Annie system leverages a general plan-based knowledge representation intended both to characterize a game-based learning environment's task domain as well as the knowledge of the tasks held by a learner.

STRIPS-style plan representations characterize actions available in a task domain schematically, defining an action in terms of its act-type, a set of *preconditions* and a set of *effects*. Preconditions are logical terms that indicate just those conditions in the task domain that must be true in order for the action to execute correctly, while effects indicate all the ways that a task domain changes as a result of the successful execution of an action. As an example, consider a task domain within a game world focused on teaching users how remove malware from a PC. One action in this domain might be named deleteFile, corresponding to the action of deleting a file from the PC's hard drive. This action would have two parameters: one indicating the character or player initiator of the task and one naming the file to be deleted. Its preconditions would indicate that, before this action can be carried out, the file must exist and must not be in use. Further, the character performing the action must be limited to the player (*e.g.*, no non-player character in the game can delete files). The effects for deleteFile would indicate that once the action succeeds, the file will no longer exist.

To build the model for what the student knows about deleting files, Annie begins by automatically deriving a set of meta-conditions from the known features of the deleteFile operator. The simplest model of the student's knowledge of the operators in the domain would register whether the student *knows* that a term appears as a precondition or an effect of a given action. For instance, Annie can generate requirements that a student knows that a file being deleted must exist, that it cannot be in use at the time, and that once the deleteFile action is performed, the file will no longer exist.

This simple approach to model construction fails to capture the uncertain nature of student knowledge in an exploratory environment where the student's understanding of the world evolves gradually. To represent this uncertainty we employ a rough-grained five-valued scale (HighlyLikely, Likely, Neutral, Unlikely, HighlyUnlikely) to represent varying estimates of the likelihood that the student believes or knows about a particular facet of the domain, where "Neutral" is the default initial value.

To illustrate, in a game that teaches the processes involved in aerobic cellular respiration, Annie may observe a student behavior that implies that the student knows an effect of the Krebs cycle is the production of CO_2 waste but may have no information yet on whether the student knows another effect of the process is the production of H_2O. This could be represented in the student model by marking the hasEffect condition corresponding to CO_2 production of a particular action in the Krebs cycle as HighlyLikely, while the effect that produces H_2O is marked as a student belief with Neutral likelihood.

Like many ITSs, Annie's core tutorial reasoning is situated in a loop interleaving student and system-controlled actions. Each time an action is taken in the world, either by the student or the system, Annie updates its student model by consulting a library of general diagnostic templates. These templates encode domain-independent plan reasoning diagnostics such as cases where a student seems to be ignorant of a precondition of a particular action. For example, if a student attempts an action for which some of the preconditions are not satisfied, a rule in one of these diagnostic templates fires to update the student model by lowering its confidence that the student is aware of those preconditions.

Annie uses the updated student model in consulting a second domain-independent library containing remediation templates that can be used to generate scaffolding. For example, if the plan shows that a particular task must be performed for the student to make progress toward plan goals, and Annie notes particular gaps in the student model pertaining to that action (*e.g.*, student has an incorrect model of its effects), it will prompt the student about that action.

As mentioned above, execution loops are common in ITSs that often operate with nested loops, one iterating over problems or tasks and another, nested within it, operating over individual steps in the problem. As described here, Annie's loop is often focused on individual steps in a task. Unlike a concentric loop architecture, however, Annie is free to switch to a completely different higher-level task or problem as a student interleaves tasks within the game.

A potentially difficult paradox for Annie's design is that as the student progresses, Annie gains more and more information about the state of the student's knowledge, but has less and less time remaining to act on these inferences. In order to characterize how close a student is to achieving important goals or milestones within a game world, we leverage the planning-based representation of the game world's task domain to compute the game world's *plan space* – a directed graph that characterizes the space of all possible plans for achieving a given set of goals in a specific game world. Planning algorithms called plan-space planners (Kambahmpati, Knoblock and Yang, 1995) construct plan spaces as part of their search process when solving a planning problem. Because the proper sequencing of actions within a plan relies on valid student knowledge regarding the tasks involved, Annie can use the plan space it constructs to prioritize and sequence its strategies for guiding the student toward acquiring the requisite knowledge.

A plan-based representation can provide a language simultaneously describing learning content and game play. With automated planning techniques, we can ensure that the spine of the game is traversed, while encouraging the player to explore far beyond the small set of detours built into a golden path. Through planning, a widely varied golden landscape unfolds where individual users can explore a variety of experiences tailored to their particular educational and entertainment aspirations.

Recapitulating Game-Based Learning Through Planning

Gee (2003) described a rich set of learning principles evident in commercial games and Quintana *et al.* (2004) described a framework that identified many of the scaffolding techniques used in exploratory ITS research, but neither of these descriptions lends itself to a generative model. Each leaves it to the artistic spirit of game or tutorial designers to decide when, where, and how extensive the computational support should be. Annie, however, requires a generative model for game-based learner guidance. We have built such a model inspired by the descriptions of Gee and Quintana, providing the following capabilities:

1) Each learning principle is articulated through one or more plan-based templates to allow automatic generation of game play elements that embody that principle.

2) Generation is performed at run-time, allowing the game to dynamically adapt to the behaviors exhibited by the student.

3) Systems can measure or specify the frequency and extent to which learning principles are realized. In other words, the model provides researchers with a mechanism to freely vary the prevalence of one principle *vs*. another and measure the effects.

Nine of the 36 learning principles articulated by Gee were selected as initial candidates for testing this generative model. Three of these are described briefly here.

> *Overt telling is kept to a well-thought-out minimum, allowing ample opportunities for the learner to experiment and make discoveries.*

We use the term *remediation* to describe an action Annie inserts into the game environment to attempt to correct what it perceives to be a misapprehension on the part of the student. We can count the number of remediations applied for each student, the best-case, worst-case and average number of remediations required for each particular knowledge component, and the comparative frequency of stronger or weaker hints that correspond to different type of remediations. Across a broad range of students, these measurements can be used to characterize the difficulty of different parts of the game world and help pinpoint areas where more student guidance opportunities may be required.

Remediations are organized in such a way as to allow Annie to choose between successively more explicit modes of instruction. This builds on extensive ITS research into the optimal selection strategy between the frequently used guidance options of ‘Prompt’, ‘Hint’, ‘Teach’, or ‘Do’.

> *There are multiple ways to make progress or move ahead. This allows learners to make choices, rely on their own strengths and styles of learning and problem-solving, while also exploring alternative styles.*

Annie can quantify the number of distinct successful plans, the number of qualitatively different plans in the plan space, the number of actions that must be included in any successful plan, or even the ratio of the number of these critical actions to the mean total number of actions in successful plans.

Annie allows for extensive mining of the space of potential plans to reveal bottlenecks, potential for off-task activity, etc. in a way that could be much cheaper and more extensive than traditional game design play testing strategies.

> *The learner is given explicit information both on-demand and just-in-time, when the learner needs it or just at the point where the information can best be understood and used in practice.*

The timeliness of explicit information can be measured by the duration of the interval between when the information is provided and when it is needed. This can be compared and contrasted with the number of opportunities for on-demand information in the environment. For some students or groups of students Annie may want to vary how far in advance help can be provided based on projected memory persistence of those students. As post-hoc measurements, analysis of these properties over many students can be used to calibrate guidance within Annie.

Advantages of Plan-Based Game Design

Our intention with the development of the Annie system was to demonstrate that a nominal plan-based knowledge representation can lead to a computational framework that can automatically synthesize and adapt gameplay/teaching at an atomic level. In this work, we selected a set of learning principles and leveraged a plan-based design to realize these principles in arbitrary domains. Specifically, our knowledge representation synthetically generates game structures that implement these principles, requiring less time, and money, resulting in a shorter and cheaper development cycle. Because these structures are automatically generated, their instantiation can be shifted to run-time, so they can be tailored to the immediate and subtle learning needs of the individual rather than the statically defined and obvious extremes of an entire population. Finally, the rules governing how and when to change course are visible and modifiable, rather than entwined with tutorial algorithms. This enables the system to conform to externally specified metrics for particular applications.

The use of a plan-based knowledge representation breaks the game spine into interchangeable parts, allowing for dynamic synthesis of game progression while ensuring that the player eventually traverses segments of the spine nominated as particularly critical. Any fixed branching structure could be implemented through a plan-based representation by representing each critical action choice as a distinct operator with unique prerequisites and effects. But planning not only replicates the expressivity of existing game progression, it allows for a much wider variety of scaffolding techniques, partial-ordering of actions, and varied bindings of particular game elements and arbitrary number of repetitions or cycling through particular types of actions.

Game Tailor: Generating and Contextualizing Skills Progressions

Problem generation assesses the question of what problem the learner should work on next. Serious games can take a lesson from entertainment-based games by using an unfolding plotline to motivate problems and to create affective engagement with content. In computer games the skills progression is an important part of creating a sense of mastery and fun. Game Tailor creates a skills progression as a sequence of skill-based events (sandboxes) that is tailored to an individual player and provides a storyline that sets up and explains the skill-based events.

Challenge tailoring (CT) is the problem of matching the difficulty of skill-based events over the course of a game to a specific player's abilities. While not strictly narrative generation, we first consider the problem of generating a skills progression tailored to individual player abilities. This is analogous to the creation of a string of skill-based pearls, but without the narrative "string" that ties the skill-based events together. Once we know the sequence of skill-based events that a player will encounter, the next step it to generate the narrative string that contextualizes each skill-based event. We emphasize the selection of the right sequence of skill-based events for the right player at the right time. Although our approach to challenge tailoring is applicable to a number of serious games, we will illustrate our approach through a simple combat game inspired by The Legend of Zelda. In The Legend of Zelda, the player must lead a team of avatars into periodic combat with teams of opponent monsters. In such a game challenge tailoring may manifest as configuring the number, health, or damage dealt by various enemies at various times throughout the game. CT is similar to Dynamic Difficulty Adjustment (DDA), which only applies to online, real-time changes to game mechanics to balance difficulty. In contrast, CT generalizes DDA to both online and offline optimization of game content and is not limited to adapting game difficulty. Challenge contextualization (CC) is the problem of constructing a chain of non-skill-based events and/or non-interactive sequences that set up the conditions for skill-based events and motivate their occurrence to the player. For example, the challenge of slaying a dragon may be contextualized by the dragon kidnapping a princess.

Challenge Tailoring

Realizing challenge tailoring requires both a player model and an algorithm to adapt content based on that model. Effective player modeling for the purposes of challenge tailoring requires a data-driven approach that is able to predict player behavior in situations that may have never been observed. Because players are expected to master skills over time when playing a game, the player model must also account for temporal changes in player behavior, rather than assume the player remains fixed. Modeling the temporal dynamics of a player enables an adaptive game to more effectively forecast future player behavior, accommodate those changes, and better direct players toward content they are expected to enjoy. Further, forecasting enables player models to account for interrelations among sequences of experiences—accounting for how foreshadowing may set up a better future revelation or how encountering one set of challenges builds player abilities to overcome related challenges that build off of those. We employ tensor factorization techniques to create temporal models of objective player game performance over time. We demonstrate the efficacy of the approach below in a turn-based role-playing game. Further details and evaluation can be found in Zook and Riedl (2012).

Tensor factorization techniques decompose multidimensional measurements into latent components that capture underlying features of the high-dimensional data. Tensors generalize matrices, moving from the two-dimensional structure of a matrix to a three or more dimensional structure. For our player modeling approach we extend two-dimensional matrices representing player performance against particular enemy types to add a third dimension representing the time of that performance measure. Tensor factorization is an extension of matrix factorization, which offers the key advantage of leveraging information from a group of users that has experienced a set of content to make predictions for what a new group of individuals that has only been partially exposed to that content will do. Specifically, if matrix factorization represents user data as a matrix $M = U \times I$ indicating user preference ratings on items, then tensor factorization represents user data as a matrix $Z = U \times I \times T$. Both approaches extract latent factors relating to users and items (and time). The latent factors extracted from the matrix are used to predict missing user ratings of items. The technique for extracting latent factors from the matrix is beyond the scope of this chapter (*c.f.*, Zook and Riedl, 2012). In our usage of tensor factorization, items are challenges—combat, puzzles, or problems to be solved—and ratings are measures of player performance. While we believe our work is the first application of tensor factorization to challenge tailoring problems, we note that similar techniques have been used to model student performance over time on standardized tests (Thai-Nghe, Horvath, and Schmidt-Thieme, 2011).

In our turn-based combat domain, the player leads a team of hero characters against a team of opposing monsters. Each combat is a single skill-based event in a skills progression involving a number of combats. The player can cast a number of spells and different spell types work against different types of monsters. While the role-playing game is a good demonstration of challenge tailoring, it is also a skill learning task. We intentionally created a spell system that was difficult to completely memorize, but contained intuitive combinations—water spells are super-effective against fire enemies—and unintuitive combinations—undeath spells are super-effective against force enemies—ensuring that skill mastery could only be achieved by playing the game. Players do not do well if they do not learn from experience the effectiveness of spells against different opponents. More complicated domains in which the learner must correctly perform complex procedures—such as those used by Annie—are also possible.

We model performance instead of difficulty because performance is objectively measurable while difficulty is subjective. Difficulty and performance have been shown to be significantly (inversely) correlated in the domain of turn-based combat (Zook and Riedl, 2012). Tensor factorization tends to outperform matrix factorization by taking into account the rate at which the player learns the skill of effectively casting spells against opponents of different types. That is, it can predict the actual effectiveness of a player many combats into the future after training. Accuracy of the model is dependent on (a) the number of combats observed of a given individual and (b) the number of overall users represented in the tensor. We find that for our simple combat game, we can achieve high accuracy with as few as 6 training examples per individual and as few as 30 different players. However, more complicated games will a require larger database of player traces. Fortunately, matrix and tensor factorization spreads the model training over a large number of users such that the system need only observe a small number of ratings per user.

To generate particular skill progression of combat episodes, the system utilizes an author-defined performance curve. Typically a performance curve presents the player with a smooth increase in difficulty, *i.e.*, a decrease in player performance over time. Other curves are possible. For example, a curve expressed by $p = c$ (a horizontal line at a fixed constant, c) indicates a game in which the difficulty appears to remain the same, even as the player's skills improve. A dramatic arc, in which the player progressively faces more and more dire challenges until the toughest challenge is overcome and difficulty eases off—can be created with a U-shaped curve. More complicated patterns, such as a series of rises and falls, can express complex designer intentions.

Skills progression generation is an optimization process in which skill-based events are selected such that distance between the predicted performance of the individual on the skills of each event and the performance curve is minimized. A variety of techniques may be applied to solve this dynamic optimization problem including constraint satisfaction, dynamic programming, and heuristic search techniques such as genetic algorithms (Smith and Mateas, 2011; Togelius *et al.*, 2011; Sorenson, Pasquier, and DiPaola, 2011). In contrast to the reactive, near-term changes typically employed in dynamic difficulty adjustment (Magerko, Stensrud, and Holt, 2006; Hunicke and Chapman, 2004), temporal player models are able to also proactively restructure long-term content to optimize a global player experience. Our technique selects sets of enemies for each skill-based event automatically through combinatorial optimization using Answer Set Programming (Baral, 2003). Answer Set Programming is a declarative programming language used for finite domain constraint solving using logic programming semantics.

Challenge Contextualization

But why is the player engaging in the activities that require skills to be practiced? While the sequence of skill-based events can be considered a narrative, the transition from skill-based event to skill-based event creates the context necessary for the player to understand how the skill-based events fit together. Challenge contextualization addresses the issue of player motivation by embedding the skills progression into a larger narrative that does not directly challenge the learner, but engages the learner via fictional means. Challenge contextualization is a form of narrative generation. While challenge tailoring and challenge contextualization can be performed in parallel, we assume a tailored sequence of skill-based events already exists; the selection and parameterization of skill-based events takes precedence in serious games. Thus, the narrative generation problem becomes one of selecting and spacing all skill-based events before "filling the gaps" with non-skill-based, contextualizing events.

Planning is one of the most common approaches to story generation. Planning is the search for a sequence of operations—in this case, events—that transform the world from an initial state into one in which a goal situation holds. To apply story planning to challenge contextualization, the goal situation must be such that it is achieved only if the conditions necessary to establish each skill-based event in turn are achieved at some point in the plan and in order. Skill-based events are sandboxes, and while the actions that occur within a sandbox simulation is dependent on the player and therefore uncertain, all sandboxes have an initial condition (*e.g.*, player and enemy are co-located in the virtual world; a computer has become infected with malware) and a terminal condition (*e.g.*, the opponent is dead; the computer is free of malware).

There are two ways of utilizing story planning in challenge contextualization. The first is to produce distinct planning problems for each pair of skill-based events. In the first iteration of this technique, the initial state is the initial state of the world as specified by a game designer, and the first goal situation is the initial conditions of the first skill-based event. In subsequent iterations, the initial state is the world state that results from executing the plan from the prior iteration updated with the terminal conditions of the last skill-based event, and the goal situation will be the initial conditions of the next skill-based event. The advantage of this approach is that the planning problems are smaller and therefore more tractable. The disadvantage is that narrative decisions made in prior iterations become locked-in and cannot be changed if it is later discovered that it is impossible or awkward to fill a later gap between two skill-based events.

The second story planning approach to challenge contextualization is to consider the entire sequence of skill-based events as part of a single, larger planning problem. We cannot hope that a planner will serendipitously establish the conditions necessary for each skill-based event. To generate a single narrative plan, we must determine how to incorporate all skill-based events simultaneously. In planning, an *island* (Hayes-Roth and Hayes-Roth 1979) is a set of states through which the solution plan must traverse. Any sequence of operators that does not traverse through at least one state in an island at any point is pruned. The initial conditions of each skill-based event is an island and each island must be traversed in the order determined by challenge tailoring. Riedl (2009) describes a technique for incorporating islands into partial order planning. Islands are represented as events with preconditions and effects. The initial plan is seeded with the islands, which are temporally ordered according to the skills progression generated during challenge tailoring. Thus, the preconditions of each island become sub-goals that must be achieved by the planner by inserting non-skill-based events. The solution to the challenge contextualization planning problem is a sequence of events that interleave skill-based and non-skill-based events.

Together challenge tailoring and challenge contextualization provides a solution to the "problem generation" portion of intelligent tutoring systems that focuses on motivating the learner and creating affective and engagement through narrative. The narrative—in particular the non-skill-based events—is not strictly necessary, but breaks up the skill-based events and provides a *reason* for why skills and knowledge must be brought to bear on a sequence of increasingly difficult problems.

RECOMMENDATIONS AND FUTURE RESEARCH

Narrative is one of the fundamental modes for understanding the worlds around us, whether those worlds are real or virtual. Psychological studies show that narrative is read approximately twice as fast as informational text but remembered twice as well (Graesser, Olde, & Klettke, 2002) so clearly it holds a distinguished status in the cognitive system. Virtual environments like computer games have come to blur the distinction between fictional worlds and everyday life as millions of people extend their daily social, leisure and professional identities into these contexts. To a great extent, these interactive systems rely on the explicit role that narrative plays in the design of their users' interactions for their effectiveness.

As intelligent systems develop the capability to model narrative and players' interactions within a narrative space, we argue that the capability to reason about and manipulate story structure in response to learner needs is critical. One key element to this capability is centered on a shift from current games' design focus of linear storylines to more open-ended exploratory environments. Annie's modular model of a game's task environment allows the system to track players as they explore the narrative space of a game and dynamically adjust the story content to address misconceptions as they are identified. The creation of tailored narrative experiences that provide individual learners with the right learning experience at the right time is generally intractable within the context of modern game design practices. For serious games to have the optimal impact on learning and mastery, the narrative experience must address both the pedagogical needs of the learner and encourages affective engagement with content, context to understand why problems are being solved, and motivation to work on progressively harder problems over a long duration of time. Game Tailor seeks to mask a progression of open-ended problem spaces as an unfolding plotline similar to those found in modern computer games while directly addressing the need for tailored pedagogical and narrative content.

Despite recent progress in remediation in open-ended exploratory environments, skills progression generation, and story generation, there are a number of future steps that will make for more robust, scalable, and affectively engaging experiences. First and foremost, automated story generation is a hard problem. While we have shown a considerable gain in computational story generation capabilities, story generation systems such as that used by Game Tailor still do not reliably create narrative structures that fully engage players and learners affectively. That is, automated story generation systems do not understand how the structures they generate produce affective responses in human readers, players, and learners. Recent work suggests that it may be possible for automated story generation systems to computationally model human affective responses to suspense (Cheong, 2007; O'Neill, 2013) and intentionally produce dramatic conflict between virtual characters (Ware *et al*., forthcoming). Second, the linkages between tasks and story are not always clear, nor easy to computationally model. More sophisticated generative models of task progressions are necessary that incorporate procedure and skill level (*c.f.*, Andersen, Gulwani, and Popović, 2013). But even this is not enough, sandboxes are simulations that support open-ended exploration and being able to embed a procedure, task, or skill into a virtual exploratory environment is still not well understood. To the extent that procedures can be represented as narratives—albeit at the level of action instead of event—Hartsook *et al.* (2011) present initial steps toward dynamically creating open-ended virtual worlds that simultaneously support specific narrative elements and open-ended exploration. As these technologies progress, intelligent tutoring systems that exist within the context of serious games and interactive narratives will present learners with more immersive, more engaging learning experiences. By offloading many of the creative and pedagogical decisions onto intelligent systems embedded within these games, we may be able to reach larger populations of learners in informal and non-traditional learning environments.

REFERENCES

Andersen, E., Gulwani, S, & Popović, Z. (2013). A trace-based framework for analyzing and synthesizing educational progressions. *Proceedings of the 2013 International Conference on Human Factors in Computing Systems.*

Aylett, R., Louchart, S., Dias, J., & Paiva, A. (2005). FearNot! an experiment in emergent narrative. *Proceedings of the 5th International Conference on Intelligent Virtual Agents*.

Baral, C. (2003). *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press.

Bateman, C. (2007). *Game writing: Narrative skills for videogames*. Charles River Media.

Cavazza, M., Charles, F. & Mead, S. (2002). Planning characters' behaviour in interactive storytelling. *Journal of Visualization and Computer Animation*, 13, 121-131.

Cheong, Y. (2007). *A Computational Model of Narrative Generation for Suspense*. PhD dissertation, North Carolina State University.

Fikes, R. & Nilsson, N. (1971). STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2, 189-208.

Gee, J.P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment*, 1(1), 20–20.

Gervás, P., Díaz-Agudo, B., Peinado, F., & Hervás, R. Story plot generation based on CBR. *Journal of Knowledge-Based Systems*, 18(4-5), 235-242.

Graesser, A., Lang, K.L., & Roberts, R.M. (1991). Question answering in the context of stories. *Journal of Experimental Psychology: General*, 120(3), 254-277.

Graesser, A.C., Olde, B., and Klettke, B. (2002). How does the mind construct and represent stories? In M.C. Green, J.J. Strange, & T.C. Brock (Eds.), *Narrative Impact: Social and Cognitive Foundations* (231-263). Mahwah NJ: Lawrence Erlbaum Associates.

Hartsook, K., Zook, A., Das, S., & Riedl, M.O. (2011). Toward supporting stories with procedurally generated game worlds. *Proceedings of the 2011 IEEE Conference on Computational Intelligence in Games*.

Hayes-Roth, B. & Hayes-Roth, F. (1979). A cognitive model of planning. *Cognitive Science*, 3.

Hunicke, R., & Chapman, V. (2004). AI for dynamic difficulty adjustment in games. *Proceedings of the AAAI Workshop on Challenges in Game Artificial Intelligence*.

Johnson, W.L. & Valente, A. (2009). Tactical language and culture training systems: Using AI to teach foreign languages and cultures. *AI Magazine*, 30(2), 72-83.

Kambhampati, S., Knoblock, C.A., & Yang, Q. (1995). Planning as refinement search: a unified framework for evaluating design tradeoffs in partial-order planning. *Artificial Intelligence*, 76(1-2), 167-238.

Koster, R. (2005). *A Theory of Fun for Game Design*. Paraglyph Press.

Lebowitz, M. (1987). Planning stories. *Proceedings of the 9th Annual Conference of the Cognitive Science Society*.

Magerko, B., Stensrud, B., & Holt, L. (2006). Bringing the schoolhouse inside the box—a tool for engaging, individualized training. *Proceedings of the 25th Army Science Conference*.

Marsella, S., Johnson, W.L., & LaBore, K. (2000). Interactive pedagogical drama. *Proceedings of the 4th International Conference on Autonomous Agents*.

Mateas, M. & Stern, A. (2005). Structuring Content in the Façade Interactive Drama Architecture. *Proceedings of the 1st AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

McIntyre, N. & Lapata, M. (2009). Learning to tell tales: A data-driven approach to story generation. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing*.

Meehan, J. (1976). *The Metanovel: Writing Stories by Computers*. PhD dissertation, Yale University.

Mott, B.W., Callaway, C., Zettlemoyer, L., Lee, S.Y., & Lester, J. (1999). Towards Narrative-Centered Learning Environments. In M. Mateas and P. Sengers (eds.) *Narrative Intelligence: Papers from the AAAI Fall Symposium* (Technical Report FS-99-01). AAAI Press.

Mott, B.W. & Lester, J. (2006). U-director: a decision-theoretic narrative planning architecture for storytelling environments. Proceedings of the 2006 International Conference on Autonomous Agents and Multi Agent Systems.

O'Neill, B.C. (2013). *A Computational Model of Suspense for the Augmentation of Intelligent Story Generation*. Ph.D. Dissertation, Georgia Institute of Technology.

Pérez y Pérez, R. & Sharples, S. (2001). MEXICA: A computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence*, 13, 119-139.

Porteous, J. & Cavazza, M. (2009). Controlling narrative generation with planning trajectories: the role of constraints. *Proceedings of the 2nd International Conference on Interactive Digital Storytelling*.

Quintana, C., Reiser, B.J., Davis, E.A., Krajcik, J., Fretz, E., Duncan, R.G., Kyza, E., Edelson, D., & Soloway, E. (2004). A scaffolding design framework for software to support science inquiry. *The Journal of the Learning Sciences*, 13(3), 337–386.

Riedl, M.O. (2009). Incorporating authorial intent into generative narrative systems. In S. Louchart, M. Mehta, and D. Roberts (eds.) Intelligent Narrative Technologies II: Papers from the AAAI Spring Symposium (Technical Report SS-09-06). AAAI Press.

Riedl, M.O. (2010). Case-based story planning: Creativity through exploration, retrieval, and analogical transformation. *Minds and Machines*, 20(4).

Riedl, M.O., Saretto, C.J. & Young, R.M. (2003). Managing interaction between users and agents in a multi-agent storytelling environment. *Proceedings of the 2003 International Conference on Autonomous Agents and Multi Agent Systems*.

Riedl, M.O. & Bulitko, V. (2013). Interactive narrative: An intelligent systems approach. *AI Magazine*, 34(1), 67-77.

Riedl, M.O., Stern, A., Dini, D., & Alderman, J. (2008). Dynamic experience management in virtual worlds for entertainment, education, and training. *International Transactions on System Science and Applications*, 3(1), 23-42.

Riedl, M.O. and Young, R.M. (2010). Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39, 217-268.

Rowe, J., Shores, L., Mott, B., & Lester, J. (2011). Integrating learning, problem solving, and engagement in narrative-centered learning environments. *International Journal of Artificial Intelligence in Education*, 21, 115-133.

Smith, A. & Mateas, M. (2011). Answer set programming for procedural content generation: A design space approach. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3), 187–200.

Sorenson, N., Pasquier, P., & DiPaola, S. (2011). A generic approach to challenge modeling for the procedural creation of video game levels. *IEEE Transactions on Computational Intelligence and AI in Games,* 3(3), 229–244.

Swanson, R. & Gordon, A. (2008). Say anything: A massively collaborative open domain story writing companion. *Proceedings of the First International Conference on Interactive Digital Storytelling*.

Thai-Nghe, N., Horvath, T., & Schmidt-Thieme, L. (2011). Factorization models for forecasting student performance. *Proceedings of the 4th International Conference on Educational Data Mining*.

Thomas, J. & Young, R.M. (2010). Annie: Automated generation of adaptive learner guidance for fun serious games. *IEEE Transactions on Learning Technologies*, 3(4), 329 – 343.

Thomas, J. & Young, R.M. (2011). Dynamic guidance for task-based exploratory learning. *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Education*.

Togelius, J., Yannakakis, G., Stanley, K., & Browne, C. (2011). Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3), 172–186.

Trabasso, T., Secco, T., & van den Broek, P. (1984). Causal cohesion and story coherence. In H. Mandl, N.L. Stein, and T. Trabasso (eds.), *Learning and Comprehension in Text*. Lawrence Erlbaum Associates.

Turner. S.R. (1994). *The Creative Process: A Computer Model of Storytelling*. Lawrence Erlbaum Associates.

Van Joolingen, W.R., De Jong, T., & Dimitrakopoulou, A. (2007). Issues in computer supported inquiry learning in science. Journal of Computer Assisted Learning, 23(2), 111–119.

VanLehn, K. The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education*, 16.

Vernieri, T.M. (2006). A Web Services Approach to Generating and Using Plans in Configurable Execution Environments. Master's thesis, North Carolina State University.

Vygotsky, L. (1978). *Mind and Society: The Development of Higher Psychological Processes*. Harvard University Press.

Ware, S.G., R.M. Young, Harrison, B. & Roberts, D.L. (forthcoming) A computational model of narrative conflict at the fabula Level. *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*.

Young, R.M. (1999). Notes on the use of plan structures in the creation of interactive plot. In M. Mateas and P. Sengers (eds.), *Narrative Intelligence: Papers from the AAAI Fall Symposium* (Technical Report FS-99-01), 164-167. AAAI Press.

Zook, A. & Riedl, M.O. (2012). A temporal data-driven player model for dynamic difficulty adjustment. *Proceedings of the 8th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Zook, A., Riedl, M.O., Holden, H.K., Sottilare, R.A., & Brawner, K.W. (2012). Automated scenario generation: toward tailored and optimized military training in virtual environments. *Proceedings of the 7th International Conference on the Foundations of Digital Games*.