A MMORPG Prototype for Investigating Adaptive Quest Narratives and Player Behavior

Emmett Tomai
University of Texas – Pan American
1201 W. University Dr.
Edinburg, TX 78539, USA
tomaie@utpa.edu

Rosendo Salazar
University of Texas – Pan American
1201 W. University Dr.
Edinburg, TX 78539, USA
rsalazar6@broncs.utpa.edu

Dave Salinas
University of Texas – Pan American
1201 W. University Dr.
Edinburg, TX 78539, USA
dsalinasez@broncs.utpa.edu

ABSTRACT
In this paper, we present a research agenda towards adaptive quest narratives in MMORPG shared worlds. There is a significant incompatibility in these games between the dominant *quest* narrative model that guides and motivates gameplay, and player driven change in the world. We discuss incremental extensions to improve the trade-offs in that conflict, and present a simulation system for principled exploration of aggregate player dynamics. This situates and motivates the need for player behavior models and collection of player decision data. We discuss the development of an interactive prototype for gathering this data and evaluating novel extensions to MMORPG game mechanics.

Categories and Subject Descriptors
J [**Computer Applications**]

General Terms
Algorithms, Measurement, Design, Human Factors.

Keywords
Games, interactive narrative, simulation.

1. INTRODUCTION
The MMORPG genre has seen dramatic growth in the past decade. But while many diverse designs exist, the explosive popularity of first Sony's *EverQuest* and then Blizzard's *World of Warcraft* has entrenched a dominant model. One of the core elements of this model is *quest*-driven gameplay. In brief, players control avatar characters in a shared, persistent world. They are free to roam in the world, interacting with the environment and other players as they see fit. Throughout the world there are system-controlled entities (*Non-Player Characters or NPCs*) that the player can interact (speak) with to receive quests. These quests specify task requirements (e.g. kill 10 rats) and rewards (e.g. progress points and a shiny hat), and provide narrative text intended to situate and motivate the task (e.g. help us, the rats are eating all our food!). Multiple quests can be accepted, worked on, put aside, abandoned or completed at the players' discretion. This non-linear, on-demand system of guidance and motivation has proven remarkably effective, in spite of text that is often trite and formulaic and tasks that are highly repetitive.

FDG'12, May, 2012, Raleigh, NC, USA.

This does not mean that game designers or audiences are content with the quest model. It has significant limitations, one of the most striking of which is that it forces the shared world to be mostly unchanging, in order to support multiple players working on quests at different times. Interestingly, the most successful games have lived with this limitation, making static worlds. But developers continue to look for creative ways to add dynamic change in response to player actions. Two of the biggest MMORPGs, BioWare's *Star Wars: The Old Republic* and the later updates to *World of Warcraft*, have heavily used techniques to separate players into copies of the world at key times, enabling localized, non-shared change. In contrast, Trion Worlds' *Rift*, one of 2011's significant successes, attempted to make a more dynamic world, where player actions freely change the world. They omitted quest text to make this work, but found in testing that players felt disconnected from the world. They re-added quests, creating two layers of gameplay: one static and one dynamic, with limitations on the latter [14].

For MMORPGs, it appears that the ability to tell motivating stories is highly valued and necessary, yet there is clear initiative to provide more dynamic, responsive worlds for an increasingly experienced audience. The established quest model provides a simplistic but proven way to adapt static narratives to players' schedules and interactions in a mostly static shared world. We propose an agenda of incrementally increasing both the dynamism of the world (in response to player actions) and the ability of quest narratives to adapt accordingly. In this paper, we present supporting work for this agenda. Extending the quest model is not in itself difficult, but those changes must ultimately be evaluated according to their impact on player experience. And every player's experience is impacted by aggregate player dynamics that can be very complex and non-obvious. Evaluating these extensions in a live game would be ideal, but is highly impractical. Even focused, small-scale controlled testing would require a significant number of players and amount of time. To address these challenges, we have created a lightweight MMORPG simulation for predicting aggregate player dynamics. This simulation is being integrated into a prototype system for small-scale play testing, which will be used to collect player behavior data. This data is necessary to generate player models and investigate whether we can accurately simulate aggregate player decisions. We describe these combined systems in the context of a simple extension to the standard quest model.

2. RELATED WORK
There has been a rise in research on MMORPGs as the genre has grown, primarily focused on broad player motivations and behaviors (cf. [13]), and correlating them with practical, external concerns like network load [2], addictive behavior [4] and real-

money trading [3]. Very few projects, to our knowledge, have explored modeling and altering game play systems like quests to assess their impact on in-game player experience. Even work directly investigating what players do and why [11] is concerned with what types of activities players pursue; a much higher level of abstraction than immediate decisions and outcomes in response to game play systems.

The goal of computer-mediated narratives that adapt to dynamic situations is at the heart of interactive narrative research. By far the most mature pieces of work in the field, resulting in deployed playable game experiences, have worked with relatively small sets of deeper character interactions in a single-player setting (cf. [5] [6]). But the idea of extending the quest system is not novel. The GrailGM system [10] aims to extend RPG quests to provide more diverse player choices than combat actions. However, although it draws examples from both single-player and MMO RPGs, there is no discussion of the constraints on world change in a shared space. Similarly, [1] and [8] both use the multiplayer MMORPG quest model, but focus on the constraints of automatic plot generation between a few directly interacting players. Neither project appears to be targeting the issues of large-scale interaction. The True Story system [7] proposes and implements a quest generation system intended for MMORPGs, but it is not evaluated, and it is not stated how it would deal with the problem of change in the world.

3. ADAPTIVE QUEST NARRATIVE

In this section, we provide more details about the standard quest model and present a preliminary extension for illustrative purposes. One of the significant strengths of the quest system is its modularity. Sets of quests are decoupled from other sets (typically in linear *chains*), allowing players to experience multiple overlapping storylines on-demand. It has the additional benefits of allowing efficient, parallel authoring, and enabling parts to be updated over time without recreating the whole. As we approach adaptive quest narrative, we are not seeking to replace authors with a system that will automatically generate quests. Rather, the goal is to create a system that can take these small, static, modular stories and assemble them at runtime for multiple players in a way that is coherent with a changing world.

Consider the quest text in Figure 1, taken from an early quest in *World of Warcraft*.

> Hey, you there! Ye're a stout-looking <class>. Lend us a hand?
>
> It looks like today's big earthquake shook a bunch o' those barbarous troggs out of the ground, and they're sure steamed about something. My men are doing what they can to hold them off, but we could use a hand.
>
> Do yer part - head just south of here and help dig me mountaineers outta trouble. We've got to hold the line!

Figure 1. Quest text (copyright Blizzard Entertainment).

This quest text reflects a certain state of the world: an ongoing conflict, a specific event leading to a specific battle, and an immediate need. It comes from a *kill* quest, where the player is tasked to kill 6 “Rockjaw Invaders”. These enemies are in-game entities known as *mobs*. Mobs are inserted into the world and typically wander within an authored region or path. Each region has one or more specific types of mobs that *spawn* there, controlled by *spawn points* that ensure the population is always available for the next player who comes along looking to kill them. The sample quest text works with a spawn point that is located where the battle is taking place, which maintains a population of Rockjaw Invader mobs. As discussed in section 1, this system presents a very static world where the Rockjaw Invaders are always there, always fighting, and the player actions have no real effect.

A simple incremental extension to create a more dynamic world and increase player impact would be allowing player actions to reduce that mob population and eliminate the spawn point, actually defeating the enemy. But that would break the quest for players currently working on it, and make it unavailable for others. However, a matching extension to the standard quest model could alter quest locations and mob details to distribute players among appropriate, active spawn points, creating new ones as necessary. This incremental extension is feasible for implementation, and attempts to increase the responsiveness of the world while maintaining, and even improving, narrative coherence.

But the implications of this change are neither simple nor obvious. The aggregate dynamics of a large population of players attempting to complete numerous, overlapping quests in the same area are complex to begin with. This change greatly increases that complexity, making it very challenging to predict the impact on player experiences.

4. SIMULATION OF AGGREGATE PLAYER DYNAMICS

To address this problem for extensions of the quest model, we have created a lightweight MMORPG simulator. The simulator implements the standard quest model, with quest-giving NPCs, mobs and spawn points. These are organized within a single *zone*, a geographical area with author specified regions that serve as *quest hubs* and *questing areas*. Quest-giving NPCs are placed in the former, while mob and node spawn points are placed in the latter. The simulator runs any number of concurrent simulated players through a zone where they seek out, accept and complete quests. It is intended to show potential aggregate player dynamics by using a reasonable player model to make individual decisions. Player modeling is a very broad concept. [9] presents a taxonomy of player models, under which this is a straightforward Individual Generative Action model. It stands in the place of an actual player’s in-game actions. However, the goal is not to accurately predict what an individual will do. Rather, we want the simulation to sample the space of likely interaction possibilities across a population of players, looking for potential problems.

For example, in preliminary experiments, we have explored how the simple extension described in section 3 impacts the competition that players experience amongst themselves for mobs to kill (widely considered a negative). Our initial simulated results showed that allowing players to defeat spawn points greatly increased competition, as expected, but that combining that with altering locations in the quest text could actually lower competition [12]. While there were many simplifying assumptions in this experiment, the results are promising both for adaptive quest narratives, and using the simulation to predict the impact of quest extensions. We are currently investigating the correlations between the design decisions in the zone (e.g. topological layout, densities, quest constraints) and the effectiveness of the extension.

4.1 Player Modeling

The player models in that preliminary experiment were quite naïve. They were constructed based on the game play experiences of the development team. At certain junctures in play, the player model has to choose either an incomplete quest to pursue or a complete quest to turn-in. These decisions were made using a linear weighted sum of preference features: distance to travel, time commitment, spatial clustering, reward value, difficulty and completion percentage. Random weights were assigned for each simulated player to create diversity in the player population. Returning to the taxonomy of [9], the model source was Synthetic, based on subjective intuitions. A clear goal is to move towards a more sophisticated probabilistic model that is Induced (learned) from actual player data.

We are planning experimental play tests to collect player action traces in order to construct more accurate player models. Our first hypotheses are that, given a set of individual player action traces, with a known zone configuration, we can train reinforcement learning models for individual decisions that can predict aggregate behavior for the same and a different zone configuration. The set of features from the preliminary experiment will be used, which necessitates that the players in the test are made aware of the feature values for each quest. The learned player models will be Markov decision policies over those values. We will further use this data to investigate and develop a model of transfer of this predictive knowledge to incremental extensions.

5. MMORPG PROTOTYPE

To gather player data, we are building a small-scale prototype interactive game based on the standard MMORPG quest model. True MMORPG development is an enormous undertaking, so there are three key ways that we are limiting the scope. First, our experimental play tests will take place in a controlled lab. This is important because it mitigates a huge amount of work in network protocol optimization and cheating prevention. The clients are not, as they say, "in the hands of the enemy", and so can be trusted with calculations that would otherwise have to be verified on the server. For example, the clients in our prototype are authoritative on their own position in the world.

Second, our experimental play tests will be limited to a single session in a single zone. MMORPGs are expected to provide orders of magnitude more hours of game play than standard, single player games. This creates a need to generate massive amounts of content in a huge world. For technical and game play reasons, these worlds are divided into multiple zones. Each zone typically requires a distinct theme, unique enemies, music and other art assets. Further, players expect a variety of avatars to choose from, each with diverse and compelling combat abilities they learn as the player progresses through the game. Players also expect to obtain increasingly better (and visually diverse) equipment for their character as they play. For our purposes, we need only a single player character type with two abilities (melee and ranged combat), and a single zone with a few structures and 2-3 types of enemies. We need only a handful of equipment upgrades, to motivate quest completion. Visual quality and consistency are not high priorities and we are relying on modifications of freely available art assets.

Third, we are interested in a limited number of gameplay mechanics. To keep players engaged, MMORPGs layer numerous mechanics and systems on top of the quest system, such as in-game crafting of useful items, social networking and organizational tools, achievements and player-vs-player combat. There is also an increasing need for designers to come up with diverse quest types and narratives. For our experiments, we are only concerned with supporting questing. This is similar to the early game in many MMORPGs, before other mechanics are introduced and encouraged.

5.1 Prototype Design

We are using the same codebase for both the prototype and the simulator, so that simulated players and interactive players are presented with the same environment. In fact, simulated and interactive players can coexist in a single game, which we believe will be useful for approximating larger populations down the road.

The prototype will support the same physical entities in the world as the pre-existing simulator – player characters, NPCs and mobs – plus *nodes* (interactive objects in the world that spawn in the same way as mobs, but do not move or fight) and *named mobs* (mobs that have a single instance in the world at a time). Players can interact with entities in the following ways.

- **Attack**: use an ability to cause damage to a mob
- **Speak**: talk to an NPC via dialogue boxes, often in order to obtain or turn-in quests
- **Gather**: interact with a node to obtain an item
- **Trigger**: interact with a node to satisfy a quest condition
- **Loot**: interact with a dead mob to obtain items (ostensibly items the mob was "carrying")
- **Discover/explore**: arrive at a region, or a certain point within a region

The five most prevalent types of quests are implemented.

- **Kill**: kill a certain number of a type of mob
- **Collect**: obtain a certain number of a type of item, either by gathering or looting
- **Interact**: trigger a node or speak to an NPC
- **Explore**: explore a specified region
- **Deliver**: turn-in to an NPC who is not the quest giver, possibly while holding a required item

Authoring a zone in this prototype involves defining the topological regions and placing quest-giving NPCs and spawn points for mobs and nodes. So-called *loot tables* must be defined for mobs and nodes to determine what items can be obtained from them, and mobs must be given combat statistics, such as how much damage they can take. Quests must be defined with objectives, a quest giver and completer, narrative text and pre-requisite constraints.

The simple extension described in section 3 is already implemented in the simulator and will be carried over to the prototype. The necessary textual alterations will be simplified to template substitutions for this stage of the project.

The prototype will be instrumented to collect data on player behavior. Of course player intentions cannot be directly captured, only the actions they choose to take. We will capture all the quest related events (i.e. accept, progress made, complete, turn-in), as well as region transitions and combat statistics. Since this is a controlled client, we have the option of capturing more granular data such as what entities are on the screen and what entities are being selected with the mouse. This data is typically client-side only for performance reasons. We plan to dynamically upload captured data to the server through the same communication channels as game play, but if necessary we have the option to fall back on local storage since we control the clients.

The user interface for the prototype will follow the very standard conventions of the genre. Significantly, indicators for the player model features must be present. These include map markers for distance and clustering, number-of-objectives for time commitment, relative mob difficulty, objective tracking for completion percentage and exaggerated distinctions in reward value.

5.2 Prototype Implementation

We are using the *Unity3d* game engine (http://unity3d.com) for the client. This complete rapid development environment allows for a great deal of dropping in features in the form of modular, pre-made assets and scripts. The licensing is free for non-commercial development with no limitations that concern us, and there is an active community developing and sharing features, both free and at cost. Further, it can compile to a web browser-based client, making for easy deployment in our labs. We are using *SmartFox Server 2X* (http://www.smartfoxserver.com/) for the server, which has built-in Unity support and a full featured community edition license for up to 100 concurrent connections. SmartFox is a message passing architecture, providing network communication and server-side management and debugging tools. Unity scripting can be done in JavaScript or C#, and SmartFox server extensions can be written in Java or Python. These are good language choices for us, as we can easily find students comfortable with them. Given the scope limitations we have established, rapid development and maintainable code are much higher priorities than performance. Authorial specifications are stored in a MySQL database, again with simplicity and efficient workflow first in mind. Given the single, controlled session design, we do not have to worry about maintaining player data across sessions or in the event of a server crash. This is a significant simplification, as we can cache database data in memory and avoid difficult asynchronous update issues across a large number of players.

6. CONCLUSION AND FUTURE WORK

We have proposed an agenda for incremental extension of the standard MMORPG quest model towards adaptive narratives in a shared world. The aim is to improve the trade-off between the ability for authors to tell motivating narratives and the ability for players to change the world. We have presented a simulation approach to support this agenda, by enabling the exploration of aggregate player dynamics and the impact of novel mechanics on certain types of player experience. This approach requires research and experimentation on player decision modeling, for which we are building a small-scale MMORPG prototype.

We believe the problem we are tackling has a lot in common with the problem an author faces in an MMORPG. Design decisions, like system extensions, result in complex aggregate dynamics that are difficult to predict. We expect that this combination prototype/simulator will also be useful in creating authoring support tools that allow authors to more easily explore the implications of their designs.

7. ACKNOWLEDGMENTS

This material is based upon work supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under grant number W911NF-11-1-0150.

8. REFERENCES

[1] Fairclough, Chris; Cunningham, Pádraig. A Multiplayer Case Based Story Engine. Dublin, Trinity College Dublin, Department of Computer Science, TCD-CS-2003-43, 2003, pp6.

[2] Feng, Wu-chang, Brandt, David, and Saha, Debanjan. 2007. A long-term study of a popular MMORPG. In *Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games (NetGames '07)*. ACM, New York, NY, USA, 19-24.

[3] Fujita, A., Itsuki, H., and Matsubara, H. Detecting Real Money Traders in MMORPG by Using Trading Network. In *Proceedings of the 7th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, pp. 26-31, Oct., 2011.

[4] Hsu, Shang Hwa, Wen, Ming-Hui and We, Muh-Cherng. Exploring user experiences as predictors of MMORPG addiction. *Computers & Education*. Volume 53, Issue 3, November 2009, Pages 990–999.

[5] Mateas, M., and Stern, A., 2002. A Behavior Language for Storybased Believable Agents. In *IEEE Intelligent Systems, special issue on AI and Interactive Entertainment*.

[6] Mott, Bradford, Lee, Sunyoung, and Lester, James. Probabilistic Goal Recognition in Interactive Narrative Environments. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, Boston, Massachusetts, pp. 187-192, 2006.

[7] Pita, James, Magerko, Brian and Brodie, Scott. True story: dynamically generated, contextually linked quests in persistent systems. In *Proceedings of the 2007 conference on Future Play (Future Play '07)*. 2007. ACM, New York, NY, USA, 145-151.

[8] Riedl, Mark, Li, Boyang, Ai, Hua, and Ram, Ashwin. Robust and Authorable Multiplayer Storytelling Experiences. *Proceedings of the Seventh International Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*. 2011.

[9] Smith, A.M., Lewis, C., Hullett, K., Smith, G., Sullivan, A. An Inclusive Taxonomy of Player Modeling. Technical Report UCSC-SOE-11-13. University of California – Santa Cruz. 2011.

[10] Sullivan, A., Mateas, A., and Wardrip-Fruin, N. Rules of engagement: moving beyond combat-based quests. In *Proceedings of the Intelligent Narrative Technologies III Workshop*. 2010.

[11] Suznjevic, Mirko, Matijasevic, Maja. Why MMORPG players do what they do: relating motivations to action categories. *International Journal of Advanced Media and Communication*. Vol 4, Number 4/2010, pp 405-424.

[12] Tomai, Emmett, Salazar, Rosendo and Salinas, David. Using Adaptive Quests for Dynamic Change in MMORPGs. *(Submitted as a poster to this conference)*.

[13] Yee, Nick. Motivations for Play in Online Games. *CyberPsychology & Behavior*. December 2006, 9(6): 772-775.

[14] Zacny, R. 2012. Rift Executive Producer Scott Hartsman explains how dynamic content drives the world of Telara. PC Gamer website. http://www.pcgamer.com/2012/03/06/rift-executive-producer-scott-hartsman-explains-how-dynamic-content-drives-the-world-of-telara