Mining Game Statistics from Web Services: A World of Warcraft Armory case study

Chris Lewis, Noah Wardrip-Fruin
University of California, Santa Cruz
1156 High St, Santa Cruz, California, USA
{cflewis,nwf}@soe.ucsc.edu

ABSTRACT

Collecting large sets of quantitative video game play data can take many months or years. This delays the progress of interpreting data and drawing interesting conclusions. Mining game data from publicly accessible web services allows us to quickly retrieve quantitative results. This will allow the pace of quantitative research in video games to increase, as well as provide pointers towards maximizing the efficiency of future qualitative study.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

Keywords

video game, world of warcraft, data mining, web service

1. INTRODUCTION

Video game research can take many forms and be conducted from many viewpoints and approaches. One common requirement is the collection of *data*, be it quantitative or qualitative.

The process of observing and collecting data from game play can be arduous. Wood et al. [16] identified four methods of collecting data on games: survey-based studies, online testing, participant observation and online interviews. All of these methods involve some form of participant recruitment, which the authors note is, "One of the biggest problems with any kind of social science research." In the specific realm of Massively Multiplayer Online Games (MMOGs), such as *World of Warcraft*, a common technique for researchers is to create an avatar and conduct in-world research by hand, observing others or conducting surveys [8, 10]. These techniques are the only way to approach inherently qualitative questions and are useful for small investigations. However, they become more and more time-consuming and impractical as the sample size increases.

FDG 2010 June 19-21, Monterey, CA, USA

In order to find larger data sets quickly, we can look to the substantial data about game play accessible on the Internet. Several video game developers offer statistics about players on their web sites or through an online Application Programming Interface (API). We will informally refer to both methods of delivering information over the Web as "web services."

The most prominent of game web services are the *World of Warcraft Armory* by Blizzard Entertainment [1] (henceforth referred to as 'the Armory') and Bungie Online's *Halo 3* statistics [2]. Both offer data for individual characters, allowing players to visit the site to learn about their own achievements as well as compare themselves with others. It is likely that the popularity of such services will grow, as gaming moves further into a continuously network-connected environment.

Such online databases are not available to be downloaded in their entirety, preventing researchers from performing any large-scale data querying or analysis. In this paper, we present a web crawler, dubbed "WoWSpyder" that queries the Armory page-by-page and stores the results into a local database. This data can then be queried at will, allowing researchers to quickly find the level and class of most of the *World of Warcraft* population, the items they equip, and the talents they choose, as well as delve into 330 other statistics. These statistics range from how many times a player dies, how many quests they have completed or abandoned, or even how many virtual hugs they have given other characters. In this paper, we specifically target common sources of debate for *World of Warcraft*, such as which class advances most quickly through the game or which class dies more often. While providing interesting quantitative data, this data set can also aid the identification of important players whom may be useful to qualitatively survey, which should make executing qualitative studies more efficient.

The paper explores this potential, also including a broader discussion of the process of identifying and collecting Web-based data and how they can be used to empirically answer questions from the meta-discussion that surrounds popular games. We include topics such as analyzing the underlying data representation, how to navigate through the web servers and whether such a practice is even permissible by developers. We conclude by presenting the results from a preliminary data exploration, where we successfully predict characters based on what they are wearing, show that classes level at the same pace, that some classes die less than others, illustrate popular items, as well as highlighting other possible research avenues.

2. RELATED WORK

To our knowledge, Williams et al. [15] are the researchers who have been granted access to the largest data set collected by developers. Sony Online Entertainment granted access to *Everquest 2* logs, resulting in a dump of 60 terabytes. They have stated that they are investigating social networks, the demographics of players and analyzing the game's churn rate (the proportional loss of subscribers) [11].

However, Williams describes the work as an example of "be careful what you wish for." Such a large size required Williams and his team to purchase $70,000 worth of computer equipment to be able to store and query the data, with the total expenditure for equipment and researchers reaching $1.5 million over three years [13]. A web crawler is designed to operate on a single machine, collecting gigabytes, rather than terabytes, of data. While this means that crawlers like WoWSpyder do not provide "complete" results, as Williams et al. can, they are significantly cheaper and quicker to deploy.

Drachen & Canossa, in collaboration with EIDOS, have presented case studies on instrumenting games from IO Interactive [3] and performed player modeling by data mining player logs of *Tomb Raider: Underworld* [4]. In particular, their discovery of four clusters of *Tomb Raider* players, labeled as Veterans, Solvers, Pacifists and Runners, validates data mining as an approach to game data that identifies patterns that were otherwise invisible.

While gaining access to developer's internal databases, as Williams et. al and Drachen & Canossa have done, removes the traditional data collection hurdles, it requires strong industry relations that many researchers do not have. In addition to making these connections, a significant amount of effort is also required from the developer to collect the data and remove private information. In contrast, web crawlers allow researchers to act independently, as well as removing any additional burden from game companies.

The PlayOn@PARC research project, with regular members Ducheneaut, Yee, Nickell and Moore, remains the most influential quantitative research of *World of Warcraft* [5, 14, 6]. Their work used a bot embedded in the game world, running queries of in-game information to create a "census" of the players. This recorded information such as the number of characters, which class and race they were, and how long they were taking to level. WoWSpyder differs because it relies on public information, whereas the in-game approach is able to find information that is not necessarily publicly shared. Each approach will provide different data sets with information that one or the other will not have, but we believe that mining web services is a quicker and more reliable method of collection in the general case.

Yee has been notable in virtual worlds research for The Daedalus Project [17]. Yee used online surveys to gather player data and draw conclusions about the motivations of online game players [18]. His work required participant recruitment, whereas crawlers take data published from the game developers directly, so do not require participant recruitment to function.

3. METHODOLOGY

3.1 When to Begin

We believe the time that data is collected affects the data collected, reflecting how the player base modifies itself to address different high-level objectives. In our *World of Warcraft* example, just after the *Wrath of the Lich King* expansion (soon after Patch 3.0), we would expect to see very few characters equipping faction tabards as they are only for players who have reached level 80. Similarly, Patch 3.2.2 rebalanced the dragon boss "Onyxia" in the game, and we may expect to see more characters wearing fire-resistant items in order to combat her. It is therefore important to apply the correct interpretative frame to any data set.

This presents an interesting obstacle: when to begin collecting data. It seems that collection should ideally occur between regular content patches, in order to try and find the population at some kind of equilibrium. However, starting collection too early may lead to the population still reconfiguring to the changes introduced, while starting too late may mean a new patch is released before enough data is collected.

Further work could compare and contrast results from different time periods.

```
XHTML:
<div class="charNameHeader">Moulin<span class
    ="suffix"> the Explorer</span>
</div>
<a class="charGuildName" href="/guild-info.xml
    ?r=Ravenholdt&amp;gn=Beasts+of+Unusual+
    Size">Beasts of Unusual Size</a><span
    class="charLvl">Level 80 Blood
    Elf Death Knight</span>

XML:
<character class="Death Knight" level="80"
    name="Moulin" guildName="Beasts of Unusual
     Size" race="Blood Elf" suffix=" the
    Explorer"/>
```

Figure 1: Actual output formats of the Armory (XML edited to only contain the same information as the XHTML snippet).

3.2 Analyzing querying suitability

In order to begin downloading the data from a web service, we must first ascertain its technical suitability for crawling. Several factors play into this. Here we look at the two most important: the structure of the data and seeding a search function.

3.2.1 Data structure

There are many ways data can be queried for online. Web browsers query for and display Hypertext Markup Language (HTML) or Extensible Hypertext Markup Language (XHTML), markup languages that express the content of a web page, often commingling information specific to web renderers with the other material. In contrast, Extensible Markup Language (XML) is the common output format for data accessed via APIs, as it allows the definition of any tags the author requires, making it a flexible method that can specify data for many different domains. In addition, as XML is usually only used for specifying data, parsers that convert XML data to programming language objects can be written in a matter of hours.

Figure 1 has an example of the different formulations of validly expressing a *World of Warcraft* character and some attributes. Notice how much more simple the XML expression is in comparison to the XHTML.

However, all these benefits doesn't mean our task is impossible in XHTML, as we can still see a structure within it. For example, we know that a guild name is wrapped inside a `<a class="charGuildName">guild_name_here</a>` expression. This means we can successfully write a parser, although not as simply as a purely XML data stream. This technique is also more fragile, as changes to the layout of the page may break the parser. XML feeds are less likely to be edited as they express the data and its structure, not the presentation.

The Armory outputs both XML if accessed with Mozilla Firefox and XHTML to other clients. Other developers offer API access that deliver XML to all, such as the *Lord of the Rings Online* (*LOTRO*) API [12].

3.2.2 Search seeding

With the Armory, there is no functionality to allow one to retrieve a list of all characters on a server. One can only search by name for guilds or characters. This makes the task of downloading characters difficult. If a list of characters cannot be obtained, how is it possible to query each one and download the data?

In our case, we were able to find character names by searching through the rankings of teams entered into the *World of Warcraft* Arena tournament, a Player versus Player (PvP) competition that is held in-game. All currently participating teams are listed in these rankings. We can download the teams, find which characters make up those teams, then use the guilds that the characters are in as our search seeds. We can then list all characters in those guilds and download them to our database.

Unfortunately, this has drawbacks. The first is that all characters without a guild, or in a guild that has no arena participants, will never appear in our data set. With no other method to gain an authoritative list of guilds on a server, there is no way of verifying how many players and guilds are covered. This may adversely affect non-PvP servers whose players may favor not engaging in PvP combat, and thus have a higher chance of being in a guild without an arena participant. This limits the questions that can be asked of the data, but our anecdotal evidence from playing *World of Warcraft* suggests that the majority of players are members of guilds with at least one arena team. Also, for many research questions, missing such players is inconsequential, and there are far more characters available than could ever be realistically downloaded by a crawler.

More troubling is that this is a specific workaround for the Armory only, and other techniques would need to be employed to function for other web services. That being said, for many popular games, we do not believe this is necessarily a difficult problem. We found many guild listing web sites for *LOTRO* which could be downloaded and used as seeds for the *LOTRO* API, while players for *Halo 3* could be found by crawling posters to the Bungie *Halo* forums.

3.3 Building a crawler

3.3.1 Overview

Once a web service has been deemed suitable for crawling, we must now begin the task of building our crawler. WoWSpyder is used as a case study, and we illustrate the general work loop: download XML, convert data to a usable form and save it, find a new page to download, repeat.

3.3.2 Downloading XML

Downloading content from a web service is a fairly simple task in most modern programming languages. However, doing this in a scalable fashion is not. When downloading from a web service, it is considered customary for a crawler to wait for a couple of seconds between requests. Anything more than this may be flagged as an abuse and automatically banned by the web server. With the Armory, Blizzard's servers will return error codes to a client that is accessing the server too quickly. This means that the crawler will spend the vast majority of its time waiting, rather than performing any useful computation.

We circumvent the banning by containing separate downloader instances in 20 worker threads. Each downloader then retrieves an individual session ID from the Armory, which each downloader then uses for their requests. This gives the appearance of 20 different clients accessing the service. A central function mediates the task of downloading a page, by finding a thread that is ready to download. The thread is requested to download and then return the content. Afterwards, the thread sleeps for a couple of seconds. This allows the program to simultaneously perform operations that require computation, such as parsing and storage, while accessing the web server and waiting when necessary.

When the Armory sends an error code to one thread, all threads are blocked for between 30 seconds and 3 minutes to responsibly back off from the server. We also mitigate the amount of traffic required from the server by caching all downloaded pages for an hour, so any subsequent queries of the page does not result in a new request.

3.3.3 Data conversion

Once a page is downloaded, it needs to be converted to a usable data format. For XML files, it is possible to interact and query them directly using a technique known as XML Path Language (XPath). For the purposes of this paper, we will assume that the reader wishes to convert the XML to a different form.

Most modern programming languages, such as Python or Java, come with XML parsers built-in, changing the XML representation to a native programming language object with minimal coding. The values of these XML objects are then assigned to a native object, ready to be synchronized to a database.

With any meaningfully large data set, saving to a database is our recommended method of ensuring *persistence* (ie. the data is saved when the program is quit). Databases are optimized for the storage, retrieval and integrity (correctness) of data, which suits our purposes perfectly. The data can then be queried directly with a database package or interfaced with by popular tools such as SPSS.

3.3.4 Creating a new URL to download

WoWSpyder chooses new URLs to download by analyzing the data it wishes to request, then creating the URL automatically, directly querying the Armory servers. For example, when downloading characters listed on guild pages, we know the name of the character, the server the guild is on,

as well as the guild's geographic market, such as the US or Europe. This is all the information we need to create the URL for a character. For example, in order to download Moulin from the Ravenholdt server in the US, we create the URL `http://www.wowarmory.com/character-sheet.xml?r=Ravenholdt&n=Moulin`. Here, we see that the domain for the US Armory is `http://www.wowarmory.com`, compared with Europe's `http://eu.wowarmory.com`. The section `r=Ravenholdt` is where we specify the Ravenholdt realm, and `n=Moulin` is where we ask for the character with the name "Moulin". All URLs on the Armory can be formulated in the same way. The same method is applicable to the *LOTRO* API, as well as the Bungie Online service.

4. RESULTS

4.1 Overview

In order to illustrate the effectiveness of downloading statistics from game web services, as well as provide examples of the possible types of querying available, we present some the results from some initial investigations into our Armory data. These results are by no means an exhaustive representation of the depth or breadth of the data. With the sheer amount of statistics offered by the Armory, there is the possibility to spend months generating new results from this data set.

It is important to note that this data is exactly as retrieved from the Armory, which we believe includes errors. For example, after calculating the number of deaths in Player versus Environment (PvE) of a character, some characters were reported as having negative PvE deaths. Our data filters characters whose statistics are impossible (such as negative deaths), but we do not make any filter data that is feasibly questionable, indicating a player skill or effort simply too great to be considered realistic. Many of the lower outliers in Figures 2 and 3 we believe are questionable, but without strong evidence to discount them, we have kept them in the data set.

The *World of Warcraft Armory* contains a vast array of data on characters. One can query for:

- items a character is wearing
- a character's stats, such as strength, stamina, and so on
- reputation with other factions
- the achievements they've performed
- the talents chosen
- a wealth of miscellaneous statistics including how much damage has been dealt, which food stuff the character has eaten most, and even how many times the character has laughed or hugged another character

This rich set of features allows us to gather plenty of data, letting us run useful data explorations from high-level results across the entire player base all the way down to pinpointing single characters of interest.

Our sample takes into account 136,047 characters from the United States and Europe. A breakdown of the server types can be seen in Table 1.

Table 1: This table shows the distribution of characters by server type and site.

Server Type	US	EU
PvE	61393	14448
PvP	10162	22934
RP-PvE	14624	3928
RP-PvP	4196	4362
Total	90375	45672

This data was collected from 16$^{\text{th}}$ April 2009 to 19$^{\text{th}}$ August 2009, which means our collection started just after the release of Patch 3.1 and ended just after the release of Patch 3.2.

4.2 Class itemization

Class itemization refers to how players of different classes equip items. The designers of the game often specify which items should be worn by each classes by affecting the type and the statistics given.

Weapons could be a sword, axe, gun, mace or other type, and only certain classes will be able to wield them. A mage, for example, is unable to equip a sword. Armor, as with weapons, have different types, which could be cloth, leather, mail or plate. Again, our fragile mage is limited to only being able to wear cloth, whereas a strong warrior is able to wear all types of armor.

Certain statistics support certain methods of play: a Rogue may favor agility to inflict more critical hits, while a warrior will want to stack up on stamina, in order to build up their health and survive fights.

These norms are validated in the design with the construction of "armor sets" which are designed for each class when they are at their maximum level. The Lightforge set, designed for Paladins, offers high intelligence and stamina, which support the survivable healer class role.

To investigate whether the class itemization game design choices are actually being heeded by players, we constructed a Naïve Bayes classifier that predicts the class of a player based on the items they are wearing. The model is then ten-fold cross validated using a stratified sample, ensuring that the model works correctly across a sample with a representative class distribution. Our classifier achieved a classification accuracy of 94.70%. The results of the classifier are broken down class-by-class in Table 4.2.

Of note is that when the classifier does classify incorrectly, it classifies incorrectly within armor type groups. Mages, Warlocks and Priests are the most frequent to be confused with one another, and these classes are all restricted to wearing cloth armor. Likewise, Shaman, Druids, Rogues and Hunters are confused with one another, and these classes wear leather (although Shaman and Hunters can wear Mail after level 40).

When analyzing misclassifications directly, it is often the case that the character is wearing generic items that do not provide any insight into the character, or that the character is wearing items that often define other classes. For example, we found a Rogue equipped with a "Venomstrike" longbow, popularly worn by Hunters (106 Hunters, 54 Rogues and 37 Warriors were equipped with the "Venomstrike" in our sample). Future work could analyze whether a human ex-

	Druid	Warlock	Mage	Rogue	Paladin	Shaman	Warrior	Priest	DK	Hunter	Precision
P: Druid	**13286**	19	14	25	23	208	21	8	30	32	97.22%
P: Warlock	25	**8859**	980	5	20	14	6	831	4	4	82.42%
P: Mage	28	1407	**11153**	3	9	14	4	948	3	6	82.16%
P: Rogue	61	2	2	**11969**	6	9	7	0	0	82	98.61%
P: Paladin	0	0	2	1	**16882**	11	262	2	25	2	98.23%
P: Shaman	129	0	0	6	36	**10454**	3	0	3	11	98.23%
P: Warrior	2	0	0	5	384	10	**12301**	0	16	12	96.63%
P: Priest	28	311	265	0	5	7	1	**10239**	0	0	94.32%
P: DK	0	0	0	0	144	2	70	0	**19561**	0	98.91%
P: Hunter	124	1	0	146	15	277	25	1	1	**14138**	95.99%
Recall	97.10%	83.58%	89.83%	98.43%	96.34%	94.98%	96.86%	85.12%	99.58%	98.96%	

Table 2: This is a confusion matrix for classifying WoW classes based on the items worn by a character. The columns specify actual classes, the rows specify the classes that were predicted.

pert is able to make significantly better predictions on the misclassified samples.

Our strong findings provide empirical evidence about *World of Warcraft*, demonstrating something about the game design that was hitherto only known by Blizzard themselves. As well as providing evidence about the game design, it also provides us with information about the players themselves. The vast majority are classified correctly, so we know that players are conforming to a game norm. Further qualitative study could identify players that don't conform, and ascertain their reasons, offering insight into the particular type of unconventional play and highlighting possible design avenues for future game development.

4.3 Class days to level 80

One of the most popular player discussions surrounding *World of Warcraft* is which class can reach level 80 the fastest. An informal survey performed by WoW.com showed much disagreement between players [7], which perhaps partly explains why the discussion has continued for so long. After each patch, often the discussion begins again, factoring in new balancing or abilities for each class. Often, this discussion is centered around the number of hours actually played on the character. However, this information is not available via the Armory. We were, however, able to look at the time to level 80 in chronological time.

Figure 2 shows a box plot of the leveling time for characters from 1 to 80. This survey comprises the 6244 characters in our sample that were created after *World of Warcraft's* 3.0 patch was deployed. This patch provided achievements, including the timestamped achievements for reaching level 10 and level 80. We calculate the time to level by subtracting the two timestamps. Characters that were started before the patch were discarded; the timestamps on their level 10 achievement are created on the day they logged in after the 3.0 patch, making it impossible to derive how long it took them. This also implicitly discard all Death Knights, who start at level 55.

Our box plot shows that no class has any significant level time advantage, the medians of all the classes hovering between 70 and 80 days. As with the median, the 5, 25, 75 and 95 percentiles, roughly match one another. This indicates that all classes are balanced in their leveling time, a result that few, including ourselves, would have predicted. This result also contradicts the information found in [5], which concluded priests leveled at a faster rate, normalized against playtime, than other classes. This disparity indicates that either Blizzard has rebalanced the game since that data was collected, or that there is a significant difference between measuring chronological time and actual play time.

Item name	Number wearing
Guild Tabard	19849
Mirror of Truth	14219
Sigil of the Dark Rider	11574
Sundial of the Exiled	7505
Tabard of the Kirin Tor	7163

Table 3: This table shows the most popular items based on number of characters wearing them.

4.4 Class deaths on way to 80

Using the same population of characters that reached level 80 since patch 3.0 that we used for Section 4.3, we decided to explore other metrics of class quality. Looking at the number of deaths on the way to 80 proved enlightening. Note that these deaths are only PvE, meaning that they only take into account deaths caused by AI controlled characters, not other players. We further filtered out deaths in instanced dungeons, as players who compete more often in dungeons are more likely to die due to their difficulty.

Figure 3 illustrates the findings of this population. Here we do see a marked difference between classes. In particular, the Paladin, Druid and Shaman classes have a similarly low median, with a tighter interquartile range than other classes. This shows either that these classes are designed to survive longer, or that the players of these classes are better skilled than those of the other classes.

The only problem with this calculation is that deaths when a character is at 80 are included in this count. Ideally, these deaths would be filtered. Filtering out the level 80 characters from this set leaves 1060 characters. This means some classes have too few characters, such as the Rogue with 65 characters, to draw any reasonable conclusions.

4.5 Most popular items

Unlike the longer-term data mining in Section 4.2, even fast statistical analysis can yield interesting results. Here we look at the most popularly worn items.

In Table 4.5, we see some surprises in just the top five items. The "Guild Tabard" in first place is an expected result, but very close to that is the "Mirror of Truth", a trinket for level 80 melee characters, as it boosts attack power. Two places below it is "Sundial of the Exiled" the spell caster version of the "Mirror of Truth", boosting spell power. If we combine them, they're even more popular than the guild tabard.

At third place, the "Sigil of the Dark Rider" is a trinket only given to Death Knights during their initial quests, starting at level 55. This indicates that there is either a substantially greater number of Death Knights in the game,

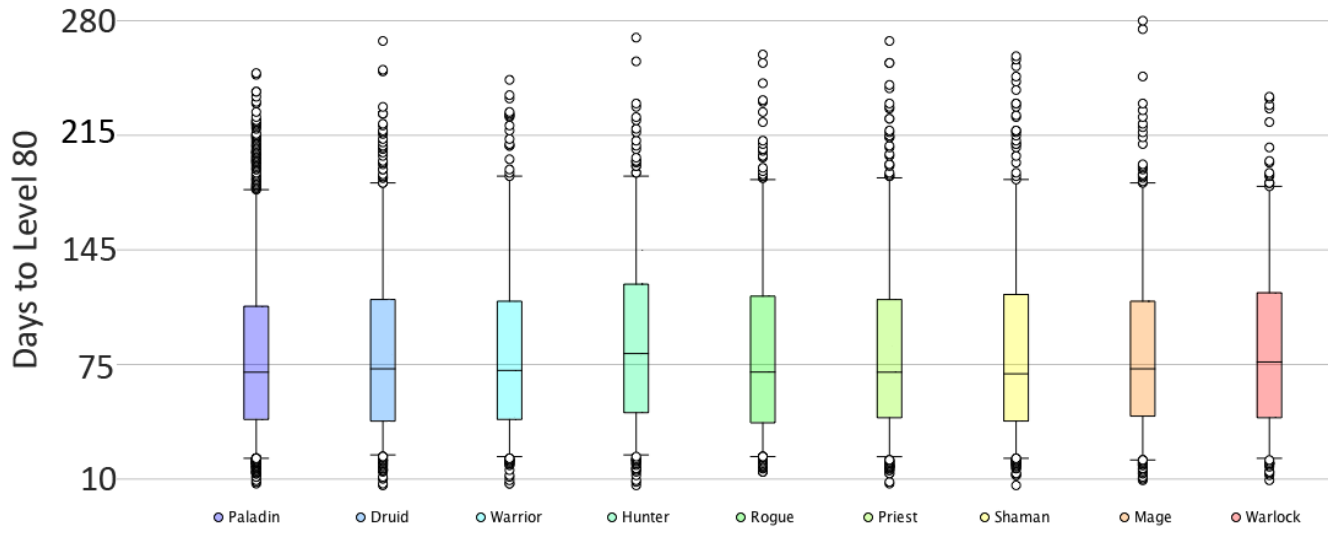

Figure 2: A box plot showing the chronological time for each class to level from 10 to 80. The boxes show the interquartile range, with the horizontal lines within them showing the median. The whiskers show the 5% and 95% percentiles, with circles representing outliers. We posit that lower outliers are errors in the Armory data.

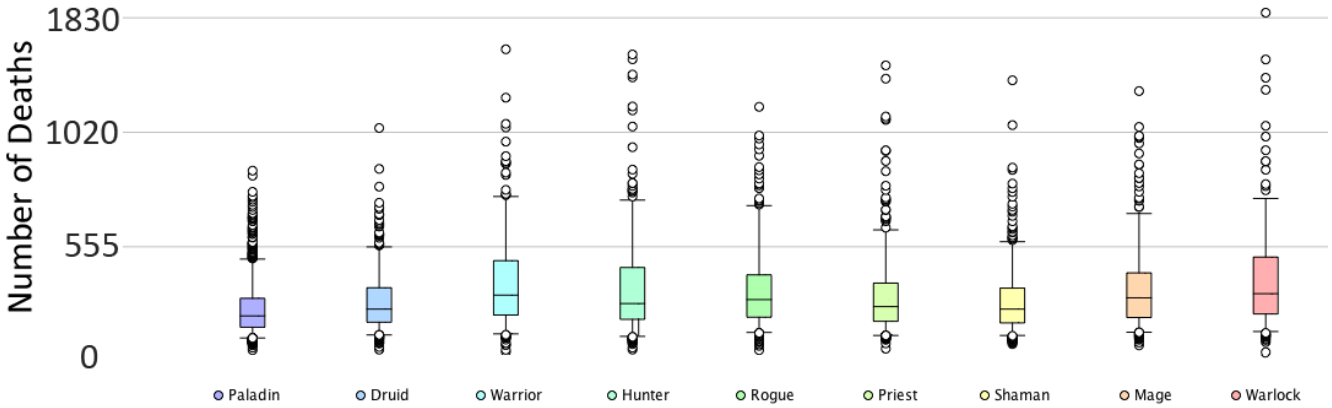

Figure 3: A box plot showing the number of deaths for characters that have leveled from 10 to 80 since Patch 3.0.

that many Death Knights were started but subsequently abandoned, or that the trinket is simply worth so much to players that they don't replace it. It is in fact, the latter option. Further analysis shows that while the drop-off of the trinket is high when Death Knights reach 80 (only 2052 of the 9756 level 80 Death Knights wear it), the numbers stay fairly constant per-level until that point.

The final item, the "Tabard of the Kirin Tor" is a faction tabard that provides reputation rewards for the Kirin Tor faction of the game, again a level 80 only item. Other tabards offer reputation gains for other factions, this tabard only narrowly edging out the "Tabard of the Wyrmrest Accord", which 6959 characters were wearing. The other two tabards, "Tabard of the Ebon Blade" and "Tabard of the Argent Crusade" had 5935 and 5412 wearers respectively. That no single faction is favored by players indicates that the design of this reputation system is mostly balanced.

Of interest is the distribution of items, illustrated in Figure 4, which follows a power law-esque distribution (please note that the chart is on a logarithmic scale for visibility purposes). The drop-off of the popularity of items is very strongly pronounced, cross-class items proving popular, then rapidly falling away to the long tail. This phenomenon has previously been discussed by Zardoz [19], who illustrated a similar distribution in a random sample of level 69 Death Knights. Zardoz posits that the most popular items, "represent a trade-off between how powerful the item is and how easy it is to get hold of." This assertion appears to hold for our sample across an entire player population, with the top items being cheap to purchase, but relatively powerful.

4.6 Outliers

In the box plots presented in Sections 4.3 and 4.4, the number of outliers is noticeable. Our large amount of data allows us to pinpoint specific characters of interest, who could then be the subjects of qualitative interviewing. The

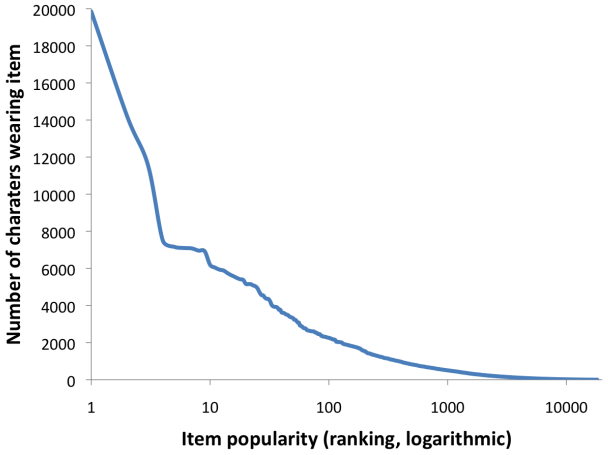

Figure 4: A chart illustrating the distribution of the number of times an item is worn by characters.

character with the most deaths in our sample had died over 3000 times in PvE combat. What motivates a player to continue playing when faced with such difficulties? Perhaps he or she actually sought out death? With our large database, it is simple to find outliers from many kinds of populations for many sorts of statistics. Understanding the psychological drivers behind outliers could yield important results.

5. DISCUSSION

5.1 Scalability

In Section 3.3.2, we described how a crawler can be threaded to masquerade as multiple parties. For the majority of our data collection, we were able to run WoWSpyder on a single client computer, communicating to a departmental database server. In testing, the client was able to download around 4000 characters every 24 hours. We found this more than sufficient for our purposes, allowing us to download over a 100,000 characters in a month on just one machine. Our actual data set (136,047 characters between 16th April 2009 and 29th August 2009) took longer to acquire as our crawler was often not running while in development.

However, despite the speed of WoWSpyder on one machine, we designed the crawler to be scalable to run on multiple machines. Should there be more clients available, running a crawler on a separate machine rapidly increases the statistics that can be downloaded every 24 hour period.

In order to scale the crawler up, it is important to identify how the task can be separated, to prevent each machine performing the same work. For WoWSpyder, we were able to split the work up by each realm. As each client is initialized, it randomly chooses an unlocked realm to work on, then locks it. Once the realm is finished, it is unlocked and a new one chosen. Different web services may require more complex locking, say at a guild or alphabetical level.

5.2 Legal implications

Neither author is a qualified lawyer, thus we cannot give legal advice. However, there are aspects of a web service that we would recommend crawler creators look at.

The first aspect is the Terms of Service (ToS). Many web

```
User-Agent: *
Disallow: /arena-ladder.xml
Disallow: /character-arenateams.xml
Disallow: /character-sheet.xml
```

Figure 5: The first four lines from the Armory `robots.txt` file, as of December 2009. The first line specifies that the file applies to all crawlers, the next lines specify files that should not be accessed.

services have a ToS in order to define the reasonable boundaries for how the data is accessed, who owns copyrights and any other disclaimers. It is important to note any clauses surrounding accessing and republishing of data.

Blizzard Entertainment have not specified any ToS on the Armory web site, so we consider the Armory data to be in the public domain.

The second aspect to consider is the presence of a `robots.txt` file on web sites. Only web *sites* specify these, typically web *services* that are only accessed in XML representations, do not. The `robots.txt` file is commonly used as a way of communicating to search engines which pages should be excluded from their search results, but in fact it has a broader definition of where automated scripts are not allowed to visit, a definition which includes WoWSpyder. An extract from the `robots.txt` file for the Armory is in Figure 5.

The Armory's `robots.txt` disallows all crawling of the site, meaning our crawler is ignoring Blizzard Entertainment's requests. Bungie Online's `robots.txt` similarly disallows crawling of their statistics. It is likely these are specified to stop search engines indexing millions of dynamically generated pages, rather than to stop individuals crawling their sites for non-commercial use.

Whether or not the ToS or `robots.txt` are legally binding, they are a communication of the wishes of the game developer. As academic researchers, there is a responsibility to avoid perceived irresponsible behavior that may cause access to be limited for future researchers or the public in general.

5.3 Ethical implications

5.3.1 Privacy

Good practice requires that data should be anonymized to the greatest possible level while still allowing the research questions to be answered. For most quantitative work, we believe it will be sufficient to anonymize character and guild names.

For this task, we recommend using a one-way hashing function. This is a mathematical function, common in computing cryptography, that creates a unique identifier from an input. For example, running "Moulin" through the MD5 hashing function results in the output `1bc29b36f623ba82aaf6724fd3b16718`. There is no companion function that can take the identifier and find the input, ensuring the anonymity of the character. The other useful property of hashing functions is that the input will always create the same output. This allows us to update the database by continuing to crawl the web service. When a character that was already saved in the database is down-

loaded, the hashing function on the name will be able to retrieve the original database record, preventing characters being inserted multiple times.

Guild names work similarly; one is able to identify individual guilds by their hash, but won't be able to derive what the actual guilds are on the server.

Anonymizing character and guild names will prevent the pinpointing of users for further qualitative research. This is why it is important to define the scope of research questions before building a crawler: this helps guide what data should be collected, and how it should be abstracted when it is stored.

5.3.2 Abuse of the web service

In this paper, we have mentioned how certain technical limitations created by the Armory can be circumvented. This is not to say that we condone working as an adversary to a web service, quite the contrary: abuse of a web service could lead to access being shut down, or developers becoming more unwilling to release information in the future.

Where possible, we have tried to ensure that we do not place undue strain on the Armory. For smaller companies with less capable technical infrastructure, it may be necessary to reduce accesses further.

However, abusing a service technically is but one side of the coin, and we must consider how work is published once it has been collected. Replicating the data set in its entirety for access to all would be a clear violation of a developer's wishes: if that were the case, the data sets would already be available for download. However, publishing quantitive abstracted data, such as that presented here, in an academic venue would be unlikely to be seen as inappropriate.

As with all scientific investigations that involve a human element, it is important to take into consideration the views and wishes of all stakeholders, and work such as this is no different. We believe that by taking care throughout the process, such work can be performed appropriately and ethically, contributing valuable results to the scientific discourse.

6. CONCLUSION

In this paper, we have demonstrated building a web crawler for video game statistic sites, using a crawler for the *World of Warcraft Armory* as a case study. We have shown that a large-scale sample that was previously inaccessible can be collected quickly and cheaply, and that this sample data can be queried for interesting results. These results are quantitative, but can lead to exciting subjects for qualitative study as well.

This approach has the potential to place video game data in the hands of far more researchers than was previously possible, leading to an expanded research community. We hope that this will improve the quality and quantity of research in this area, and look forward to the exciting results this could bring.

7. ACKNOWLEDGMENTS

The authors would like to thank Okoloth, European Kilrogg player, and maintainer of Armory Musings [9], for initial guidance and his battlegroups XML file, which we used under the Creative Commons Attribution 3.0 license. We would also like to thank T.L. Taylor and Lisa Galarneau for their feedback and comments on this paper.

8. REFERENCES

[1] Blizzard Entertainment. The World of Warcraft Armory. `http://www.wowarmory.com/`.

[2] Bungie. Bungie Online. `http://www.bungie.net/Online/Default.aspx`.

[3] Drachen, A., and Canossa, A. Analyzing user behavior via gameplay metrics. In *FuturePlay 2009* (2009), pp. 19–20.

[4] Drachen, A., Canossa, A., and Yannakakis, G. N. Player Modeling using Self-Organization in Tomb Raider: Underworld. In *CIG2009* (2009).

[5] Ducheneaut, N., Yee, N., Nickell, E., and Moore, R. J. Building an MMO with mass appeal: A look at gameplay in World of Warcraft. *Games and Culture 1*, 4 (October 2006), 281–317.

[6] Ducheneaut, N., Yee, N., Nickell, E., and Moore, R. J. The life and death of online gaming communities: a look at guilds in World of Warcraft. In *CHI '07* (2007), pp. 839–848.

[7] Harper, E. Ask Twitter: What's the fastest class to level? `http://bit.ly/uvRww` [cited August 12, 2009].

[8] McKee, H. A., and Porter, J. E. Playing a good game: Ethical issues in researching MMOGs and Virtual Worlds. *International Journal of Internet Research Ethics 2*, 1 (February 2009), 5–37.

[9] Okoloth. Armory musings... `http://armory-musings.appspot.com`.

[10] Taylor, T. L. *Play Between Worlds: Exploring Online Game Culture*. The MIT Press, April 2009.

[11] Timmer, J. Science gleans 60TB of behavior data from Everquest 2 logs [online]. February 2009. `http://bit.ly/6M6BDk`.

[12] Turbine. data.lotro.com, 2009. `http://data.lotro.com/`.

[13] Williams, D. The perils and promise of large-scale data extraction. Available at `http://dmitriwilliams.com/research.html`, 2010.

[14] Williams, D., Ducheneaut, N., Xiong, L., Zhang, Y., Yee, N., and Nickell, E. From tree house to barracks: The social life of guilds in World of Warcraft. *Games and Culture 1*, 4 (October 2006), 338–361.

[15] Williams, D., Yee, N., and Caplan, S. Who plays, how much, and why? Debunking the stereotypical gamer profile. *Journal of Computer-Mediated Communication 13*, 4 (September 2008), 993–1018.

[16] Wood, R. T. A., Griffiths, M. D., and Eatough, V. Online data collection from video game players: Methodological issues. *CyberPsychology & Behavior 7*, 5 (2004), 511–518.

[17] Yee, N. The Daedalus Project. `http://www.nickyee.com/daedalus/`.

[18] Yee, N. Motivations for Play in Online Games. *CyberPsychology & Behavior 9*, 6 (2006), 772–775.

[19] Zardoz. The median is the message. `http://bit.ly/3cEtb` [cited August 12, 2009].