# A Parametric Analysis and Classification of Quests in MMORPGs

Felipe K. C. Santos Geber L. Ramalho

Federal University of Pernambuco, Dept. of Computer Science, Brazil

## Abstract

Massively multiplayer online role-playing games (MMORPG) are a particular sub-genre of massively multiplayer online games (MMOG) whose commercial importance has experienced a growth spurt in the last five years marked by commercially successful games such as World of Warcraft. A characteristic of the genre is a focus on character progression and open-ended narratives by means of quests, which represent tasks assigned to the players in exchange for in-game rewards.

Progression mechanisms typically involve static requirements defined by the game designers, in such a way that players of different skill levels performing a set of quests always face the same challenge and obtain the same rewards. As a consequence, speed of progression through the game is completely disassociated from player experience and skills.

This paper proposes a comprehensive quest classification model, as well as the set of difficulty parameters associated to each quest class, by the exhaustive analysis of 480 quests. This allows for the creation of quest variants specifically tailored to the needs of each player by the game designers, and constitutes the first step towards a dynamic difficulty adjustment (DDA) implementation.

**Keywords**: MMORPG, difficulty adjustment, quest, classification, DDA.

**Authors' contact**:
{fkcs,glr}@cin.ufpe.br

## 1. Introduction

Massively multiplayer online Game, or MMOG, is a specific genre of games that originated from traditional single-player games with the idea of bridging the gap between individual players and form communities that bring additional value to the experience of playing a game. By introducing other players in a virtual environment, the game becomes more connected to the real world, making individual progress more rewarding. Concepts such as collaboration between characters and direct player versus player confrontations are evidence of this shift in focus from a personal experience to a social one.

A key aspect of MMOGs in general is the persistence of the virtual environment, that is constantly evolving and changing even without direct input from the player. This dynamicity in the environment is only possible due to the large number of players who play MMOGs for long periods of time. The most popular current MMOG, World of Warcraft, has reached over its lifespan over 8.5 million players worldwide, with an average of 200 thousand simultaneous connections [Raptorak 2011].

Of particular interest is a subgenre of MMOGs that associates the characteristics of interaction and community building with elements of character progression typical of role-playing games (RPG). Massively multiplayer online games (MMORPG) are atypical in that they provide a deeply immersive storytelling experience by allowing players to assume the role of a character in a narrative and exact direct control over its actions. As in all RPGs, customization is a central aspect of gameplay, a typical MMORPG allowing for both character progression and character customization. The main objective in almost all MMORPGs can be described, then, as the constant development and evolution of a player's avatars.

Several progression mechanisms must therefore be put in place to ensure that players always have challenges to face and rewards to obtain, the most common of which is the use of experience points [Mulligan and Patrovsky 2003]. Upon completion of goals and as they interact with the environment of the game, players receive points that can be accumulated and used to advance their characters in pre-defined and discrete steps, a process known as level-up. By advancing to the next level, avatars acquire new skills and increases in their basic attributes, effectively becoming more powerful in the context of the game. Experience points are traditionally gained by vanquishing enemies, crafting items, completing quests, or participating in holiday events.

Despite being the central aspect of character progression, the experience points system is often complemented with a similar system to separately manage skills. Skill points can be obtained in the same way as experience points, due to the accumulation of experience points, or any variation thereof as defined by game mechanics [Wu-Chang et al. 2007]. The connection between the two systems is dependent on their implementations, but it is common practice to put in place limitations that ensure a parallel advancement in both at a similar rate.

Other mechanisms can also have indirect impact on avatar progression, such as the use of equipment or other attribute-based items (commonly referred to as "loot" when obtained as reward from a defeated enemy). In most games of the genre, players can boost their character's strength, resistance, and vitality by switching obsolete pieces of equipment with more expensive gear. The choice of equipment is controlled by acquisition costs - the more powerful a piece of equipment is, the harder it must be to acquire. Thus, weapons and armors with better benefits are more easily obtained the farther along characters are on the progression scale. Nevertheless, it is common practice to directly restrict the use of equipment to a specific range of characters levels, to ensure early-game balance.

Finally, actual player experience and familiarity with game controls can also be interpreted as a factor of progression. The game EVE Online is a typical example, being a hybrid space simulator and MMORPG in which the ability to manage resources and control characters determine the success and progress of the player [CCP Games 2011]. It is also common the occasional inclusion of changes in control schemes or elements from other genres to alter the difficulty of quests. A typical example of this is the game World of Warcraft, known for its vehicle quests [WoWWiki 2011] in which players must achieve a specific goal while driving a vehicle with unique controls, riding an animal, controlling an aircraft, or even in control of an unfamiliar character.

The presented mechanisms for avatar progression are therefore linked to the rewards obtained by completing challenges in-game, be them experience points, new skills and abilities, or equipment upgrades. As such, the progression rate of a player is determined by how quickly and efficiently they can obtain said rewards and improve their character, which in turn improves their reward collection efficiency. This cycle of rewards and improvements can quickly escalate out of control with players advancing too rapidly towards the end-game if the requirements for progression are not scaled appropriately. For the players, this means that each game challenge has pre-defined rewards according to an estimated difficulty, and that in order to progress they must face ever more difficult challenges.

The problem with said approach is that not all players feel the same satisfaction and perceived difficulty from a standardized challenge. Players with more experience could want to progress faster towards the end-game at the expense of facing bigger challenges, and thus obtaining larger rewards. Such a need can be addressed both by statically defining quest difficulty profiles and by means of dynamic difficulty adjustment (DDA) techniques.

In this context, quests are a natural choice for difficulty adjustment, as they represent a set of self-contained tasks that the player must perform in exchange for rewards in accordance with the narrative adopted by the game.

Traditionally, an MMORPG contains a series of quests that reflect the many possible evolution stages of player characters, aiming to provide for any given character level at least one quest specifically balanced to characters of that level. Quest difficulty in this approach is static and defined as part of the development cycle based on the statistical analysis of player success indicators. In practice, players ends up facing periods of gameplay in which all quests below their level have been completed but still have not attained the necessary progression stage to face the next implemented quests. This intermediate state in which the players remain aimless and are forced to evolve on their own in order to proceed with the game is referred to as grinding [Raptorak 2011], and is primarily responsible for the loss of enthusiasm and motivation that leads a player to abandon an MMORPG.

Regarding difficulty adjustment, we must therefore balance the elements of the game directly related to progression. In light of the definition given above, quests can be seen as the direct link between the game environment and all virtual elements, allowing for a quest-centric view, by which NPCs, objects, and locations exist only when deliberately instantiated by a quest. The problem of difficulty adjustment in MMORPGs can thus be simplified to the problem of adjusting the difficulty of quests to each individual player.

This study elaborates on the topic of quest difficulty adjustment in two preliminary steps to construct a comprehensive classification structure for quests and identify quest elements that, when adjusted, have a direct impact on perceived difficulty.

We began with a literature review on the fields of procedural quest generation and DDA methodologies to construct a classification structure aimed to our specific needs. This is then followed by the analysis of 480 quests, which serve the double purpose of validating the defined categories and allowing for the identification of the difficulty-related elements of each class.

This paper is structured in six major sections. In Section 2 we consider related work in the field of difficulty adjustment, quest classification, modeling, and DDA. In Section 3 we review the methodology used for the classification and present the finalized structure. Section 4 illustrates our approach to quest parameterization process, includes a description of each class, and the tables with identified difficulty parameters for each one. In Section 5 an example quest analysis is given. Section 6 covers possible parameter conflicts when adjusting the difficulty of quests in a multiplayer environment. Finally, Section 7 presents a few closing comments and notes on future works.

## 2. Related Work

Different approaches can be found in the literature to address the problem of difficulty adjustment in games. In all cases it is necessary to define measures for determining the difficulty a player is facing at any given time, usually by means of a "Challenge Function" that maps a state of the game to a value representing the current difficulty.

Two main approaches were identified by Pfeiffer, B. [2003] to dynamically adapt the elements and parameters of a game to a new level of difficulty: a behavioral approach, by which the behavior of non-player characters is adjusted with machine learning algorithms, such as genetic algorithms [Hunicke et al. 2004], and an environmental approach, in which only the parameters that model game environment are modified.

In the context of MMORPGs, it becomes evident that an environmental approach best conciliates the characteristics of the genre of having complex and parameter-centric environments, while having simple combat mechanics and NPC behaviors.

One approach to environmental balancing proposed by Hunicke et al. [2004] uses a probabilistic analysis of the variables involved in a first-person shooter to predict the behavior of the player and alter the necessary parameters to obtain the desired level of difficulty. If, for instance, the game is too difficult, the player can be given more weapons and hit points, or enemies may become less frequent and have reduced vitality. This approach has good results, but for games in which the actions of non-player characters are complex and directly influence the difficulty of the game, as is the case in shooters, the change in environmental factors may become evident to the player. In this case, a behavioral approach to control the environmental changes might give better results.

In line with a behavioral approach of intelligent agents, Demasi and Cruz [2002] define agents, and employ genetic algorithms to select those that are best suited to each player. The method promotes the development of agents with a co-evolving online system to accelerate the learning process, using pre-defined models of agents with good genetic characteristics as ancestors in the steps of the algorithm. Andrade et al. [2005] proposes a similar solution by using a modified version of Q-learning to change the behavior of the agents in a fighting game. For this, the authors divide the problem in two dimensions: competence (knowing the best approach) and performance (choosing between possible approaches). The proposed mechanism of learning addresses both dimensions through an initial stage of off-line bootstrap learning, followed by a setup stage during play.

The focus of this paper being the difficulty adjustment of quests, two works of note on quest structural analysis and classification should also be mentioned.

The classification proposed by Doran et al. [2010, 2011], for instance, is based on the motivations of NPCs as they make requests to the players, and used to construct a grammar for procedural quest generation. In their study, 750 quests from a number of different games were analyzed and represented as a sentence in the proposed grammar.

Pita et al. [2007], on the other hand, proposed a classification based on the interaction between objects in the virtual world, according to which an object can represent a player character, a non-player character, or an item. In this context, a quest is a process involving N input objects and M output objects. An example of this can be seen in a "Kill" quest, which consists of a function with two input objects - the player character and a non-player character - and one output object (the player character).

## 3. Quest Classification

To construct our classification, we used as scaffold the ones proposed by Doran et al. [2010] and Pita et al. [2007] in conjunction with the unofficial classification available on WowWiki for the game World of Warcraft. The reason we chose to focus on this specific game for both the classification and analysis steps was due to a general lack of details made available for other commercial products. World of Warcraft has a comprehensive online database of quests and additional walkthrough information, as well as detailed statistics for game elements. For the classification proposed, however, such a choice has little impact on the overall model, due to all MMORPGs following similar premises, as pointed out by the referred works [Doran et al. 2011].

The resulting classification, which can be found with a side-by-side comparison on table 15, has a significant bias towards quest difficulty. Such a bias is what results in a class structure very similar to the one of World of Warcraft, segregating quests on game elements and mechanics. As such, the representation of a quest in the presented model can be better understood as a template for the manual adjustment of difficulty by a game designer or an eventual DDA implementation. Along with a unified framework for difficulty adjustment, this allows for easy implementation on existing games.

Aggregating quests with this approach resulted in a classification structure consisting of eleven separate classes. Quests that do not pertain to any specific class can either be decomposed into a sequence of sub-quests or classified as unique. The latter is mostly the case when the difficulty is associated with specific gameplay mechanics. Typical examples are the vehicle quests from World of Warcraft, whose difficulty is a direct result of the vehicle control scheme, and cannot easily be changed.

## 4. Quest Parameterization

To determine which quest elements have an impact on difficulty, we initially analyzed 340 quests from the total of 9850 quests present in World of Warcraft at the time [WowHead 2012], and prepared and additional test set of 140 quests. The resulting set of 480 samples corresponds to 4.01% of the population. Quest descriptions where obtained from WowHead, a community driven database, and complemented with comments from the official World of Warcraft forum and quest walkthroughs. Whenever specific pertinent details could not be found, all possible assumptions where considered and analyzed separately. This approach is valid because the additional parameters resulting from the assumptions do not interfere with those previously identified.

Each quest was selected with uniform probability from the pool of available quests at WowHead using the quest unique id as a selector. In specific instances, when a quest was deemed obsolete and associated data removed from the database, the quest was discarded and a new one selected following the same process.

Having completed the parameterization of the initial set of 340 quests, the test set composed of an additional 140 quests (29.2% of our full set of 480) was divided into groups of 35 samples each, which were analyzed separately and compared to the current parameterization to find the number of non-conformities. Each sub-division of this test set was then successively incorporated into the model before comparison began with the next one.

By using such an approach we can plot the curve of non-conformities as we approach a larger sample size and predict if the number of quests used is sufficient for the construction of a robust model. The resulting graph in figure 1 shows a notable reduction on the number of changes required to our base model as more and more quests are analyzed, and that the overall number of missing parameters, non-conformities, and misclassifications amount to 7.8% of the number of quests considered. Using that estimate we can predict that at most 768 quests in World of Warcraft would not directly conform to the presented parameterization.

While this number may seem quite large when we consider a possible implementation of a DDA mechanism for quest balancing, it is perfectly viable as a tool for the game developer to easily create quest variants of different difficulties.

We observed that, although each class corresponds to a specific set of difficulty elements, one such can belong to a number of different classes. Each class can then be interpreted as a combination of the same basic elements in different roles. By switching components, adjusting component parameters or manipulating their quantities, the difficulty of the quest can be increased or decreased. Table 16 summarizes these basic components. In addition to these components, each class can also include additional parameters or flags to represent specific values unique to each one.

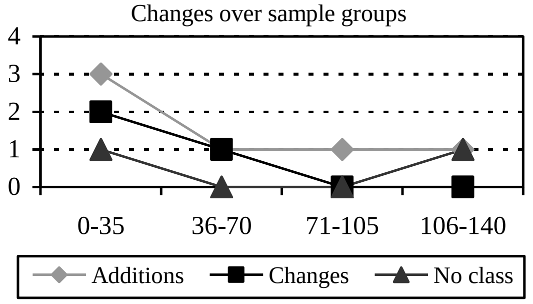

Figure 1: Changes on classification over samples

The following is a brief discussion of each class that composes our classification and their associated parameters. Specific terms and naming conventions used on the descriptions are defined in the table 1 below.

Table 1: Common Terms

| Term | Definition |
|---|---|
| Spawn | A location from which enemies are added to the game world at set intervals, covering a specific area. |
| Defender NPC | NPCs that assist the player. |
| Collateral NPC | NPCs that may or may not attack the player, existing as fauna to a specific location. |
| Ambush NPC | Similar to Collateral NPCs, but cannot be avoided and don’t revive after death. |
| Time to Finish | Time limit to complete a quest. |
| Final NPC | NPC that must be spoken to in order to conclude the quest. |
| Item Drop Rate | Probability that an enemy or object will give a specific item after being interacted with by the player. |
| Scenery Object | Object placed on the virtual world that functions similarly to enemies but cannot attack the players. |
| Check Point | A point on the path an NPC must take when moving. Can also have a wait time before the NPC is allowed to continue. |

### 4.1 Kill Quests

Kill Quests are quests of extermination in which the player is asked to eliminate a specific amount of a certain NPC distributed among various locations around the virtual world. This type of quest can determine a period of time within which the player must reach the required quota and is generally restricted to locations close to where it was obtained.

The quest may involve a specific amount of each NPC or a number of NPCs of a same species or group. Defender NPCs can aid the player during the quest, by providing assistance during combat. Additionally, the player may face Collateral NPCs not directly related to the quest, but that can also attack them on sight and must in turn be defeated or avoided.

Table 2: Kill Quest Parameters

| Range | Component | Type |
|---|---|---|
| [1, ∞[ | Target NPC | NPC |
| | Quantity | Number |
| | Spawn | Spawn |
| [0, ∞[ | Defender NPC | NPC |
| | Spawn | Spawn |
| [0, ∞[ | Collateral NPC | NPC |
| | Spawn | Spawn |
| [0, 1] | Global Quantity | Number |
| [0, 1] | Time to Finish | Number |
| [0, 1] | Final NPC Location | Location |

### 4.2 Loot Quests

Loot Quests are similar to Kill Quests, but instead of a quota of vanquished foes, have as requirement a specific number of items that must be collected from slain NPCs. The term “loot” refers to the rewards obtained in this fashion. In this context, the item drop rates (probability that a defeated foe gives a specific item as reward) for each NPC can significantly affect overall quest difficulty by increasing the average number of foes that must be defeated to obtain a certain quantity of items.

Table 3: Loot Quest Parameters

| Range | Component | Type |
|---|---|---|
| [1, ∞[ | Target NPC | NPC |
| | Item Drop Rates | Number Array |
| | Spawn | Spawn |
| [0, ∞[ | Defender NPC | NPC |
| | Spawn | Spawn |
| [1, ∞[ | Item | Item |
| | Quantity | Number |
| [0, ∞[ | Collateral NPC | NPC |
| | Spawn | Spawn |
| [0, 1] | Time to Finish | Number |
| [0, 1] | Final NPC Location | Location |

In all the quests analyzed there were no instances in which Defender NPCs assisted in the collection of the items but, as that remains a possibility, they were included in the class parameters.

### 4.3 Gathering Quests

Gathering Quests are the equivalent of Loot Quests for containers spread around the game world. In this type of quest the player must gather a number of items from those containers, usually avoiding or fighting aggressive NPCs in the area. As such, there are no specific instances of NPCs to kill, only the eventual Collateral NPCs. The same factors of item availability from Loot Quests can have an impact on the perceived difficulty.

Table 4: Gathering Quest Parameters

| Range | Component | Type |
|---|---|---|
| [1, ∞[ | Object | Object |
| | Item Drop Rates | Number Array |
| | Spawn | Spawn |
| [1, ∞[ | Item | Item |
| | Quantity | Number |
| [0, ∞[ | Collateral NPC | NPC |
| | Spawn | Spawn |
| [0, 1] | Time to Finish | Number |
| [0, 1] | Final NPC Location | Location |

### 4.4 Boss Kill Quests

Boss Kill Quests have similarities with Kill Quests in that both require a number of NPCs be eliminated. The difference lies on the maximum number of instances of each NPC present on the game world at any given time. An observed characteristic of Boss NPCs (also known as Named NPCs in some games) is that they usually appear together with a group of Ambush NPCs that the player must fight simultaneously and assist the boss during combat.

Table 5: Boss Kill Quest Parameters

| Range | Component | Type |
|---|---|---|
| [1, ∞[ | Boss NPC | NPC |
| | Spawn | Spawn |
| [0, ∞[ | Collateral NPC | NPC |
| | Spawn | Spawn |
| [0, ∞[ | Ambush NPC | NPC |
| | Spawn | Spawn |
| [0, 1] | Time to Finish | Number |
| [0, 1] | Final NPC Location | Location |

### 4.5 Escort Quests

Escort Quests are quests in which the player must assist a moving NPC to safely reach its destination. The difficulty in this case can be altered with the inclusion of Ambush NPCs along the projected path of the Escorted NPC, by changing its movement speed, or by requiring that it performs an action at each check point. In some instances, the player may have the assistance of Defender NPCs along the way.

Table 6: Escort Quest Parameters

| Range | Component | Type |
|---|---|---|
| [1, ∞[ | Escorted NPC | NPC |
| | Spawn | Spawn |
| | Movement Speed | Number |
| [0, ∞[ | Defender NPC | NPC |
| | Spawn | Spawn |
| [0, ∞[ | Ambush NPC | NPC |
| | Spawn | Spawn |
| [0, ∞[ | Collateral NPC | NPC |
| | Spawn | Spawn |
| [1, ∞[ | Check Point | Location |
| | Wait Time | Number |
| | Interruptible | Boolean |
| [0, 1] | Time to Finish | Number |
| [0, 1] | Final NPC Location | Location |

## 4.6 Interaction Quests

There are three quest categories to model player interaction with different game elements. Interaction (NPC) Quest aggregates the quests in which the player has to activate a non-aggressive NPC. This interaction may or may not require an item, and usually represents transition steps in larger quest lines.

Interaction (Scenery) Quests are the equivalent for interaction with objects, be it to activate a pedestal with a crystal or light a bonfire with oil. Finally, Interaction (Item) Quest represents the interaction with items on the player's inventory. Quests of this last category are usually called crafting quests, as they transform a group of base items into another. This class, however, also contains the quests in which the player is simply required to use an item at a specific location.

Table 7: Interaction (NPC) Quest Parameters

| Range | Component | Type |
|---|---|---|
| [1, ∞[ | Target NPC | NPC |
| | Spawn | Spawn |
| | Item Needed | Item |
| [0, ∞[ | Collateral NPC | NPC |
| | Spawn | Spawn |
| [0, 1] | Time to Finish | Number |
| [0, 1] | Final NPC Location | Location |

Table 8: Interaction (Scenery) Quest Parameters

| Range | Component | Type |
|---|---|---|
| [1, ∞[ | Scenery Object | Object |
| | Spawn | Spawn |
| | Item Needed | Item |
| [0, ∞[ | Collateral NPC | NPC |
| | Spawn | Spawn |
| [0, 1] | Time to Finish | Number |
| [0, 1] | Final NPC Location | Location |

Table 9: Interaction (Item) Quest Parameters

| Range | Component | Type |
|---|---|---|
| [1, ∞[ | Item Needed | Item |
| | Quantity | Number |
| [0, ∞[ | Item Produced | Item |
| | Quantity | Number |
| [0, ∞[ | Location | Location |
| [0, 1] | Time to Finish | Number |
| [0, 1] | Final NPC Location | Location |

## 4.7 Exploration Quests

Exploration Quests are most common as achievements, or as a way to motivate the player to explore a specific region of the game environment. Quest difficulty is in this case dependent exclusively on Collateral NPCs spread over the explored region that the player must avoid or confront. It is also possible to impact quest difficulty by incurring a time limit for each check point that must be crossed or for the overall duration of the exploration.

Table 10: Exploration Quest Parameters

| Range | Component | Type |
|---|---|---|
| [0, ∞[ | Collateral NPC | NPC |
| | Spawn | Spawn |
| [1, ∞[ | Check Point | Location |
| | Time Limit | Number |
| | Item Needed | Item |
| [0, 1] | Time to Finish | Number |
| [0, 1] | Final NPC Location | Location |

## 4.8 Delivery Quests

In Delivery quests, the player must bring a set of provided items to a specified NPC. Similarly to Interaction (NPC) Quests, these quests are usually the connecting elements in a quest line, with previous quests resulting in the player having possession of the required items. As such, the major difficulty influences reside only with the Collateral NPCs.

Table 11: Delivery Quest Parameters

| Range | Component | Type |
|---|---|---|
| [1, ∞[ | Item Needed | Item |
| | Quantity | Number |
| [0, ∞[ | Collateral NPC | NPC |
| | Spawn | Spawn |
| [0, 1] | Time to Finish | Number |
| 1 | Final NPC Location | Location |

## 4.9 Defense Quests

Defense Quests can be understood as a static variant of Escort Quests taking place in a fixed location in the virtual world. In most cases, the player is required to defend a specific location against waves of invaders, which entails the survival of all Essential NPCs. The timing between waves and their composition are the main influences on difficulty, although it is also possible to make the defenders weaker or stronger as needed.

Another factor that greatly influences perceived difficulty is the sequencing of the waves. If a wave can begin before the last one finishes, that is, before all of its constituent NPCs have been defeated, the player may have to face the remnants of multiple waves simultaneously. A time limit on this type of quest may seem redundant, seeing as the waves are themselves timed, but by disallowing waves to begin simultaneously the total time of engagement would be determined solely by player performance, making a time limit on quest completion a viable difficulty parameter.

Table 12: Defense Quest Parameters

| Range | Component | Type |
|---|---|---|
| [1, ∞[ | Essential NPC | NPC |
| | Spawn | Spawn |
| [0, ∞[ | Defender NPC | NPC |
| | Spawn | Spawn |
| [0, ∞[ | Wave NPC | NPC |
| | Spawn | Spawn |
| | Spawn Time | Number |
| | Quantity | Number |
| | Wait for Completion | Boolean |
| [0, ∞[ | Collateral NPC | NPC |
| | Spawn | Spawn |
| [0, 1] | Time to Finish | Number |
| [0, 1] | Final NPC Location | Location |

## 5. Example Analysis

The following is an example analysis of the quest "*The Left Piece of Lord Valthalak's Amulet*", a quest for level 60 players from World of Warcraft. The quest description states:

"*Use the Brazier of Beckoning to summon forth Jarien and Sothos and slay them. Return to Bodley inside Blackrock Mountain with the Left Piece of Lord Valthalak's Amulet and the Brazier of Beckoning*."

From the walkthrough available on WowHead, we find that the item "*Left Piece of Lord Valthalak's Amulet*" is obtained after killing the Boss NPCs Jarien and Sothos, which in turn are summoned with the use of the item "*Brazier of Beckoning*" provided at the beginning of the quest.

The quest can therefore be understood as a composition of an Interaction (Scenery) Quest and a Boss Kill Quest, according to our proposed classification. To confirm this hypothesis, the parameters that compose quest difficulty must be identified and compared to the parameters of both classes. In this particular instance, no specific Collateral NPCs are mentioned, but their presence can be assumed, as the quest takes place inside a dungeon, a region of the game world that by definition holds a large number of aggressive NPC. Tables 13 and 14 present the values of each component in the class.

Table 13: Interaction (Scenery) Components

| Range | Component | Value |
|---|---|---|
| 1 | Scenery Object | *Chamber of Summoning* |
| | Spawn | *Center of the Chamber* |
| | Item Needed | *Brazier of Beckoning* |
| N | Collateral NPC | *Unspecified* |
| | Spawn | *Unspecified* |

Table 14: Boss Kill Components

| Range | Component | Value |
|---|---|---|
| 1 | Target NPC | *Jarien* |
| | Spawn | *Unspecified* |
| 1 | Target NPC | *Sothos* |
| | Spawn | *Unspecified* |
| N | Collateral NPC | *Unspecified* |
| | Spawn | *Unspecified* |
| 1 | Final NPC Location | *Blackrock Mountain* |

## 6. Parameter Conflicts

A note must be made with regard to conflicts that arise from the simultaneous adjustment of quest elements, as quest variants of different difficulties must coexist in a shared virtual world between all players, a consequence of the very nature of MMORPGs. The main cause for conflicts is seen with regards to the NPC components, as the interaction between player characters and NPCs can last up to a few minutes during combat.

When players on the same quest simultaneously interact with an adjusted NPC – by attacking a common enemy, for example – a balancing mechanism must accommodate numerous sets of adjusted difficulty parameters to a single game element. Choosing one specific player, however, has a direct impact on the difficulty of the quest for all others.

The solution which best satisfies this dilemma is dependent on how the interaction between players and NPCs is implemented by the game. In games that don't allow for simultaneous interaction of different players with the same enemy, such as Atlantica Online, the solution can be as simple as adapting the conflicting NPC to the player with whom it is currently interacting. In most cases, however, this is not an option, as the majority of games do not present such a restriction. For those cases, other solutions must be envisioned.

One such possibility is to simply lock the parameters of an NPC whenever it interacts with a player. This solution can be used "as is" in most games that implement a target detection mechanism, also known as a target lock. In this instance, when two players interact with the same NPC, the parameter set will be locked to the NPCs most recent target. The same solution can be adapted to use instead any heuristic function tailored to the particular needs of a game. The problem with this approach is that although there is no longer the possibility of a higher-leveled NPC attacking a lower-leveled character, the reverse situation can be exploited by a group of players to force a reduction on the level of an NPC simply by making sure it is locked to the lower level player in the group.

A third alternative is to maintain distinct instances of the problematic NPC for each player, and resolve NPC interaction conflicts independently for each one. In this case, the result of an attack by one character on the NPC is propagated to all existing instances for all players proportionally to their levels. This approach presents no drawbacks from the perspective of the game mechanics, but reduces apparent power differences between player characters, which may defeat the purpose of progression in the game.

Other conflicts involving scenery objects, items, or locations can be resolved in a similar fashion. The exact conflict resolution scheme to be chosen depends, however, on game mechanics and controls specific to each case.

## 7. Conclusion and Further Work

This paper presented a new model for quests in massively multiplayer online games, with an approach specifically geared to facilitate quest design and DDA processes. The quest classification proposed bridges the gap between quests and game elements, with emphasis on difficulty manipulation.

The comprehensive difficulty parameters identified for all MMORPG elements can be used “as is” to assist in the construction and structuring of quests, but become of great value when considered from the perspective of a game designer to create quest variants of different levels. Potential problems and conflicts that may arise from the use of these parameters in a dynamic environment can be solved in a number of different ways, some of which were presented in previous sections of this paper, but which can easily be expanded with a focused study in conflict resolution.

To better validate the identified parameters and expand on the work done here on quest analysis, additional case studies involving different games, or a larger quest sample could create a better foundation for the future conception of DDA processes tailored to MMORPGs.

Future works might also include a method for the detailed analysis of the influence each parameter has on quest difficulty, by means of prototypes and usability studies. Creating precise metrics to automatically adjust quest difficulty is a complex venture, as different parameters when considered in conjunction can wield wildly different difficulty levels than when taken separately. The composition of a quest variant has not, therefore, the same difficulty as the simple composition of its constituent elements.

Given that MMORPGs make up a relatively new genre, and that the field of quest modeling is in its infancy, little has been proposed to tackle the specific challenges brought by the genre, such as the persistent game environment and the multiplayer aspects, emphasizing the importance of the novel approach presented here.

Table 15: Proposed Classification

| WowWiki | Pita et al. [2007] | Doran et al. [2010] | Proposed Classification |
|---|---|---|---|
| Kill | Kill | Kill pests | **Kill** |
| | | Kill enemies | |
| | | Attack enemy | **Boss Kill** |
| | | Revenge, Justice | |
| | | Capture Criminal | |
| Loot | Steal | Steal stuff | **Loot** |
| | | Steal supplies | |
| | | Steal valuables for resale | |
| | | Recover lost/stolen item | |
| Loot Container | Retrieve | Obtain rare items | **Gathering** |
| | | Obtain luxuries | |
| | | Gather raw materials | |
| | | Obtain training materials | |
| Explore | Discover | Visit a dangerous place | **Exploration** |
| | | Spy | |
| Make | | Use existing tools | **Interaction (Item)** |
| | | Assemble repair | |
| | | Treat or repair | |
| | | Make valuables for resale | |
| | | Assemble tool for new skill | |
| Interact | | Research a skill | **Interaction (NPC)** |
| | | Practice combat | |
| | | Practice skill | |
| | | Check on NPC | |
| | | Interview NPC | |
| Deliver | | Deliver item for study | **Delivery** |
| | | Deliver supplies | |
| | | Trade for supplies | |
| Escort | | Guard Entity | **Defense** |
| | | Attack threatening entities | **Escort** |
| Unique | | Create Diversion | **Interaction (Scenery)** |
| | | Assemble fortification | |
| | | Rescue captured NPC | |

Table 16: Basic Quest Components

| Component | Parameters | Type | Description |
|---|---|---|---|
| NPC | Attributes | Collection | Implementation-specific collection of attributes that define the combat system. |
| | Skills | Collection | Implementation-specific collection of NPC skills and abilities. |
| | Aggressive | Boolean | Flag indicating if the NPC attacks the players or their allies on sight. |
| | Reactive | Boolean | Flag indicating if the NPC responds to player attacks. |
| Object | Activatable | Boolean | Flag indicating if the object can be activated by player interaction. |
| | Consumable | Boolean | Flag indicating if the object disappears after being interacted with. |
| | Visible | Boolean | Flag indicating if the object is visible in the game world. |
| Item | Consumable | Boolean | Flag indicating if the item disappears after being used once. |
| Location | Point | Number | Absolute or relative position on the game world. |
| | Known/Unknown | Boolean | Flag indicating if the location is provided to the player as part of the quest. |
| Spawn | Location | Location | Instance of Location. |
| | Frequency | Number | Frequency by which the spawn point produces a specific entity. |
| | Max Number | Number | Maximum number of entities supported by this spawn point. |
| | Dispersion | Number | Indicative of dispersion of spawned entities, or their relative proximity. |
| | Radius | Number | Defines the area covered by the spawn point. |

## References

MULLIGAN, J. AND PATROVSKY, B. 2003. Developing Online Games: An Insider's Guide. *New Riders*, 474-477.

WU-CHANG, F., BRANDT, D. AND DEBANJAN, S. A Long-Trem Study of a Popular MMORPG. *In: Communications of NetGames'07. Melborne, Australia., 2007.* 19-24.

CCP GAMES. *EVE Online - FAQ.* [online] Available from: www.eveonline.com/faq/faq_01.asp. [Accessed 2 December 2011].

WOWWIKI. *VEHICLE Quests*. [online] Available from: www.wowwiki.com/Vehicles_(mechanic). [Accessed 2 December 2011].

RAPTORAK. *MMORPG's – Grinding vs Questing. Teach me to Game* [online]. 2011. Available from: www.teachmetogame.com/blog/mmorpgs-grinding-vs-questing/. [Accessed 2 December 2011].

BAILEY, C. AND KATCHABAW, M. *An Experimental Testbed to Enable Auto-dynamic Difficulty in Modern Video Games.* Department of Computer Science, University of Western Ontario, Ontario, Canada., 2005.

DORAN, J. AND PARBERRY, I. *Towards Procedural Quest Generation: A Structural Analysis of RPG Quests.* Technical Report LARC-2010-02, Laboratory for Recreational Computing, Dept. of Computer Science & Engineering, University of North Texas, 2010.

DORAN, J. AND PARBERRY, I. A Prototype Quest Generator Based on a Structural Analysis of RPG Quests. *In: Proceedings of the Second International Workshop on Procedural Content Generation in Games. Bordeaux, France.,* 2011.

PFEIFFER, B. AI to Control Pacing in Games. *In: Proceedings of the IC2 GameDev Workshop, University of Texas, Austin., 2003.*

HUNICKE, R. AND CHAPMAN, V. AI fot Dynamic Difficulty Adjustment in Games. *In: Challenges in Game Artificial Intelligence AAAI Workshop. San Jose, 2004.*

ANDRADE, G.R. AND CORRUBLE, V. Automatic Computer Game Balancing: A Reinforcement Learning Approach. *Game Developer*, 2005.

P. DEMASI AND A. CRUZ. Online Coevolution for Action Games. *In: Proceedings of The 3rd International Conference on Intelligent Games And Simulation*. London. 2002, 113–120.

PITA, J., MAGERKO, B. AND BRODIE, S. True story: dynamically generated, contextually linked quests in persistent systems. *In: Proceedings of the 2007 conference on Future Play, New York, USA,* 2007. 145-151.

WOWHEAD. *Quests – World of Warcraft* [online]. Available from: www.wowhead.com/quests [Accessed 3 June 2012].