

# **Webservice zur Auswertung und Analyse von WoW Combat Logs**

Diplomarbeit

Technische Universität Darmstadt  
Fachbereich Informatik  
Fachgebiet Knowledge Engineering

**Vorgelegt von** Björn Hirsch

**Betreuer & Prüfer** Prof. Dr. Johannes Fürnkranz

## Kurzfassung

In dieser Arbeit wird die Entwicklung eines Webservices für die Auswertung und Analyse von Combat Logs des MMORPGs World of Warcraft® beschrieben. Umgesetzt wird dies mittels PHP und MySQL. Die benutzte Datenstruktur und dessen Speicherung in der Datenbank werden ausführlich beschrieben. Die visuelle Präsentation der extrahierten und aufgearbeiteten Informationen ist ein Schwerpunkt der Arbeit, dabei werden auch die Alleinstellungsmerkmale gegenüber anderen Webservices hervorgehoben.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Motivation .....</b>	<b>5</b>
<b>2</b>	<b>World of Warcraft.....</b>	<b>7</b>
2.1	World of Warcraft zu Anfang .....	7
2.2	World of Warcraft auf maximaler Stufe .....	7
2.2.1	Gemeinschaft.....	8
2.2.2	Teams.....	8
2.2.3	Begegnungen.....	8
2.2.4	Ausrüstung.....	8
2.2.5	Instanzen .....	9
2.2.6	Klassen .....	9
2.2.7	Ressourcen.....	9
2.2.8	Fähigkeiten .....	9
2.2.9	Zustandsveränderungen .....	10
2.2.10	Talente.....	10
2.2.11	Rollen.....	11
2.2.12	Streben nach Verbesserung.....	13
2.3	Erklärung der Terminologie .....	14
<b>3</b>	<b>Combat Log.....</b>	<b>16</b>
3.1	Allgemeines .....	16
3.2	Struktur des Combat Logs.....	17
3.2.1	Basis.....	17
3.2.2	Präfix.....	21
3.2.3	Suffix.....	24
3.3	Beispiele .....	27
3.4	Beschränkungen .....	28
<b>4</b>	<b>Nachbearbeitung und Datenstruktur/Datenbanken.....</b>	<b>30</b>
4.1	Informationen vom Parser .....	30
4.1.1	Datenbanktabelle event.....	32
4.1.2	Datenbanktabelle player.....	32
4.1.3	Datenbanktabelle spell.....	32
4.2	Statische Informationen.....	32
4.2.1	Datenbanktabelle static_encounter .....	33
4.2.2	Datenbanktabelle static_spell_classtree.....	33
4.3	Informationen durch Nachbearbeitung .....	34

4.3.1	Zuordnungsproblem Begegnung .....	34
4.3.2	Zuordnungsproblem Klasse.....	36
4.3.3	Zuordnungsproblem Rolle .....	37
4.3.4	Zuordnungsproblem Begleiter .....	39
4.3.5	Filterung.....	40
4.4	Gesamtübersicht der Datenbankverknüpfungen.....	40
<b>5</b>	<b>Analysen und Präsentation .....</b>	<b>42</b>
5.1	Allgemeines.....	42
5.2	Web-Seiten .....	43
5.2.1	Gesamtübersicht aller Begegnungen .....	44
5.2.2	Übersicht (zeitlich eingeschränkt).....	45
5.2.3	Einzelübersicht – Spieler .....	47
5.2.4	Klartextausgabe .....	52
5.3	Besondere Funktionen .....	53
<b>6</b>	<b>Zusammenfassung und Ausblick .....</b>	<b>54</b>

# 1 Einleitung und Motivation

Nach eigenen Angaben hatte das Massively Multiplayer Online Role-Playing Game (abgekürzt MMORPG) World of Warcraft® [1] von Blizzard Entertainment [2] zeitweilig mehr als 12 Millionen Abonnenten. Ein Teil davon bestreitet die aktuell schwierigsten Herausforderungen des Spiels. Dies geschieht in Teams, deren Struktur, Anspruch und Führung mit denen von Teamsportarten vergleichbar ist. Die Spieler müssen sich aufeinander verlassen können, denn ein Fehler kann schon zum Misserfolg führen. Es ist wichtig Abläufe im Spiel und Verhalten der Mitspieler zu kennen und vorausszusehen. Dabei ist der Anspruch von den, für diese Arbeit relevanten, Spielern, möglichst erfolgreich beim Bestreiten dieser Herausforderungen zu sein, und das in möglichst kurzer Zeit. Unterstützt wird das durch Webservices, wie der in dieser Arbeit. Diese ermöglichen es aufgezeichnete Daten aus dem Combat Log zu analysieren und präsentieren. Aus diesen Informationen erhält der Spieler Rückschlüsse auf sein Verhalten und das des Teams, aus denen er Gründe für ein Scheitern oder Potential für eine Optimierung der Spielweise ziehen kann. Auch andere Arten von Diensten werden von den einzelnen Spielern genutzt, um ihre Effektivität zu steigern. Diese reichen von einfachen Leitfäden bis zu Programmen, die versuchen die Mechaniken des Spiels nachzubilden.

Ein Webservice ermöglicht es vielen Nutzern die aufbereiteten Daten zu konsumieren, während nur ein Administrator sich darum kümmern muss, dass das Combat Log aufgezeichnet und zum Webservice überspielt wurde. Zusätzlich kann ein Webservice, durch den Austausch von Verknüpfungen, es ermöglichen, dass zwei Nutzer ihre Ansicht der Web-Seiten synchronisieren. Deshalb wurde für diese Arbeit ein Webservice gewählt, und mittels der Skriptsprache PHP und der Datenbank MySQL umgesetzt.

Der Webservice dieser Arbeit besteht aus einem Parser, einer Nachbearbeitung und den Web-Seiten. Der Parser extrahiert die Daten aus dem Combat Log und schreibt sie in die Datenbank. Die Nachbearbeitung verarbeitet die Daten aus der Datenbank und schreibt übergeordnete Informationen in die Datenbank. Die Präsentation der Ergebnisse erfolgt über mehrere Web-Seiten. Abbildung 1: Miniaturübersicht des Webservices gibt einen Überblick über den Webservice.



**Abbildung 1: Miniaturübersicht des Webservices**

In dieser Arbeit wird zuerst eine grobe Einführung in das Computerspielprogramm World of Warcraft® gegeben, dabei wird der Schwerpunkt auf den für diese Arbeit relevanten Teilen liegen. Danach wird die Formatierung des Combat Logs ausführlich vorgestellt, und dann auf die konkrete Umsetzung für die Datenbank und die vorhandenen Probleme und Einschränkungen eingegangen. Danach werden die einzelnen Web-Seiten des Webservices vorgestellt und ausführlich beschrieben. Dabei wird auch auf spezielle Präsentationen und Abgrenzungen zu anderen Webservices dieser Art eingegangen. Abschließend werden eine kurze Zusammenfassung der Arbeit und ein Ausblick auf weitere Möglichkeiten des Themas gegeben.

Der Webservice dieser Arbeit ist kompatibel zu den Combat Log-Dateien von World of Warcraft® in Version 4.1.0.

## **2 World of Warcraft**

In diesem Kapitel wird erst eine knappe allgemeine Einführung in das Spiel World of Warcraft® gegeben und danach die Aspekte näher erläutert, die für diese Arbeit und/oder den Motivationsbereich relevant sind.

In World of Warcraft® kann sich ein Spieler mehrere virtuelle Charaktere erstellen und steuern. Da jeder Charakter von einem Spieler gesteuert wird, wird der Einfachheit halber in dieser Arbeit vermehrt der Begriff „Spieler“ benutzt, obwohl es sich stellenweise eigentlich um Charaktere handelt. Dies betrifft in manchen Kontexten auch Charaktere, die von dem Programm gesteuert werden, und somit eigentlich nicht von Spielern gesteuerte Charaktere sind (auch NSC genannt, Kurzform für Nicht-Spieler-Charakter).

### **2.1 World of Warcraft zu Anfang**

Dieser Abschnitt ist sehr allgemein gehalten und wird dem Abstrahierten nicht gerecht, allerdings bildet es für alle Spieler einen essenziellen Teil, den jeder absolvieren muss, aber der keine direkte Relevanz für den Motivationsbereich dieser Arbeit hat. Dennoch gehört es zu einer umfassenden Einführung dazu, da in diesem Abschnitt relevante Grundlagen beschrieben werden.

Die Initialisierung eines Charakters startet mit der Rassen- und Klassenauswahl, wodurch die möglichen künftigen Rollen möglicherweise schon eingeschränkt werden (siehe hierzu Klassentabelle Tabelle 1: Rollen der einzelnen Klassen nach primärem Talentbaum und Abschnitt 2.2.11 Rollen). Zu Beginn ist der Charakter Stufe 1 und kann durch Aufträge und das Besiegen von Gegnern Erfahrungspunkte sammeln. Mit diesen Erfahrungspunkten steigt der Charakter nach und nach Stufen auf, bis zu einer maximalen Stufe von derzeit 85.

Bei manchen Stufenaufstiegen ist es dem Charakter möglich bei einem Klassentrainer neue Fähigkeiten zu lernen, mit denen neue, und teilweise effektivere, Aktionen ausgeführt werden können. Klassentrainer sind Nicht-Spieler-Charaktere und existieren für jede Klasse exklusiv. Weiterführend werden Fähigkeiten in Kapitel 2.2.8 Fähigkeiten behandelt.

Bei manchen Stufenaufstiegen bekommt der Charakter einen Talentpunkt, den er auf einen seiner drei exklusiven Talentbäume verteilen kann. Mit Stufe 10 bekommt ein Charakter zum ersten Mal einen Talentpunkt und bevor er diesen verteilen kann, muss er einen der drei Talentbäume als primären Talentbaum wählen. Danach sind die beiden anderen Talentbäume gesperrt, bis im primären Talentbaum 31 Talentpunkte verteilt sind. Auf der aktuellen maximalen Stufe von 85 kann jeder Charakter 41 Talentpunkte verteilen. Weiterführend werden Talente in Kapitel 2.2.10 Talente behandelt.

### **2.2 World of Warcraft auf maximaler Stufe**

In diesem Abschnitt werden die Aspekte beschrieben, die auf maximaler Stufe und für diese Arbeit oder den Motivationsbereich relevant sind. Dies geschieht durch die ausführliche Beschreibung von Begriffen und der relevanten Spielmechanik.

### **2.2.1 Gemeinschaft**

Bei World of Warcraft® bilden mehrere Spieler zusammen eine Gemeinschaft, die vereinsähnliche Strukturen aufweist. Gemeinschaften werden gebildet, um Spieler mit ähnlichen spielerischen oder sozialen Interessen zu vereinen, daher sind sie ein zentraler Punkt und Anlaufstelle für die meisten Spieler. Eine optimale Zahl für die Anzahl an Spielern in einer Gemeinschaft ist schwer anzugeben.

### **2.2.2 Teams**

Zwischen einem Team und einer Gemeinschaft besteht ein fließender Übergang, da auch jedes Team von der begrifflichen Bedeutung her als Gemeinschaft bezeichnet werden kann. Während mit Gemeinschaft hier eher das Umfeld gemeint ist, bezieht sich Team auf die vom Programm vorgegebenen Größen, nämlich 10 und 25, für die Anzahl an Spielern, die an für diesen Dienst relevanten Begegnungen teilnehmen können.

### **2.2.3 Begegnungen**

Als Begegnungen werden Aufeinandertreffen mit ganz speziellen Gegnern bezeichnet, die quasi als Hauptgegner eines Bereichs bezeichnet werden können.

Eine Begegnung muss nicht nur aus einem Gegner bestehen, sondern kann auch aus mehreren bestehen, manchmal sogar in der Form, dass man mehrere dieser Gegner als Hauptgegner einer einzigen Begegnung ansieht. Zusätzlich kann es vorkommen, dass der Hauptgegner letztendlich nicht final besiegt werden muss, dass er ab einem gewissen Punkt freundlich gesinnt wird oder dass das Team gar nicht direkt mit ihm interagieren kann. Desweiteren gibt es auch Begegnungen in denen die Hauptperson von Anfang an freundlich gesinnt ist und auch nie, wie sonst im üblichen Schema einer Begegnung, von dem Team angegriffen werden muss, sondern eventuell nur geheilt werden muss. Bei manchen Begegnungen spielen auch Objekte in der Spielwelt eine wichtige Rolle, oder sind gar als Hauptgegner an sich anzusehen, was aus Sicht der zur Verfügung stehenden Daten einen wichtigen Unterschied bedeutet. In einigen Begegnungen kann es vorkommen, dass ein oder mehr Spieler die Steuerung von Weltobjekten oder Fahrzeugen übernehmen, was in der Regel ein ganz anderes Schema an eingesetzten Fähigkeiten bedeutet. Die meisten Begegnungen lassen sich in mehrere Abschnitte, die Phasen genannt werden, aufteilen. Die Phasen grenzen sich dadurch ab, dass während dieser Zeit grundlegend andere Fähigkeiten von dem Hauptgegner benutzt werden oder dass sich der Ablauf der Begegnung ändert.

Dieser kleine Ausschnitt an verschiedenen Arten von Begegnungen zeigt deren mögliche Mannigfaltigkeit, denn grundsätzlich ist dem Erfindungsreichtum der Entwickler des Spiels keine Grenze gesetzt.

Neben der gemeinschaftlichen Leistung, ist die Belohnung in Form von Ausrüstung ein Hauptgrund für 10er und 25er Teams in Instanzen zu gehen.

### **2.2.4 Ausrüstung**

Unter Ausrüstung, die man unter anderem von erfolgreichen Begegnungen bekommen kann, versteht man die Gegenstände, die ein Charakter angelegt hat (im Sinne von tragen). Durch das Anlegen kann der Charakter verschiedene Arten von Boni bekommen, die es ihm ermöglichen seiner Rolle gerecht zu werden, bzw. diese besser zu erfüllen.



### 2.2.5 Instanzen

Als Instanzen bezeichnet man örtlich abgegrenzte Bereiche, in die man nur durch spezielle Zugänge gelangen kann. Instanzen werden dabei speziell für ein Team instanziiert, so dass zwei verschiedene Teams zwar namentlich in der gleichen Instanz sein können, aber nicht in derselben. In der Regel ist das nur für Teams in Größen von 10 oder 25 Spielern möglich. Der Fortschritt in einer Instanz wird für jeden Spieler getrennt behandelt und gespeichert, so dass eine Mischung von Teams zu späteren Zeitpunkten, solange der Fortschritt noch nicht gelöscht wurde, möglich ist. In der Regel wird der Fortschritt nach bis zu sieben Tagen gelöscht, da das wiederholte Bestreiten der Begegnungen, die in den Instanzen möglich sind, von den Spielern erwünscht und für die allgemeine Verbesserung der Ausrüstung nötig sind.

### 2.2.6 Klassen

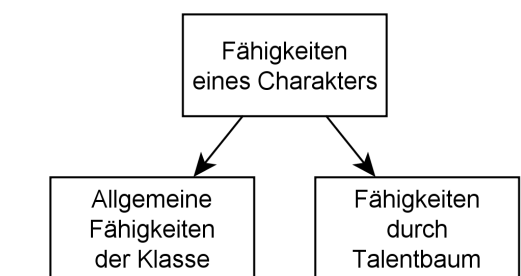
In World of Warcraft® gibt es zehn Klassen, und jede davon hat ihre eigenen einzigartigen drei Talentbäume und Fähigkeiten. Wobei es einige wenige Fähigkeiten oder Talente geben kann, die zwischen zwei Klassen sehr ähnlich oder sogar gleich sind. Zusätzlich hat jede Klasse ihre speziellen, aber nicht notwendigerweise exklusiven, Ressourcen. Die zehn Klassen sind: Druide, Hexenmeister, Jäger, Krieger, Magier, Paladin, Priester, Schamane, Schurke und Todesritter.

### 2.2.7 Ressourcen

Ressourcen sind in der Regel die Grundlage für das Ausführen von Aktionen, und somit der Interaktion mit der Spielwelt. Eine für alle Klassen gemeinsame Ressource sind die Lebenspunkte, ansonsten hängen die weiteren Ressourcen von der Klasse des Charakters ab. Diese Ressourcen können z.B. sein: Mana, Runen, Runenmacht, Fokus, Wut. Tabelle 12: Power-Type enthält eine ergänzende Beschreibung.

### 2.2.8 Fähigkeiten

Fähigkeiten gibt es in zwei verschiedenen Formen, aktive und passive. Passive Fähigkeiten gewähren Boni und aktive Fähigkeiten sind Aktionen, die den Charakter mit der Spielwelt interagieren lassen. Die Aktionen kosten in der Regel die vorher beschriebenen Ressourcen und/oder haben eine Abklingzeit, die eine sofortige Wiederbenutzung verhindert. Jede Klasse hat dabei eine allgemeine Menge an Fähigkeiten und zusätzliche Fähigkeiten durch die Wahl der Talente im Talentbaum, wie aus der



**Abbildung 2: Fähigkeiten allgemein**

Abbildung 2: Fähigkeiten allgemein zu entnehmen ist. Fähigkeiten können auch Zustandsveränderungen verursachen.

### **2.2.9 Zustandsveränderungen**

Zustandsveränderungen gibt es in zwei verschiedenen Formen, positive und negative. Positive Zustandsveränderungen sind in der Regel Spielmechaniken, die den Charakter effektiver machen. Negative Zustandsveränderungen sind in der Regel Spielmechaniken, die den Charakter weniger effektiv machen oder ihm auf andere Art schaden. Die Wirkung der meisten Zustandsveränderungen ist zeitlich begrenzt.

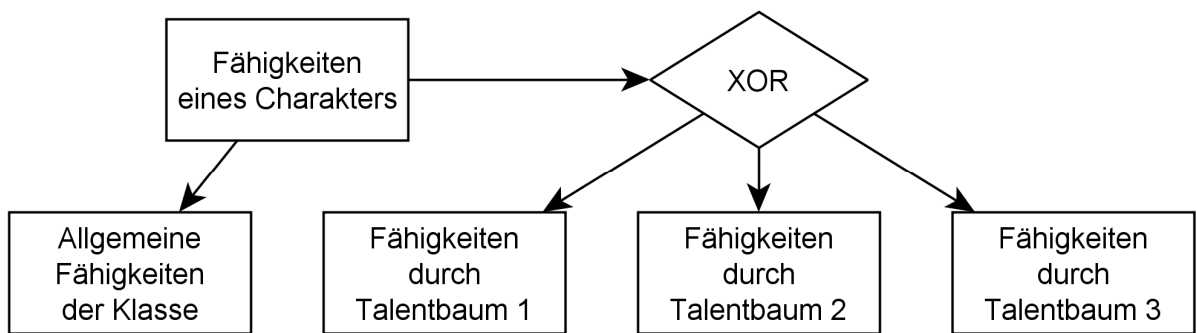
### **2.2.10 Talente**

Talente gibt es in zwei verschiedenen Formen, aktive und passive. Aktive Talente sind Fähigkeiten, die man durch den Einsatz von Talentpunkten bekommen kann. Passive Talente, die deutlich häufiger vorkommen, beeinflussen die Spielmechanik des Charakters dauerhaft. Diese Beeinflussung kann auf sehr unterschiedliche Art und Weise geschehen, von simplen Veränderungen der Effektivität, Kosten oder Abklingzeiten, über Auslösen von Zustandsveränderungen oder starke Veränderungen von Fähigkeiten, bis hin zu Auslösechancen unter bestimmten Bedingungen für verschiedene Ereignisse. Dabei kann das vorher genannte auch in beliebiger Kombination vorkommen.

Jede Klasse hat drei verschiedene exklusive Talentbäume. Jeder Talentbaum besteht aus mehreren Talenten, auf die jeweils ein bis drei Talentpunkte verteilt werden können, abhängig von der maximalen Anzahl an Talentpunkten, die auf dieses Talent verteilt werden können. Für alle Talente innerhalb eines Talentbaums, die darin sequentiell gruppiert sind, gibt es eine Beschränkung, abhängig von der Position des Talents im Talentbaum, wie viele Talentpunkte schon in diesem Talentbaum verteilt sein müssen, bevor Talentpunkte auf dieses Talent verteilt werden können. Bevor auf manche Talente Talentpunkte verteilt werden können, muss auf ein bestimmtes anderes Talent (im selben Talentbaum) schon die maximale Anzahl an Talentpunkten für dieses Talent verteilt sein. Außer den beiden zuvor beschriebenen Bedingungen, gibt es keine weiteren sequentiellen Bedingungen. Das bedeutet, dass zwei Spieler mit der gleichen aufsummierten Talentpunktverteilung, hierbei wird die Summe jedes Talentbaums einzeln angegeben (z.B. 7/31/3), nicht dieselben Talente haben müssen.

Wie bereits in Kapitel 2.1 World of Warcraft zu Anfang erwähnt, muss ein Charakter einen primären Talentbaum wählen, und in diesen 31 Talentpunkte verteilen, bis er die verbliebenen 10 Talentpunkte in die anderen beiden Talentbäume verteilen darf. Dadurch ist bedingt, dass nur auf eine sehr begrenzte Menge an Talenten in den nicht primären Talentbäumen Talentpunkte verteilt werden können. Durch die Wahl des primären Talentbaums und durch aktive Talente (also Fähigkeiten), auf die Talentpunkte in diesem Talentbaum verteilt werden, die nur mit Vergabe von mehr als 10 Talentpunkten erreicht werden können, bekommt der Charakter seine sogenannten Signaturfähigkeiten. Signaturfähigkeiten eines primären Talentbaums sind somit exklusiv zu allen anderen Fähigkeiten und identifizieren diesen Talentbaum eindeutig.

Die allgemeinen Fähigkeiten einer Klasse und die Fähigkeiten durch Talente, dies müssen nicht zwangsweise nur Signaturfähigkeiten sein, bilden die gesamten Fähigkeiten eines Charakters. Dies wird in Abbildung 3: Fähigkeiten gesamt dargestellt, wobei aus Gründen der Lesbarkeit darin nicht abgebildet ist, dass durch das Verteilen von Talentpunkten in nicht primären Talentbäumen der Charakter, in sehr seltenen Fällen, auch neue Fähigkeiten bekommen kann.



**Abbildung 3: Fähigkeiten gesamt**

Durch die Spielmechanik „duale Spezialisierung“ kann ein Charakter zwei verschiedene Talentpunktverteilungen haben, und außerhalb einer Begegnung diese auch jederzeit miteinander wechseln.

Durch die Wahl des primären Talentbaums wird auch bestimmt, welche Rollen ein Charakter wahrnehmen kann.

### **2.2.11 Rollen**

Es gibt verschiedene Rollen, die ein Spieler übernehmen kann. Als Abstraktion werden allgemein die folgenden Rollen verwendet: Schadensfänger, Heiler, Schadensausteiler(nah) und Schadensausteiler(fern). Eingeschränkt wird der Spieler hierbei einmal durch seine Klasse und zusätzlich durch den gewählten primären Talentbaum. Auf Grund der Fähigkeiten mancher Klassen oder Eigenschaften mancher Talentbäume, kann es zwar zu Überlappungen kommen, aber an sich sind die verschiedenen Rollen gut gegeneinander abgegrenzt. Unabhängig von ihrer eigentlichen Rolle kann es immer nötig sein, dass ein Spieler spezielle Aufgaben übernimmt. Solche Sonderrollen sind bei der Interpretation der Informationen, des durch diese Arbeit entstehenden Dienstes, zu berücksichtigen. Nachfolgend wird einzeln auf die verschiedenen Rollen eingegangen, die in Abbildung 4: Use-Case der Rollen als idealisiertes Use-Case Diagramm dargestellt sind. Tatsächlich ist die Nutzung und Auswirkung der Spielmechanik etwas komplizierter und vermischer, dessen Darstellung würde aber nicht den Grundgedanken der Rollen verständlich wiedergeben.

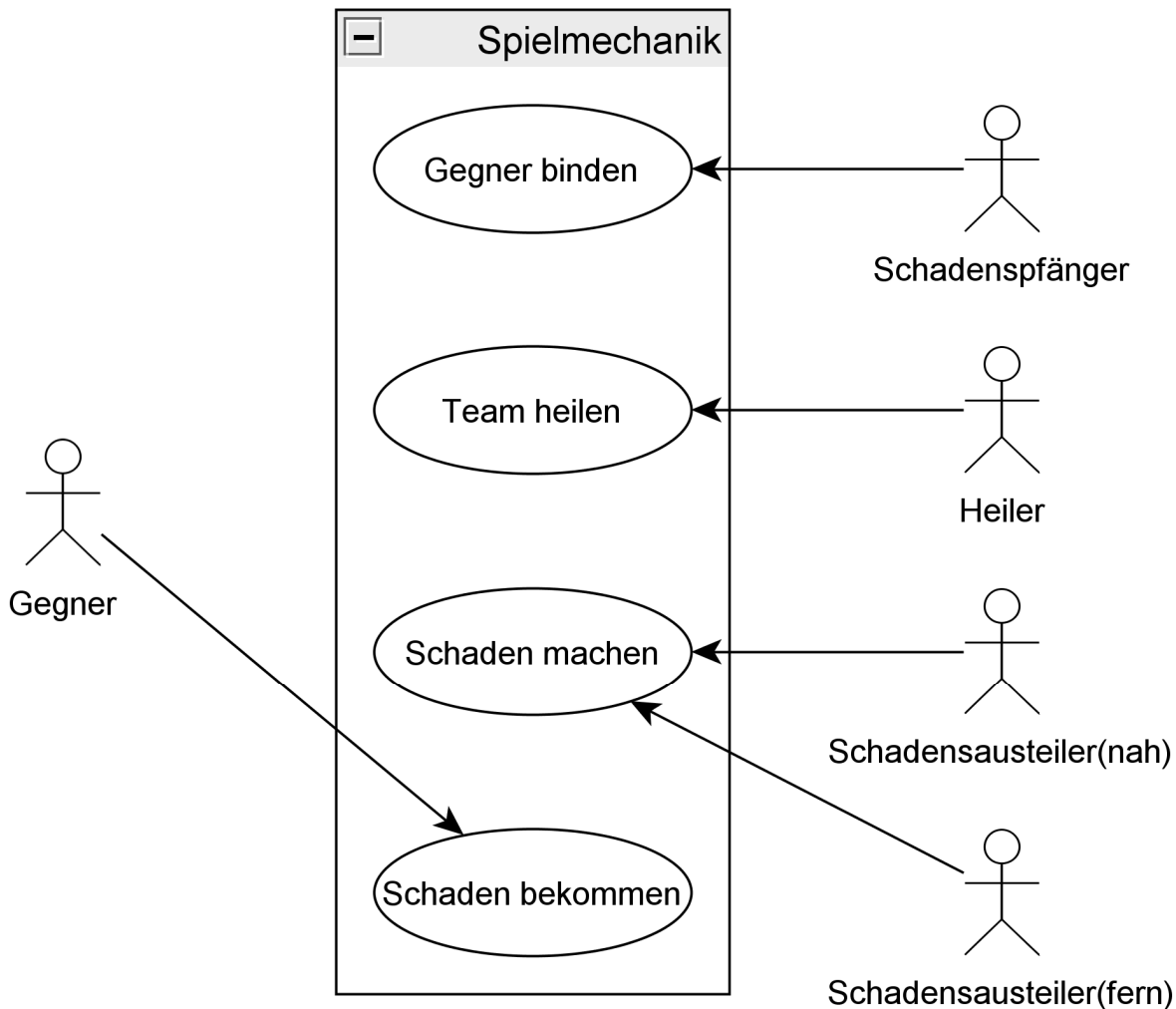


Abbildung 4: Use-Case der Rollen

### 2.2.11.1 Schadensfänger

Der Schadensfänger muss dafür sorgen, dass die Gegner hauptsächlich ihn angreifen und nicht einen anderen Spieler des Teams, da er durch seine Talentpunktverteilung und Ausrüstung dafür optimiert ist Schaden auf sich zu lenken, auszuhalten und zu vermindern. Dadurch, dass der Schadensfänger eine höhere Bedrohung bei einem Gegner verursacht, als alle anderen Spieler des Teams, bringt er den Gegner dazu, hauptsächlich ihn anzugreifen. Bedrohung ist eine Spielmechanik, die bestimmt, wen ein Gegner als Ziel seiner Angriffe nimmt. In der Regel wird durch Schaden und Heilung Bedrohung zwischen dem Spieler und Gegner generiert. Der Schadensfänger hat dabei den Vorteil, dass durch Talente und Fähigkeiten seine Generierung von Bedrohung erhöht ist. Es ist nicht immer zu vermeiden, dass auch andere Spieler aus dem Team Schaden nehmen, da manche Gegner Flächeneffekte oder Fähigkeiten, die die Bedrohungsreihenfolge ignorieren, einsetzen, und der Schadensfänger nichts dagegen unternehmen kann.

### 2.2.11.2 Heiler

Die Aufgabe des Heilers besteht darin, seine Fähigkeiten so einzusetzen, dass immer alle Spieler des Teams Heilung bekommen und kein Spieler besiegt wird, indem die Ressource

Lebenspunkte den Wert Null erreicht. In den meisten Fällen wird hauptsächlich der Schadensfänger Heilung benötigen, da er bedingt durch die Aufgaben seiner Rolle am meisten Schaden bekommt. Da Heiler ihre Heilung über eine gewisse Distanz einsetzen können, sollte dies abhängig von der Begegnung auch ausgenutzt werden.

#### **2.2.11.3 Schadensausteiler(nah)**

Die Aufgabe des Schadensausteilers(nah) besteht darin, seine Fähigkeiten so einzusetzen, dass der Gegner Schaden bekommt. Dafür muss er jedoch in der Regel in der Nähe des Gegners sein.

#### **2.2.11.4 Schadensausteiler(fern)**

Die Aufgabe des Schadensausteilers(fern) besteht darin, seine Fähigkeiten so einzusetzen, dass der Gegner Schaden bekommt. Im Gegensatz zum Schadensausteiler(nah) kann er dabei seinen Reichweitenvorteil einsetzen, und dem Gegner aus der Ferne Schaden zufügen. Ansonsten sind die beiden Rollen sich sehr ähnlich.

#### **2.2.11.5 Übersicht**

Jedem primären Talentbaum kann genau eine Rolle zugeordnet werden. Die einzige Ausnahme hiervon ist der primäre Talentbaum „Wilder Kampf“ der Klasse „Druide“, mit dem sowohl die Rollen Schadensfänger als auch Schadensausteiler(nah) wahrgenommen werden können. Eine Übersicht über die Verknüpfung von primären Talentbaum und Rolle findet sich in Tabelle 1: Rollen der einzelnen Klassen nach primärem Talentbaum. Aus der Tabelle ist auch ersichtlich, dass nicht jede Klasse jede Rolle übernehmen kann.

#### **2.2.12 Streben nach Verbesserung**

Ein wichtiger Aspekt des Spiels auf maximaler Stufe ist das Streben nach ständiger Verbesserung der Effektivität des Charakters und Teams. Dies kann auf viele Arten geschehen, von denen einige hier beschrieben werden.

Erfolgreiche Begegnungen bringen dem Team neue und meist bessere Ausrüstung, die das Team effektiver macht. Dadurch ist das Team dann in der Lage schwierigere Begegnungen zu bestreiten, um dort wiederum bessere Ausrüstung zu bekommen. Dieser stufenweise Fortschritt ist in der Regel die Langzeitmotivation der Teams.

Mit Diensten, wie der dieser Arbeit, ist es möglich die Ursachen für nicht erfolgreiche Begegnungen zu suchen, oder sogar in erfolgreichen Begegnungen noch Optimierungspotential zu finden. Für das Team als Gesamtheit ist es natürlich erstrebenswert diese Ursachen zu erkennen und zu verbessern. Manche Ursachen sind allerdings auch so trivial, dass sie direkt während der Begegnung ausgemacht werden können.

Jeder Spieler selbst sollte nach der individuellen Verbesserung seiner Spielweise streben. Dies ist durch ständiges Beschäftigen mit der Spielmechanik und der Verteilung der Talentpunkte und dem optimierten Einsetzen der Fähigkeiten, unter Berücksichtigung der limitierten Ressourcen, möglich. Mitunter ist dies auch stark abhängig von einer bestimmten Begegnung und auszufüllender Rolle. Mögliche Quellen von Leitfäden, die dieses Thema

behandeln, sind die Klassenforen von Elitist Jerks [3] und das Klassentreffen von Buffed [4]. Mitunter findet man in den Foren Tabellenkalkulationen und andere Hilfestellungen zu diesem Aspekt des Spiels.

Klasse	primärer Talentbaum	Rollen			
		Schadensfänger	Heiler	Schadens-austeiler (nah)	Schadens-austeiler (fern)
Druide	Gleichgewicht				X
	Wilder Kampf	X		X	
	Wiederherstellung		X		
Hexenmeister	Gebrechen				X
	Dämonologie				X
	Zerstörung				X
Jäger	Tierherrschaft				X
	Treffsicherheit				X
	Überleben				X
Krieger	Waffen			X	
	Furor			X	
	Schutz	X			
Magier	Arkan				X
	Feuer				X
	Frost				X
Paladin	Heilig		X		
	Schutz	X			
	Vergeltung			X	
Priester	Disziplin		X		
	Heilig		X		
	Schatten				X
Schamane	Elementar				X
	Verstärkung			X	
	Wiederherstellung		X		
Schurke	Meucheln			X	
	Kampf			X	
	Täuschung			X	
Todesritter	Blut	X			
	Frost			X	
	Unheilig			X	

**Tabelle 1: Rollen der einzelnen Klassen nach primärem Talentbaum**

## 2.3 Erklärung der Terminologie

Zum allgemeinen Verständnis werden in dieser Arbeit viele spezifische Begriffe umgangssprachlich umschrieben. Eine Übersicht dazu findet sich in Tabelle 2: Umschreibungen von spezifischen Ausdrücken.

<b>Umschreibung in dieser Arbeit</b>	<b>Spezifischer Ausdruck</b>
Schadensfänger	Tank
Schadensausteiler	DD (Damage Dealer)
Schadensausteiler(fern)	Range DD
Schadensausteiler(nah)	Melee
Positive Zustandsveränderung	Buff
Negative Zustandsveränderung	Debuff
Begegnung	Bosskampf / Boss / Encounter
Gemeinschaft	Gilde (in der Regel)
Team	Raid / Gruppe / Schlachtzug
Talentspunktverteilung	Skillung
Gegner	Mob (Mobile Unit Enemy)
Bedrohung	Aggro (Aggression)

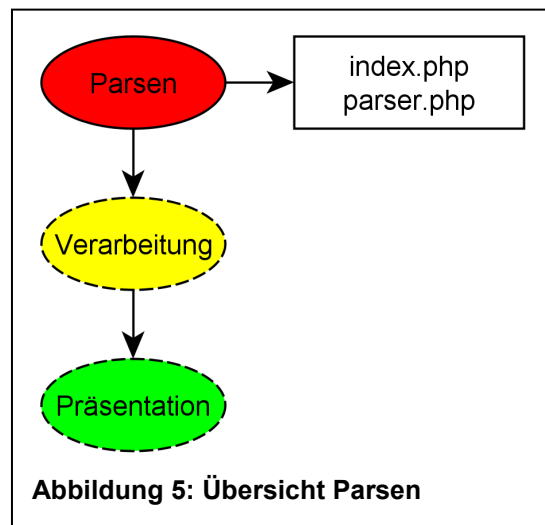
**Tabelle 2: Umschreibungen von spezifischen Ausdrücken**

Hilfestellung zur Benutzung von externen Webservices können in Anhang B gefunden werden. Screenshots der Web-Seiten des Präsentationsteils können in den Anhängen C, D; E und F betrachtet werden.

### 3 Combat Log

In diesem Kapitel werden zuerst die allgemeinen Grundlagen zum Combat Log von World of Warcraft® erklärt. Danach wird ausführlich auf die konkrete Struktur und die Bedeutung der einzelnen Variablen eingegangen. Dabei sind die hier aufgeführten Variablennamen auch die Namen der Felder, der für diesen Dienst verwendeten Datenbank, die in Kapitel 4.1 Informationen vom Parser beschrieben wird. Manche der extrahierten Variablen beinhalten mehrere kodierte Informationen, diese werden hier aufgeschlüsselt und direkt die fortan verwendeten zusätzlichen Variablennamen angegeben. Zusätzlich werden für alle statischen Werte, die in den Variablen des Combat Logs vorkommen, Indizes vergeben, die in den Tabellen der entsprechenden Variablenwerte in diesem Kapitel direkt mit angegeben werden und für die spätere Speicherung in der Datenbank benutzt werden. Abschließend werden die Beschränkungen des Combat Logs erläutert.

In diesem Kapitel wird hauptsächlich die formale Struktur des Combat Logs beschrieben, an der sich die Funktionsweise des Parsers orientiert. Details zur Implementierung sind daher nur an Stellen angegeben, an denen eine weitere Erläuterung zur Datenverarbeitung notwendig ist. Dieser Teil der Arbeit gehört zum Bereich Parsen und beinhaltet die Web-Seiten `index.php` und `parser.php`, was in Abbildung 5: Übersicht Parsen veranschaulicht ist. Wobei die Web-Seite `index.php` ein Eingabeformular darstellt, in dem die Combat Log-Datei ausgewählt werden kann, dass dann diese Informationen an die `parser.php` Web-Seite übergibt. Die Web-Seite `parser.php` liest die Rohdaten aus der übergebenen Datei, nach der in diesem Kapitel vorgestellten Struktur, aus und schreibt diese in die Datenbank.



#### 3.1 Allgemeines

In einem Combat Log von World of Warcraft® stehen alle aufgezeichneten Daten einer Sitzung. Aufgezeichnet werden dabei aber nicht alle Vorgänge im Spiel, sondern hauptsächlich der Einsatz und die Auswirkungen von Fähigkeiten. Deshalb stehen in der Aufzeichnung auch keine Daten zu Bewegung und Positionen der Charaktere, des Chats und vieler weiterer Vorgänge. Dennoch beschreibt ein Combat Log eine Begegnung ausreichend genug, um deren Verlauf aus Sicht von Einsatz und Auswirkungen von Fähigkeiten zu rekonstruieren.

Für den Combat Log in seiner Gesamtheit gibt es keine offizielle Dokumentation, für wenige Einzelheiten allerdings schon. Die Grundlage für die in dieser Arbeit erläuterte Struktur bilden die Informationen von WoWWiki [5], die sich aber hauptsächlich um die Combat Log-API im Spiel selbst beschäftigt, die in Einzelheiten nicht identisch sein muss mit den Daten der Combat Log-Datei. Ergänzt werden diese Informationen durch Quellen von Entwicklerseite, die in der Regel auch auf WoWWiki verarbeitet werden. Diese Quellen sind das AddOn Kit und sogenannte „blueposts“. Bei dem AddOn Kit [6] handelt es sich um ein Programm, das



aus kodiert gespeicherten Programmteilen dekodierte Quelltextdateien der in das Spiel integrierten Skriptsprache Lua erstellt. Diese Quelltextdateien dienen eigentlich als Beispiele für Entwickler, um Erweiterungen und Veränderungen für die Benutzeroberfläche von World of Warcraft® zu erstellen. Allerdings beinhalten sie auch den Quellcode zu einem Programmteil, der in der Benutzeroberfläche eine lesbare Version der programminternen Combat Log-Daten erzeugt. Die Struktur der programminternen Combat Log-Daten ist, den in dieser Arbeit verwendeten externen Combat Log-Daten, ähnlich. Dennoch kann dieser Programmteil, gerade bei Erweiterungen des Programms, helfen die Struktur der Daten zu analysieren. Allerdings ist das nur eingeschränkt möglich, da zum Beispiel die Struktur der GUIDs nicht erklärt ist, die später ausführlich erwähnt wird. Bei den sogenannten „blueposts“ handelt es sich um Beiträge von Angestellten des Unternehmens Blizzard in den offiziellen World of Warcraft® Foren. Da deren Beiträge immer in blauer Schrift dargestellt werden, hat sich der Begriff „bluepost“ etabliert. Vereinzelt enthalten diese Beiträge auch Informationen zu der Struktur des Combat Logs.

Die Aufzeichnung des Combat Logs, in der Textdatei „logs/combatLog.txt“ im Programmverzeichnis, wird im Spiel durch den Konsolenbefehl „/combatlog“ aktiviert. Falls diese Datei schon existiert, werden an ihrem Ende die neuen Daten angehängt, ansonsten wird sie neu erstellt. Mit demselben Befehl, oder durch Ausloggen des Charakters, wird die Aufzeichnung beendet, bzw. unterbrochen.

Auf den oben genannten Quellen basiert die Erläuterung des Combat Logs in diesem Kapitel. Allerdings wurde im Rahmen dieser Arbeit die Struktur erweitert und einzelne Informationen anders präsentiert, um im Einzelnen und insgesamt ein homogenes Gesamtbild zu erzeugen. Da es, wie oben erwähnt, keine offizielle Dokumentation gibt, sind die hier verwendeten Angaben und Strukturen teilweise als spekulativ anzusehen, wurden aber soweit durch Beobachtungen verifiziert.

## **3.2 Struktur des Combat Logs**

In jeder Zeile eines Combat Logs steht jeweils ein aufgezeichnetes Ereignis. Bestandteil eines jeden solchen Ereignisses sind mehrere Parameter, die sich nach WoWWiki [7] in Basis-, Präfix- und Suffixparameter unterteilen lassen, wobei Präfix- und Suffixparameter von dynamischer Anzahl sind. Für diese erdachte Unterteilung gibt es keine offizielle Bestätigung, allerdings vereinfacht sie das Verständnis der Struktur des Combat Logs, durch die Zusammenfassung redundanter Informationen, und bietet somit die Grundlage für eine strukturiertere Verarbeitung. Deshalb wird für diese Arbeit die Unterteilung von WoWWiki [7] übernommen und erweitert um die Ereignisse, die in WoWWiki, wahrscheinlich bedingt durch eine spätere Implementierung, nur als spezielle Ereignisse (Special Events) erwähnt sind.

Nachfolgend werden die Parameter, gegliedert nach Basis, Präfix und Suffix, ausführlich erläutert.

### **3.2.1 Basis**

Jede Zeile eines Combat Logs beginnt mit den acht Basisparametern, die in Tabelle 1: Basisparameter zu sehen sind. Alle Parameter, nicht nur die Basisparameter, sind immer durch Kommata ohne zusätzliche Leerzeichen voneinander getrennt. Einzige Ausnahme bilden der erste und zweite Parameter, also die Felder timestamp und event, diese sind nur

durch zwei Leerzeichen voneinander getrennt. Die Basisparameter drei bis fünf und sechs bis acht sind vom Typ her gleich. Sie unterscheiden sich nur darin, ob es sich um die Quelle, erkennbar an dem vorangestellten „source-“, oder das Ziel, erkennbar an dem vorangestellten „dest-“, des Ereignisses handelt.

Nachfolgend werden alle Basisparameter einzeln erklärt und wenn nötig ihre Kodierung ausführlich aufgeschlüsselt.

1.Param	2.Param	3.Param	4.Param	5.Param	6.Param	7.Param	8.Param
timestamp	event	sourceGUID	sourceName	sourceFlag	destGUID	destName	destFlag

**Tabelle 3: Basisparameter**

### 3.2.1.1 timestamp

Das Feld timestamp enthält den Zeitpunkt des Ereignisses als Zeichenkette im Format: „Monat/Tag Stunden:Minuten:Sekunden.Zehntel-/Hundertstel-/Tausendstelsekunden“ (Beispiel: 2/17 20:16:50.875). Für die Speicherung in die Datenbank wird die Zeichenkette ohne Millisekunden konvertiert in Standard Ganzzahl Repräsentation in Sekunden. Danach wird die Zahl mit 1000 Multipliziert und die Millisekunden addiert, so dass eine Ganzzahl Repräsentation in Millisekunden entsteht.

### 3.2.1.2 event

In dem Feld event ist die Ereignisart als Zeichenkette kodiert. In der Zeichenkette sind mehrere Worte jeweils durch einen Unterstrich voneinander separiert. Durch dieses Feld lassen sich die Parameter in, die in Kapitel 3.2 Struktur des Combat Logs erwähnten, Präfix- und Suffixparameter unterteilen. Die für die Einordnung relevanten Teile der Zeichenkette sind durch einen Unterstrich getrennt, allerdings kann auch mehr als ein Unterstrich vorkommen. Daher können auch innerhalb des für Präfix- oder Suffixparameter relevanten Teils Unterstriche vorkommen. In Tabelle 7: Präfixe und Parameter und Tabelle 10: Suffixe und Parameter sind alle Präfixe und Suffixe aufgelistet. Diese können zwar in verschiedenen Kombinationen miteinander vorkommen, allerdings gibt es tatsächlich nur eine sehr eingeschränkte Anzahl an Kombinationen.

### 3.2.1.3 sourceGUID und destGUID

In den Feldern sourceGUID und destGUID sind hexadezimal, mit voran gestelltem „0x“, mehrere Daten kodiert. Das grundlegende Schema orientiert sich an den Informationen zu GUIDs auf WoWWiki [8] und wurde so erweitert, dass mehrere Fälle in ein Schema passen. Die Felder können nach folgendem Schema in Subfelder unterteilt werden:  
0xAABCCCCDDDDDDDD

B (0x7)	idType
0	Player/Spieler
1	World Object/Gegenstand
3	NPC/NSC/Nicht- Spieler-Charakter
4	Pet/Begleiter
5	Vehicle/Fahrzeug

**Tabelle 4: GUID-Arten**

Die Bedeutung des Bereichs AA ist bisher noch ungeklärt und scheint keine relevanten Daten zu enthalten.

Der Bereich B, hexadezimal maskiert mit 0x7, bestimmt von welcher Art die GUID (Feld idType) ist, eine Übersicht darüber ist in Tabelle 4: GUID-Arten zu finden. Abhängig von der bestimmten Art, haben die Bereiche C und D unterschiedliche Bedeutungen.

Ist die Art 0, so stehen die Bereiche C und D zusammen genommen für die einzigartige Identifikationsnummer (Feld GUID) dieses Spielers auf einem Server. Ob wirklich alle Elemente der Bereiche C und D auch für diesen Zweck gedacht sind, ist noch unklar, da bis jetzt keine so großen Identifikationsnummern existieren. Allerdings ergibt sich mit dieser Einteilung ein homogenes Gesamtbild der Belegungsstruktur der Variable, deshalb wurde die Vorlage in diesem Sinn erweitert.

Für die Arten 1, 3, 4 und 5 steht der Bereich C für die Identifikationsnummer (Feld GUID) des Objektes und der Bereich D für die Erzeugungsnummer (Feld Spawn). Die Identifikationsnummer ist für jedes Objekt einzigartig, außer für gleichartige Objekte. Diese haben jeweils unterschiedliche Erzeugungsnummern, die einer einfachen Erhöhung unterliegen. Die Kombination von Identifikations- und Erzeugungsnummer macht somit wieder jedes konkrete Objekt einzigartig.

Die Füllung der jeweiligen Bereiche oder Bereichskombinationen ist immer rechtsbündig.

Diese Daten werden in der Datenbanktabelle *player* gespeichert.

#### **3.2.1.4 sourceName und destName**

Die Felder sourceName und destName enthalten den Namen der jeweiligen Einheit als Zeichenkette in doppelten Anführungszeichen. Diese Daten werden zusammen mit denen aus dem Bereich sourceGUID und destGUID in der Datenbanktabelle *player* gespeichert.

#### **3.2.1.5 sourceFlag und destFlag**

In den Feldern sourceFlag und destFlag sind hexadezimal, mit voran gestelltem „0x“, mehrere Informationen kodiert. Als Grundlage für dieses Schema dienten die Informationen zu UnitFlags von WoWWiki [9]. Allerdings wurde das Schema so angelegt, dass alle Informationen untereinander exklusiv sind, im Gegensatz zu den zu Grunde liegenden Informationen. Das bedeutet zum Beispiel, dass es nur eine Art der Zugehörigkeit und Reaktion gibt, diese aber beide gleichzeitig in einer Variablen vorkommen können. In dem Schema zu UnitFlags auf WoWWiki sind die speziellen Fälle (Bereich special cases) nicht exklusiv dargestellt. Dieser Bereich wurde für diese Arbeit so unterteilt, dass alle Informationen daraus exklusiv dargestellt werden. In Tabelle 6: UnitFlag werden alle enthaltenen Informationen aufgelistet und beschrieben. Der Index Bool in der Tabelle gibt dabei an, ob dieses Objekt eine Eigenschaft hat oder nicht. Im Gegensatz dazu haben Felder bei denen im Index 0 und 1 vorkommen auch für beide Werte unterschiedliche benennbare Eigenschaften, obwohl sie wie die Felder mit Bool nur durch ein Bit repräsentiert werden.

Index	Zeichen
0	Stern
1	Kreis
2	Diamant
3	Dreieck
4	Mond
5	Quadrat
6	Kreuz
7	Totenschädel

**Tabelle 5: Zeichen**

Feldname	Index	Bedeutung	Bitfeld	Beschreibung
Affiliation	MASKE		0x0000000F	Die Zugehörigkeit (Feldname Affiliation) des Objekts gibt an, in welchem Verhältnis es zum Charakter des aufzeichnenden Spielers steht. Unterschieden wird dabei zwischen Meine (Mine), Gruppe (Party), Team (Raid) und Außenseiter (Outsider).
	0	Mine	0x00000001	
	1	Party	0x00000002	
	2	Raid	0x00000004	
	3	Outsider	0x00000008	
Reaction	MASKE		0x000000F0	Die grundlegende Gesinnung, bzw. Reaktion, (Feldname Reaction) eines Objekts ist unterteilt in: freundlich (friendly), neutral (neutral), und feindselig (hostile). Charaktere die eigentlich im selben Team wie der aufzeichnende Charakter sind, aber kurzzeitig unter der Kontrolle eines feindselig gesinnten Außenseiters stehen, nehmen auch diese Kennzeichnung an und sind nur über die GUID (siehe Abschnitt sourceGUID und destGUID) als Spieler erkennbar.
	0	Friendly	0x00000010	
	1	Neutral	0x00000020	
Controller	MASKE		0x00000300	Der Kontrolleur (Feldname Controller) eines Objekts oder Charakters kann ein Spieler (Player) oder das Programm (NPC) sein.
	0	Player	0x00000100	
Type	MASKE		0x0000FC00	Der Typ (Feldname Type) eines Objektes gibt an, ob es sich um einen Spieler (Player), Nicht-Spieler-Charakter (NPC), Begleiter (Pet), Wächter (Guardian) oder Gegenstand (Object) handelt. Objekte, die unter fremder Kontrolle stehen, werden als Begleiter markiert.
	0	Player	0x00000400	
	1	NPC	0x00000800	
	2	Pet	0x00001000	
	3	Guardian	0x00002000	
Target	MASKE		0x00010000	Dieses Feld (Feldname Target) gibt an, ob es sich bei einem Objekt oder Charakter um das tatsächliche Ziel handelt. Zu jedem Zeitpunkt kann davon nur eins existieren.
	Bool	Target	0x00010000	
Focus	MASKE		0x00020000	Dieses Feld (Feldname Focus) gibt an, ob es sich bei einem Objekt oder Charakter um das Fokusziel handelt. Zu jedem Zeitpunkt kann davon nur eins existieren.
	Bool	Focus	0x00020000	
Maintank	MASKE		0x00040000	Dieses Feld (Feldname Maintank) gibt an, ob es sich bei einem Objekt oder Charakter um den markierten Hauptschadensfänger handelt.
	Bool	Maintank	0x00040000	
Mainassist	MASKE		0x00080000	Dieses Feld (Feldname Mainassist) gibt an, ob es sich bei einem Objekt oder Charakter um den markierten Hauptzielvorgabe handelt, der das als nächstes zu besiegende Ziel vorgibt.
	Bool	Mainassist	0x00080000	
Raidtarget	MASKE		0x0FF00000	Dieses Feld (Feldname Raidtarget) gibt an, ob und mit welchem von acht möglichen Zeichen das Objekt markiert ist. In Tabelle 5: Zeichen sind die Zeichen zu den entsprechenden Indizes angegeben.
	0	Raidtarget1	0x00100000	
	1	Raidtarget2	0x00200000	
	2	Raidtarget3	0x00400000	
	3	Raidtarget4	0x00800000	
	4	Raidtarget5	0x01000000	
	5	Raidtarget6	0x02000000	
	6	Raidtarget7	0x04000000	
None	MASKE		0x80000000	Dieses Feld (Feldname none) gibt an, ob überhaupt eine Quelle oder ein Ziel vorhanden ist. Bei dem Wert 1 sind alle sonstigen Quell- oder Zielvariablen 0.
	Bool	None	0x80000000	

**Tabelle 6: UnitFlag**

### 3.2.2 Präfix

Abhängig vom Präfix wird eine variable Anzahl an weiteren Parametern verarbeitet. In Tabelle 7: Präfixe und Parameter sind die verschiedenen Präfixe, die entsprechenden weiteren Parameter und Indizes ersichtlich. Diese Tabelle basiert auf den Informationen von WoWWiki [7], wurde aber erweitert, so dass die dortigen Spezialfälle auch in das Schema passen. Dies betrifft die Präfixe „DAMAGE“, „ENCHANT“, „PARTY“ und „UNIT“.

Index	Präfix	+1 Parameter	+2 Parameter	+3 Parameter
0	SWING			
1	RANGE	spellId	spellName	spellSchool
2	SPELL	spellId	spellName	spellSchool
3	SPELL_PERIODIC	spellId	spellName	spellSchool
4	SPELL_BUILDING	spellId	spellName	spellSchool
5	ENVIRONMENTAL	environmentalType		
6	DAMAGE	spellId	spellName	spellSchool
7	ENCHANT	spellName	itemId	itemName
8	PARTY			
9	UNIT			

Tabelle 7: Präfixe und Parameter

#### 3.2.2.1 spellId

Das Feld spellId enthält die einzigartige Identifikationsnummer einer Fähigkeit oder negativer oder positiver Zustandsveränderung. Das Nachschlagen einer solchen Identifikationsnummer ist mit einem Webservice, wie zum Beispiel Buffed [10], möglich. Das Feld spellId wird als Fremdschlüssel in der Datenbanktabelle *event* benutzt und ist in der Datenbanktabelle *spell* Primär-Schlüssel.

#### 3.2.2.2 spellName

Das Feld spellName enthält den Namen der jeweiligen Fähigkeit oder Zustandsveränderung als Zeichenkette in doppelten Anführungszeichen. Das Nachschlagen eines solchen Namens ist ebenfalls mit einem Webservice, wie z.B. Buffed [www-Buffed], möglich. Diese Information wird zusammen mit der aus dem Bereich spellId in der Datenbanktabelle *spell* gespeichert.

#### 3.2.2.3 spellSchool

Das Feld spellSchool gibt an, um was für eine Art von Fähigkeit es sich handelt. Dabei ist nicht deren Wirkung gemeint, sondern welcher spieltechnischen Ursprungsart sie zuzuordnen ist. Bei dem Variablenwert handelt es sich um eine Dezimalzahl, die die Informationen auf der Bit-Ebene enthält. In Tabelle 8: School-Type sind die verschiedenen Grundarten zu sehen. Besonders ist hierbei nun, dass eine Fähigkeit auch mehr als eine Grundart haben kann, und somit dann mehrere Bits den Wert 1 haben.

Der Typ Keine ist ein Spezialfall, wenn eine Fähigkeit keiner Ursprungsart zuzuordnen ist. Bei den Totemsets der Klasse Schamane ist das zum Beispiel der Fall.

Diese Information (in der dezimalen Form) wird zusammen mit der aus dem Bereich spellId in der Datenbanktabelle *spell* gespeichert.

spellSchool			Type
hex	binär	dezimal	
0x00	00000000	0	Keine
0x01	00000001	1	Physisch
0x02	00000010	2	Heilig
0x04	00000100	4	Feuer
0x08	00001000	8	Natur
0x10	00010000	16	Frost
0x20	00100000	32	Schatten
0x40	01000000	64	Arkan

**Tabelle 8: School-Type**

### 3.2.2.4 environmentalType

Das Feld environmentalType gibt mit doppelten Anführungszeichen an, um was für eine Art von Umgebungsschaden es sich handelt. In Tabelle 9: Environmental-Type sind die möglichen Werte und ihre Indizes angegeben. Nachfolgend sind die einzelnen Werte von Umgebungsschaden grob erklärt.

Die Liste könnte möglicherweise in Zukunft erweitert werden, so dass noch mehr Arten von Umgebungsschaden vorkommen können.

Index	Environmental Type	Beschreibung
0	„DROWNING“	Schaden durch Ertrinken
1	„FALLING“	Schaden durch hohen Fall
2	„FATIGUE“	Schaden durch Ermüdung, i.d.R. durch zu weites Entfernen von der begehbaren Spielwelt
3	„FIRE“	Schaden durch Feuer
4	„LAVA“	Schaden durch Lava
5	„SLIME“	Schaden durch Schleim

**Tabelle 9: Environmental-Type**

### 3.2.2.5 itemId

Das Feld itemId enthält die Identifikationsnummer eines Gegenstandes, die für alle gleichen Gegenstände identisch ist. Es ist somit nicht möglich verschiedene Inkarnationen des gleichen Gegenstandes auseinander zu halten mit den Informationen aus dem Combat Log. Das Nachschlagen einer solchen Identifikationsnummer ist mit einem Webservice, wie zum Beispiel Buffed [10], möglich.

### 3.2.2.6 itemName

Das Feld itemName enthält den Namen des jeweiligen Gegenstandes als Zeichenkette in doppelten Anführungszeichen. Das Nachschlagen eines solchen Namens ist ebenfalls mit einem Webservice, wie z.B. Buffed [10], möglich.

Index	Suffix	+1. Param	+2. Param	+3. Param	+4. Param	+5. Param	+6. Param	+7. Param	+8. Param	+9. Param
0	_DAMAGE	amount	overkill	school	resisted	blocked	absorbed	critical	glancing	crushing
1	_MISSED	missType	amountMissed							
2	_HEAL	amount	overHealing	absorbed	critical					
3	_ENERGIZE	amount	powerType							
4	_DRAIN	amount	powerType	extraAmount						
5	_LEECH	amount	powerType	extraAmount						
6	_INTERRUPT	extraSpellId	extraSpellName	extraSchool						
7	_DISPEL	extraSpellId	extraSpellName	extraSchool	auratType					
8	_DISPEL_FAILED	extraSpellId	extraSpellName	extraSchool						
9	_STOLEN	extraSpellId	extraSpellName	extraSchool	auratType					
10	_EXTRA_ATTACKS	amount								
11	_AURA_APPLIED	auratType								
12	_AURA_REMOVED	auratType								
13	_AURA_APPLIED_DOSE	auratType	amount							
14	_AURA_REMOVED_DOSE	auratType	amount							
15	_AURA_REFRESH	auratType								
16	_AURA_BROKEN	auratType								
17	_AURA_BROKEN_SPELL	extraSpellId	extraSpellName	extraSchool	auratType					
18	_CAST_START									
19	_CAST_SUCCESS									
20	_CAST_FAILED	failedType								
21	_INSTAKILL									
22	_DURABILITY_DAMAGE									
23	_DURABILITY_DAMAGE_ALL									
24	_CREATE									
25	_SUMMON									
26	_RESURRECT									
27	_SHIELD	amount	overkill	school	resisted	blocked	absorbed	critical	glancing	crushing
28	_SPLIT	amount	overkill	school	resisted	blocked	absorbed	critical	glancing	crushing
29	_SHIELD_MISSED	missType	amountMissed							
30	_APPLIED									
31	_REMOVED									
32	_KILL									
33	_DIED									
34	_DESTROYED									

Tabelle 10: Suffixe und Parameter

### 3.2.3 Suffix

Abhängig vom Suffix wird eine variable Anzahl an weiteren Parametern verarbeitet. In Tabelle 10: Suffixe und Parameter sind die verschiedenen Suffixe, die entsprechenden weiteren Parameter und Indizes ersichtlich. Diese Tabelle basiert auf den Informationen von WoWWiki [7], wurde aber erweitert, so dass die dortigen Spezialfälle auch in das Schema passen. Dies betrifft die Suffixe „\_SHIELD“, „\_SPLIT“, „\_SHIELD\_MISSED“, „\_APPLIED“, „\_REMOVED“, „\_KILL“, „\_DIED“ und „\_DESTROYED“.

Die in der Tabelle stehenden Variablennamen können je nach Kontext, also verwendetem Suffix, unterschiedliche Bedeutungen haben. So kann die Variable amount zum Beispiel einmal den Wert einer Heilung, eines Schadens oder einen anderen ereignisspezifischen Wert haben. Allerdings kann es nicht vorkommen, dass eine Variable innerhalb eines Ereignisses für unterschiedliche Werte steht.

#### 3.2.3.1 amount

Die Bedeutung des Felds amount ist von dem Kontext abhängig, in dem es benutzt wird. Bei den Suffixen „\_DAMAGE“, „\_DRAIN“, „\_LEECH“, „\_SHIELD“ und „\_SPLIT“ steht das Feld für Schaden. Bei dem Suffix „\_HEAL“ steht das Feld für Heilung. Bei dem Suffix „\_ENERGIZE“ steht das Feld für die Regeneration einer Ressource. Bei den Suffixen „\_AURA\_APPLIED\_DOSE“ und „\_AURA\_REMOVED\_DOSE“ steht das Feld für die Anzahl an Stapellungen einer Zustandsveränderung. Bei dem Feld „\_EXTRA\_ATTACKS“ steht das Feld für die Anzahl an zusätzlichen Angriffen.

#### 3.2.3.2 extraAmount

Das Feld extraAmount kommt nur im Falle von Fähigkeiten vor, die eine Ressource für die Quelle generieren und eine Ressource (i.d.R. Lebenspunkte) beim Ziel verringern, und gibt dabei den generierten Wert der Ressource an, der sich von dem Wert im Feld amount, im selben Ereignis, unterscheiden kann.

#### 3.2.3.3 overkill

Das Feld overkill gibt den Wert an, um wie viel mehr eine Fähigkeit einem Charakter oder Objekt Schaden zugefügt hat, als nötig wäre, um dieses zu besiegen.

#### 3.2.3.4 extraSpellId

Das Feld extraSpellId enthält die einzigartige Identifikationsnummer einer Fähigkeit oder negativer oder positiver Zustandsveränderung. Das Nachschlagen einer solchen Identifikationsnummer ist mit einem Webservice, wie zum Beispiel Buffed [10], möglich. Das Feld extraSpellId wird, analog zu 3.2.2.1 spellId, als Fremdschlüssel in der Datenbanktabelle *event* benutzt und ist in der Datenbanktabelle *spell* Primär-Schlüssel

#### 3.2.3.5 extraSpellName

Das Feld extraSpellName enthält den Namen der jeweiligen Fähigkeit oder Zustandsveränderung als Zeichenkette in doppelten Anführungszeichen. Das Nachschlagen



eines solchen Namens ist ebenfalls mit einem Webservice, wie z.B. Buffed [10], möglich. Diese Information wird zusammen mit der aus dem Bereich extraSpellId, und analog zu 3.2.2.2 spellName, in der Datenbanktabelle *spell* gespeichert.

### **3.2.3.6 extraSchool**

Das Feld extraSchool gibt, genau wie das Feld spellSchool, die spieltechnische Ursprungsart einer Fähigkeit an. Ausführlich wird diese Feldart in 3.2.2.3 spellSchool und Tabelle 8: School-Type beschrieben. Diese Information (in der dezimalen Form) wird zusammen mit der aus dem Bereich extraSpellId, und analog zu 3.2.2.3 spellSchool, in der Datenbanktabelle *spell* gespeichert.

### **3.2.3.7 school**

Das Feld school hingegen beschreibt, was für eine Wirkungsart eine Fähigkeit hat. Die Interpretation der Information ist allerdings analog zu Abschnitt 3.2.2.3 spellSchool und Tabelle 8: School-Type.

### **3.2.3.8 resisted, blocked und absorbed**

Die Felder resisted, blocked und absorbed enthalten den Wert, um den das Ziel den eingehenden Schaden, oder in bestimmten Fällen auch Heilung, vermindern konnte. Die Verminderung kann partiell oder total sein.

Das Feld resisted gibt an, ob einer Fähigkeit widerstanden worden ist.

Das Feld blocked gibt an, ob eine Fähigkeit geblockt worden ist.

Das Feld absorbed gibt an, ob eine Fähigkeit absorbiert worden ist.

### **3.2.3.9 critical, glancing und crushing**

Die Felder critical, glancing und crushing haben entweder den Wert 1 oder nil, somit ist deren Einfluss auf Fähigkeiten nur total oder nicht existent.

Das Feld critical gibt an, ob eine Fähigkeit kritisch eingesetzt wurde, wodurch ihre Wirkung erhöht wird. Dies kann sowohl für Heilungen, als auch für ausgeteilten oder empfangenen Schaden gelten.

Das Feld glancing gibt an, ob eine Fähigkeit nur in verminderter Menge effektiv war, im Vergleich zum Normalfall.

Das Feld crushing gibt an, ob eine Fähigkeit in gesteigerter Menge effektiv war, im Vergleich zum Normalfall.

Ein Charakter kann im Allgemeinen nicht direkt beeinflussen, ob seine Fähigkeiten critical, glancing oder crushing sind. Dies wird durch auf Wahrscheinlichkeiten basierenden Zufallszahlen gesteuert.

### **3.2.3.10 missType und amountMissed**

Das Feld missType gibt in doppelten Anführungszeichen an, weshalb eine Fähigkeit, obwohl sie von der Seite der Quelle erfolgreich zum Abschluss gebracht worden ist, nicht ihre

eigentliche Wirkung entfaltet hat. In Tabelle 11: Miss-Type sind die möglichen Werte und ihre Indizes angegeben. Nachfolgend sind die einzelnen Werte grob erklärt.

Index	Miss Type	Beschreibung
0	„ABSORB“	Fähigkeit wurde absorbiert
1	„BLOCK“	Fähigkeit wurde geblockt
2	„DEFLECT“	Fähigkeit wurde umgeleitet
3	„DODGE“	Fähigkeit wurde ausgewichen
4	„EVADE“	Fähigkeit wurde entkommen
5	„IMMUNE“	Ziel gegen Fähigkeit immun
6	„MISS“	Fähigkeit hat verfehlt
7	„PARRY“	Fähigkeit wurde pariert
8	„REFLECT“	Fähigkeit wurde reflektiert
9	„RESIST“	Fähigkeit wurde widerstanden

**Tabelle 11: Miss-Type**

Bei den Werten „ABSORB“, „BLOCK“ und „RESIST“ existiert der zusätzliche Parameter `amountMissed`, mit der entsprechenden Menge des entgangenen Schadens. Bei den anderen Werten existiert der Parameter `amountMissed` nicht. Dessen Existenz muss somit abhängig vom Wert von `missType` bestimmt werden.

Der grundsätzliche Unterschied zwischen `missType` und `failedType` ist, dass der auslösende Grund für `missType` beim Ziel liegt und bei `failedType` bei der Quelle. Hinzu kommt, dass beide gänzlich unterschiedliche Werte und Ursachen haben.

### **3.2.3.11 overhealing**

Das Feld `overhealing` gibt den Wert an, um wie viel mehr eine Fähigkeit einem Ziel Lebenspunkte hinzugefügt hat, als nötig wäre, um die Lebenspunkte dieses Ziels vollständig wiederherzustellen. Dabei gilt für dieses Ereignis, dass das Feld `overhealing` nicht von der gesamten Heilung im Feld `amount` abgezogen ist. So dass die Differenz der beiden Felder erst der effektiven Heilung entspricht.

### **3.2.3.12 powerType**

Das Feld `powerType` orientiert sich an den Arten von Ressourcen im Spiel, einleitend beschrieben in 2.2.7 Ressourcen. In Tabelle 12: Power-Type sind die verschiedenen Ressourcen, ihre Indizes, die in dieser Form direkt im Combat Log stehen, und eine Beschreibung angegeben.

Die Ereignisse, die Ressourcen betreffen und im Combat Log aufgezeichnet werden, sind nicht die einzige Möglichkeit, wie Charaktere Ressourcen regenerieren können. Die Aufzeichnung der anderen Möglichkeiten wie Ressourcen regeneriert werden, wäre aus analytischer Sicht zwar sehr interessant, allerdings würden sie den Umfang des Combat Logs extrem vergrößern.

Index	Power-Type	Beschreibung
-2	Lebenspunkte	Ressource aller Charaktere
0	Mana	Ressource von Druiden, Hexenmeistern, Magiern, Paladinen, Priestern, Schamanen und Begleitern
1	Wut	Ressource von Druiden (in Bärenform) und Kriegern
2	Fokus	Ressource von Jägern und Begleitern
3	Energie	Ressource von Druiden (in Katzenform) und Schurken
4	Begleiterzufriedenheit	Ressource von Begleitern
5	Runen	Ressource von Todesrittern
6	Runenmacht	Ressource von Todesrittern
7	Seelensplitter	Ressource von Hexenmeistern
8	Finsternis	Ressource von Druiden (primärer Talentbaum Gleichgewicht)
9	Heilige Kraft	Ressource von Paladinen
10	Alternative Ressource	Zusätzliche Ressource in bestimmten Begegnungen

**Tabelle 12: Power-Type**

### 3.2.3.13 auraType

Das Feld auraType kann nur zwei Werte haben, die als Zeichenkette in doppelten Anführungszeichen angegeben sind. Der Wert „DEBUFF“ bedeutet eine negative Zustandsveränderung und der Wert „BUFF“ bedeutet eine positive Zustandsveränderung.

Index	auraType
0	„DEBUFF“
1	„BUFF“

**Tabelle 13: Aura**

### 3.2.3.14 failedType

Das Feld failedType enthält den Grund eines Fehlschlags einer Fähigkeit als Zeichenkette in doppelten Anführungszeichen. Diese Zeichenkette enthält als Information einen meist ganzen oder zumindest halben Satz. Von diesen Sätzen sind mehr als 30 bekannt, die in englischer Sprache in Anhang A aufgelistet sind, und ständiger Erweiterung durch neue Programmversionen unterliegen können.

Der grundsätzliche Unterschied zwischen missType und failedType ist, dass der auslösende Grund für missType beim Ziel liegt und bei failedType bei der Quelle. Hinzu kommt, dass beide gänzlich unterschiedliche Werte und Ursachen haben.

## 3.3 Beispiele

Hier sind einige Beispielzeilen für Ereignisse aus einem Combat Log:

```
4/3 19:32:51.609 SWING_DAMAGE,0x03000000023F3019,"Vierxdrei",0x40512,0xF130B3A80000298E,
"Valiona",0xa48,4618,-1,1,0,0,0,nil,nil,nil
4/3 19:32:51.609 SWING_MISSED,0x03000000023F3019,"Vierxdrei",0x40512,0xF130B3A80000298E,
"Valiona",0xa48,PARRY
4/3 19:32:51.609 SPELL_PERIODIC_HEAL,0xF1300DC700003C08,"Totem des heilenden Flusses",0x2114,
0x0300000000112E6F,"Zondra",0x514,52042,"Totem des heilenden Flusses",0x8,1555,1555,0,nil
4/3 19:32:55.203 SPELL_AURA_APPLIED,0x030000000127A151,"Mooby",0x40512,0xF130B3A80000298E,
"Valiona",0x10a48,33745,"Aufschlitzen",0x1,DEBUFF
4/3 19:33:00.796 SPELL_ENERGIZE,0x030000000127A151,"Mooby",0x40512,0x030000000127A151,"Mooby",
0x40512,16959,"Urfuror",0x1,5,1
```

### 3.4 Beschränkungen

Der Combat Log von World of Warcraft® unterliegt einigen Beschränkungen. Ein Grund ist die fehlerhafte Benutzung des Combat Logs durch zu frühes oder zu spätes Einschalten oder Abschalten der Aufzeichnung. Auch ein in der Spielwelt zu großer räumlicher Abstand führt dazu, dass nicht mehr alle Daten des Teams aufgezeichnet werden können. Dadurch kann nicht garantiert werden, dass eine Aufzeichnung vollständig ist. Ein anderer Ursprung sind die ständig wiederkehrenden Neuinitialisierungen des Servers, hervorgerufen durch Neustarts, bei denen das Feld Spawn (so fern vorhanden) von Einheiten zurücksetzt wird. Wie in Abschnitt 3.1 Allgemeines schon erwähnt, werden keinerlei Bewegungsinformationen aufgezeichnet. Weitere Ursprünge liegen in der Struktur des Combat Logs und werden nachstehend angesprochen.

Ereignisse in denen die Variable failedType vorkommt, also bei dem Suffix „\_CAST\_FAILED“ (Index 20), werden nur von dem aufzeichnenden Charakter aufgezeichnet. Bei den Ereignissen mit dem Präfix „ENCHANT“ (Index 7), bei denen ein Gegenstand mittels einer Fähigkeit modifiziert wird, fehlt die Angabe der Identifikationsnummer der auslösenden Fähigkeit. Das Ereignis mit dieser auslösenden Fähigkeit, welches das Ereignis „SPELL\_CAST\_SUCCESS“ ist, scheint immer das vorhergehende Ereignis des Charakters zu sein. Allerdings kann das auf Grund von Verzögerungen bei der Signalübertragung hier eventuell auch zu Inkonsistenzen führen. Zusätzlich lässt sich die auslösende Fähigkeit nicht klar dem folgenden Ereignis zuordnen, da es keinerlei gemeinsame Informationen der beiden Ereignisse geben muss. Das Feld spellName, dass in beiden Ereignissen vorkommt, beinhaltet bei dem Präfix „ENCHANT“ lediglich die namentliche Modifikation des Gegenstands und bei dem Ereignis „SPELL\_CAST\_SUCCE“ den Namen der auslösenden Fähigkeit, die nicht miteinander identisch sein müssen. Es ist allerdings wahrscheinlich, dass die Felder namentlich ähnlich sind, so dass die namentliche Modifikation des Gegenstands in dem Namen der auslösenden Fähigkeit vorkommen sollte.

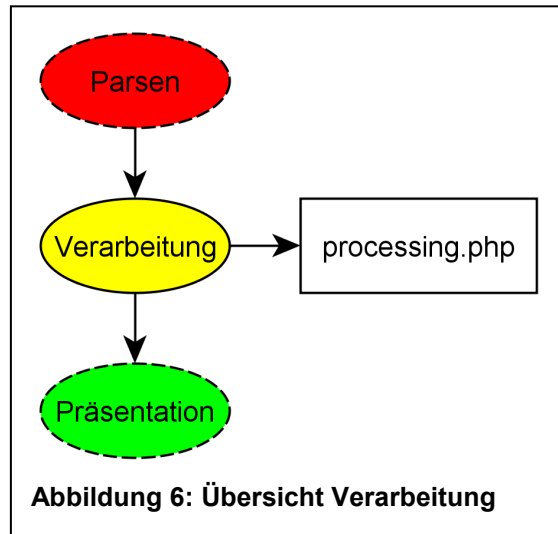
Es existieren keine exklusiven Ereignisse, die den Anfang oder das Ende einer Begegnung kennzeichnen. In der Regel ist es problematischer zu bestimmen, wann eine Begegnung vorbei ist. Dies betrifft hauptsächlich nicht erfolgreich versuchte Begegnungen, in denen das Team besiegt wurde. Mitunter wird dabei aber nicht jeder einzelne Charakter besiegt, da es Spielmechaniken gibt, die das umgehen können, oder die Aufzeichnung beinhaltet nicht alle Informationen. Sicher erkennbar ist so ein Fehlversuch, wenn der gleiche Hauptgegner mit einer neuen Erzeugungsnummer (Feld spawn) auftaucht. Erfolgreich versuchte Begegnungen enden gewöhnlich mit dem Besiegen eines Hauptgegners und dem Ereignis „UNIT\_DIED“ mit dem Hauptgegner als Ziel. Es gibt aber auch einige erfolgreich versuchte Begegnungen, bei denen das Ende nicht auf gewöhnliche Weise zu erkennen ist. Dies können Begegnungen sein, in denen die Haupteinheit von Anfang an freundlich gesinnt ist. Diese werden am Ende der Begegnung dann eventuell eine spezielle Fähigkeit einsetzen, anhand der das Ende bestimmt werden kann. Eine weitere Möglichkeit sind Begegnungen bei denen die Haupteinheit am Ende der Begegnung die Gesinnung (Flag Reaction, bzw. Feldnamen sourceReaction und destReaction) von feindlich auf freundlich oder neutral wechselt, und somit auch im Combat Log nachvollziehbar sein sollte. Eine weitere Möglichkeit sind Begegnungen, die mehrere Hauptgegner haben, hierbei muss dann darauf geachtet werden, ob alle Hauptgegner besiegt worden sind. Eine weitere Möglichkeit sind Begegnungen in denen der Hauptgegner nicht direkt besiegt wird, aber auch nicht die Gesinnung ändert. Bei solchen Begegnungen kann das Einsetzen einer speziellen Fähigkeit

oder das Eingreifen weiterer Einheiten das Ende der Begegnung anzeigen. Eine weitere Problematik können die möglichen Variationen an Begegnungsarten sein, die schon in 2.2.3 Begegnungen beschrieben worden sind.

Außerdem können die beschriebenen Strukturen jederzeit einer Änderung durch die Entwickler unterliegen, womit die vorgestellten Parameter oder das Trennungsprinzip von dem Feld event in Präfix und Suffix hinfällig werden.

## 4 Nachbearbeitung und Datenstruktur/Datenbanken

In diesem Kapitel werden die Datenstruktur der Datenbanken und die nötigen Nachbearbeitungsschritte erklärt. Beide Themen sind voneinander abhängig und werden daher teilweise parallel erklärt. Angefangen wird mit einem Überblick der Informationen, die vom Parser generiert werden. Danach werden die statischen Informationen beschrieben und wofür sie benötigt werden. Abschließend werden die nötigen Nachbearbeitungsschritte (Web-Seite `processing.php`) und die daraus gebildeten Informationen beschrieben. Die Einordnung in den Gesamtzusammenhang wird in Abbildung 6: Übersicht Verarbeitung veranschaulicht.



### 4.1 Informationen vom Parser

Die vom Parser (Web-Seite `parser.php`) generierten Daten aus der Combat Log-Datei werden in den Datenbanktabellen *event*, *char* und *spell* gespeichert, und stellen die elementaren Informationen dar. Dabei entspricht ein Eintrag in der Datenbanktabelle *event* einer Zeile in der Combat Log-Datei, wobei jedoch redundante Informationen bezüglich der Charaktere und der Fähigkeiten in die Datenbanktabellen *char* und *spell* ausgelagert werden.

Abbildung 7: Verknüpfungen von den Datenbanktabellen *event*, *player* und *spell* zeigt die Verknüpfung der Datenbanktabellen *event*, *char* und *spell*.

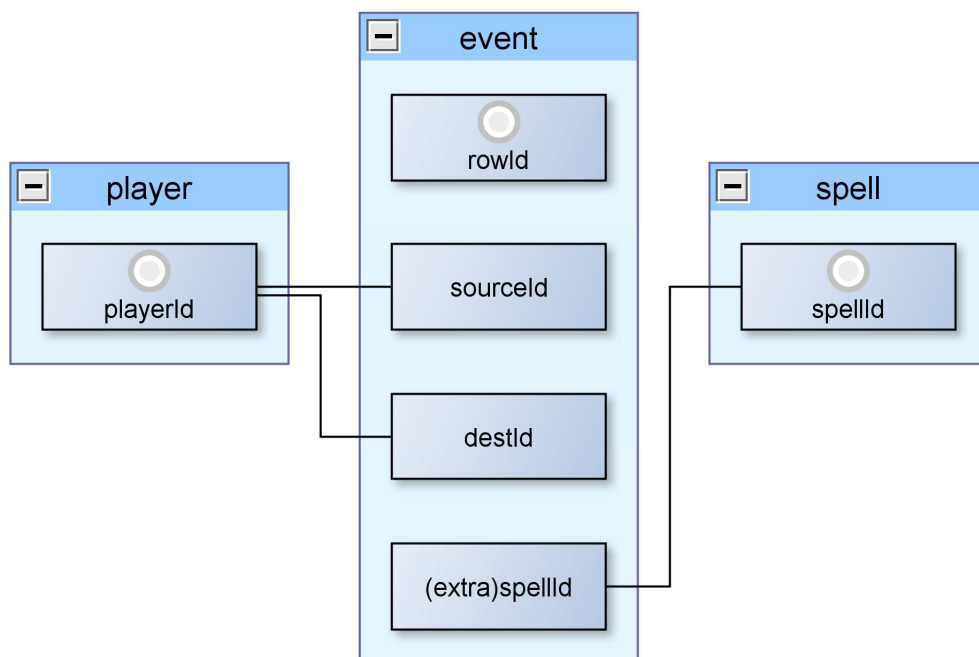


Abbildung 7: Verknüpfungen von den Datenbanktabellen *event*, *player* und *spell*

<b>Feldname</b>	<b>Typ</b>	<b>Beschreibung</b>
rowId	int(10) unsigned	Zeile in der Combat Log-Datei
millimestamp	bigint(13) unsigned	Zeitpunkt des Ereignisses
prefix	int(1) unsigned	Präfix des Ereignisses
suffix	int(2) unsigned	Suffix des Ereignisses
sourceId	int(10) unsigned	Identifikationsnummer (eigenes Format) der Quelle
sourceType	int(1) unsigned	Typisierung der Quelle
sourceController	bit(1)	Kontrollleur der Quelle
sourceReaction	int(1) unsigned	Gesinnung der Quelle
sourceAffiliation	int(1) unsigned	Zugehörigkeit der Quelle
sourceNone	bit(1)	Wert 1 = keine Quelle
sourceRaidtarget	int(1) unsigned	Eventuelles Zeichen der Quelle
sourceMainassist	bit(1)	Quelle ist Hauptzielvorgeber
sourceMaintank	bit(1)	Quelle ist Hauptschadensfänger
sourceFocus	bit(1)	Quelle ist das Fokusziel
sourceTarget	bit(1)	Quelle ist das tatsächliche Ziel
destId	int(10) unsigned	Identifikationsnummer (eigenes Format) des Ziels
destType	int(1) unsigned	Typisierung des Ziels
destController	bit(1)	Kontrollleur des Ziels
destReaction	int(1) unsigned	Gesinnung des Ziels
destAffiliation	int(1) unsigned	Zugehörigkeit des Ziels
destNone	bit(1)	Werte 1 = kein Ziel
destRaidtarget	int(1) unsigned	Eventuelles Zeichen des Ziels
destMainassist	bit(1)	Ziel ist Hauptzielvorgeber
destMaintank	bit(1)	Ziel ist Hauptschadensfänger
destFocus	bit(1)	Ziel ist das Fokusziel
destTarget	bit(1)	Ziel ist das tatsächliche Ziel
spellId	int(10) unsigned	Identifikationsnummer der Fähigkeit (selbe SpellId, wie im Programm)
environmentalType	int(1) unsigned	Art des Umgebungsschadens
amount	int(10) unsigned	Wertigkeit eines Ereignisses
overkill	int(10) unsigned	Überzähligen Schaden an besiegten Gegnern
school	int(3) unsigned	Wirkungsart der Fähigkeit
resisted	int(10) unsigned	Widerstandener Wert
blocked	int(10) unsigned	Geblockter Wert
absorbed	int(10) unsigned	Absorbierter Wert
critical	bit(1)	Kritisches Ereignis
glancing	bit(1)	Streifendes Ereignis
crushing	bit(1)	Krachendes Ereignis
missType	int(1) unsigned	Verfehlungsart
amountMissed	int(10) unsigned	Wertigkeit der Verfehlung
overHealing	int(10) unsigned	Überzählige Heilung
powerType	int(1)	Ressourcenart
extraAmount	int(10) unsigned	Extra Wertigkeit eines Ereignisses
extraSpellId	int(10) unsigned	Extra Identifikationsnummer einer Fähigkeit (selbe SpellId, wie im Programm)
auraType	bit(1)	Art der Zustandsveränderung

**Tabelle 14: Datenbanktabelle event**

#### 4.1.1 Datenbanktabelle event

In der Datenbanktabelle *event* (siehe Tabelle 14: Datenbanktabelle event) werden die Informationen zu den einzelnen Ereignissen (Zeilen in der Combat Log-Datei) gespeichert, mit Ausnahme der ausgelagerten Informationen in den Datenbanktabellen *player* und *spell*. Dies betrifft die Felder *sourceId*, *destId*, *extraSpellId* und *spellId*, die Fremdschlüssel für die jeweilige Datenbanktabelle darstellen.

#### 4.1.2 Datenbanktabelle player

In der Datenbanktabelle *player* (siehe Tabelle 15: Datenbanktabelle player) werden die Informationen zu den einzelnen Charakteren gespeichert, unabhängig davon, ob es sich dabei um von Spielern gesteuerte Charaktere handelt oder nicht. Dabei ist die verwendete Identifikationsnummer (Feld *playerId*) ein eigenes Format und hat nichts mit der des Programms (Feld *GUID*, bzw. die Basisparameter *source-* und *destGUID*) zu tun. Die Identifikationsnummer aus dem Feld *GUID* kann allerdings bei Nicht-Spieler-Charakteren als Verweis zu anderen Diensten benutzt werden

Feldname	Typ	Beschreibung
<i>playerId</i>	<i>int(10) unsigned</i>	Identifikationsnummer des Charakters oder Objekts (eigenes Format)
<i>idType</i>	<i>int(1) unsigned</i>	Art des Eintrags (Spieler, Welt Objekt, Nicht-Spieler-Charakter, Begleiter, Fahrzeug)
<i>GUID</i>	<i>int(10) unsigned</i>	Identifikationsnummer des Charakters (ungleich zu <i>playerId</i> ). Bei Spielern ist diese, bis auf geringe Einschränkungen, immer gleich. Bei anderen Arten beschreibt es nur das Objekt, aber nicht eine genaue Instanz davon.
<i>spawn</i>	<i>int(10) unsigned</i>	Erzeugungsnummer des Objektes
<i>name</i>	<i>varchar(30)</i>	Name des Charakters oder Objekts

Tabelle 15: Datenbanktabelle player

#### 4.1.3 Datenbanktabelle spell

In der Datenbanktabelle *spell* (siehe Tabelle 16: Datenbanktabelle spell) werden die Informationen zu den einzelnen Fähigkeiten gespeichert. Dabei sind die Identifikationsnummer der Tabelle (Feldname *spellId*) und die des Programms dieselben. Dadurch sind auch Verweise auf andere Dienste möglich, die mit diesen Identifikationsnummern arbeiten.

Feldname	Typ	Beschreibung
<i>spellId</i>	<i>int(10) unsigned</i>	Identifikationsnummer der Fähigkeit (selbe <i>SpellId</i> , wie im Programm)
<i>spellName</i>	<i>varchar(30)</i>	Name der Fähigkeit
<i>spellSchool</i>	<i>int(3) unsigned</i>	Schule der Fähigkeit

Tabelle 16: Datenbanktabelle spell

## 4.2 Statische Informationen

Für die Umsetzung dieses Dienstes werden auch statische Informationen benötigt. Diese Informationen werden benutzt, um statische Sachverhalte, wie Begegnungen und Klassen-/Rollenbeziehungen in den Daten des Combat Logs zu identifizieren. Zur einfachen



Visualisierung und Erweiterbarkeit stehen diese Daten ebenfalls in Datenbanktabellen, und werden in diesem Abschnitt beschrieben.

#### 4.2.1 Datenbanktabelle *static\_encounter*

Damit einzelne Ereignisse Hauptbegegnungen zugeordnet werden können, werden Informationen zu den Identifikationsmerkmalen der Hauptbegegnungen benötigt. Diese Informationen sind in der Datenbanktabelle *static\_encounter* eingetragen (siehe Tabelle 17: Datenbanktabelle *static\_encounter*).

In der Regel handelt es sich um die Identifikationsnummer (Feld GUID) des Hauptgegners einer Begegnung.

Feldname	Typ	Beschreibung
staticEncounterId	int(10) unsigned	eigenes Format
GUID	int(10) unsigned	Gleiche Identifikationsnummer wie in Datenbanktabelle player (Verweis auf externe Dienste möglich)
name	varchar(30)	Name der Begegnung

**Tabelle 17: Datenbanktabelle *static\_encounter***

Nachfolgend sind einige Beispiele für Einträge in der Datenbanktabelle *static\_encounter* zu finden (siehe Tabelle 18: Beispiele für *static\_encounter*):

staticEncounterId	GUID	name
17	44600	Halfus Wyrmbrecher
18	45993	Theralion
19	43324	Cho`gall

**Tabelle 18: Beispiele für *static\_encounter***

#### 4.2.2 Datenbanktabelle *static\_spell\_classtree*

Damit Spielern bestimmte Rollen (siehe 2.2.11 Rollen) zugeordnet werden können, werden Informationen benötigt, die eine Beziehung zwischen eingesetzten Fähigkeiten und primärem Talentbaum herstellt. Genutzt werden dazu die sogenannten Signaturfähigkeiten, die einzigartig sind und auf einen bestimmten primären Talentbaum (und somit auch auf eine Klasse) hinweisen. Anhand des primären Talentbaums kann dann die Rolle des Spielers bestimmt werden (siehe Tabelle 1: Rollen der einzelnen Klassen nach primärem Talentbaum). Diese Informationen sind in der Datenbanktabelle *static\_spell\_classtree* eingetragen (siehe Tabelle 19: Datenbanktabelle *static\_spell\_classtree*).

In der Datenbanktabelle *static\_spell\_classtree* sind solche Beziehungen zwischen Identifikationsnummer einer Fähigkeit (Feldname spellId) und Talentpunktverteilung einer Klasse (Feldname classtree), und somit auch Klasse (Feldname class) und Rolle (Feldname role), eingetragen.

Feldname	Typ	Beschreibung
spellId	int(10) unsigned	Identifikationsnummer der Fähigkeit (selbe SpellId, wie im Programm)
name	varchar(30)	Name der Signaturfähigkeit
class	Aufzählung(10 Werte)	Zugeordnete Klasse
classtree	Aufzählung(30 Werte)	Zugeordneter primärer Talentbaum
role	Aufzählung(4 Werte)	Zugeordnete Rolle

**Tabelle 19: Datenbanktabelle static\_spell\_classtree**

Nachfolgend sind einige Beispiele für Einträge in der Datenbanktabelle *static\_spell\_classtree* zu finden (siehe Tabelle 20: Beispiele für static\_spell\_classtree).

spellId	name	class	classtree	Role
85288	Wütender Schlag	Krieger	Krieger - Furor	Schadensausteiler(nah)
98008	Schnelligkeit der Natur	Schamane	Schamane - Wiederherstellung	Heiler
49143	Froststoß	Todesritter	Todesritter - Frost	Schadensausteiler(nah)

**Tabelle 20: Beispiele für static\_spell\_classtree**

## 4.3 Informationen durch Nachbearbeitung

Während durch das Parsen (Web-Seite parser.php) die elementaren Informationen aus der Combat Log-Datei extrahiert wurden, werden durch die Nachbearbeitung (Web-Seite processing.php) die darin enthaltenen übergeordneten Informationen gebildet. Dazu zählen unter anderen die Begegnungen, Rollen, Klassen und Begleiterzugehörigkeit.

Zuvor werden einige Ereignisse aus dem Datenbestand entfernt, da diese Art nur von dem aufzeichnenden Spieler aufgezeichnet werden, oder auf eine andere Art nicht inhaltlich sinnvoll sind, und somit nicht wertvoll für einen Dienst sind, der sich mit allen Einheiten des Teams/Begegnung beschäftigt.

### 4.3.1 Zuordnungsproblem Begegnung

Da es, wie in 3.4 Beschränkungen erwähnt, keine gesonderten Ereignisse gibt, die den Beginn und das Ende einer Begegnung (und somit auch Hauptbegegnung) markieren, wird nach dem ersten und letzten Vorkommen einer bestimmten Identifikationsnummer gesucht, um die Grenzen einer Hauptbegegnung zu bestimmen.

Alle für diesen Dienst relevanten Identifikationsnummern von Hauptbegegnungen stehen in der Datenbanktabelle *static\_encounter* in dem Feld GUID. Deren Vorkommen in der Datenbanktabellen *player* (Feld GUID) zu den im Combat Log stehenden Hauptgegnern führt. Bei erfolglosen Versuchen einer Hauptbegegnung, entsteht der Hauptgegner in einer neuen Inkarnation, die in der Datenbanktabelle *player* mit dem Feld spawn unterschieden werden kann, wobei zusätzlich für alle Einträge mit dem Feld playerId eine exklusive Identifikationsnummer gebildet wird, und fortan als eindeutige Identifizierung benutzt wird. Getrennt für alle Inkarnationen von Hauptgegnern, werden in der Datenbanktabelle *event* das erste und letzte Ereignis gesucht, in dem das Feld destId oder sourceId gleich der

gesuchten Identifikationsnummer ist. Von diesen Ereignissen wird die Identifikationsnummer (Feld rowId) und der Zeitstempel (Feld militimestamp) für den späteren Gebrauch benötigt.

In der Datenbanktabelle *encounter\_rel* sind dann pro Hauptbegegnung Informationen zur Dauer, Versuchsnummer, Erfolgsstatus, Fremdschlüssel der Datenbanktabellen *event* und *static\_encounter* und die eigene Identifikationsnummer gespeichert (siehe Tabelle 21: Datenbanktabelle *encounter\_rel*). Die Information der Dauer ist Redundant zu den Fremdschlüsseln für die Datenbanktabelle *event* (Felder *minrow* und *maxrow*), über die diese Information auch erhalten werden kann, allerdings wird durch die Redundanz bei der Präsentation eine Datenbanktabellenverknüpfung auf die größte Datenbanktabelle *event* gespart und die Abfrage schneller bearbeitet.

Bei Begegnungen, bei denen es darum geht einen Hauptgegner zu besiegen, ist es in der Regel möglich, mit einem speziellen Ereignis den erfolgreichen Abschluss der Begegnung zu bestimmen. Wie oben beschrieben, lässt sich jeder gefundenen Hauptbegegnung eine eindeutige Identifikationsnummer (Feld *playerId* in der Datenbanktabelle *player*) zuordnen. Für diese Identifikationsnummer wird überprüft, ob es das Ereignis „UNIT\_DIED“ (Prefix-Index 9 und Suffix-Index 33) gibt. Ist so ein Ereignis vorhanden, wird in das entsprechende Feld *succes* eine eins für „wahr“ eingetragen, ansonsten ist der Standardeintrag des Feldes null für „falsch“.

Feldname	Typ	Beschreibung
<i>encounterId</i>	int(10) unsigned	Identifikationsnummer (eigenes Format)
<i>minrow</i>	int(10) unsigned	erste zugeordnete Zeile in der Datenbanktabelle <i>event</i>
<i>maxrow</i>	int(10) unsigned	letzte zugeordnete Zeile in der Datenbanktabelle <i>event</i>
<i>mintime</i>	bigint(13) unsigned	Beginn der Begegnung
<i>maxtime</i>	bigint(13) unsigned	Ende der Begegnung
<i>staticEncounterId</i>	int(10) unsigned	Fremdschlüssel von der Datenbanktabelle <i>static_encounter</i>
<i>try</i>	int(10) unsigned	Versuchsnummer
<i>succes</i>	tinyint(1)	Erfolgssatus

**Tabelle 21: Datenbanktabelle *encounter\_rel***

Nachfolgend steht die Datenbankabfrage zur Ermittlung der Begegnungen:

```
INSERT INTO encounter_rel (minrow, maxrow, mintime, maxtime, staticEncounterId)
SELECT min(rowId), max(rowId), min(militimestamp), max(militimestamp),
static_encounter.staticEncounterId
FROM static_encounter
JOIN player ON (player.GUID = static_encounter.GUID)
JOIN event ON (event.sourceId = player.playerId OR event.destId = player.playerId)
GROUP BY playerId
```

Nachfolgend steht die Datenbankabfrage zur Ermittlung des Erfolgsstatus der Begegnung:

```
UPDATE encounter_rel, event, player, static_encounter
SET encounter_rel.succes = true
WHERE (event.rowId BETWEEN encounter_rel.minrow AND encounter_rel.maxrow) AND
(player.playerId = event.destId) AND (player.GUID = static_encounter.GUID) AND (event.suffix = 33)
```

Eine Übersicht der Verknüpfungen der Felder der Datenbanktabelle *encounter\_rel* zu anderen Datenbanktabellen gibt Abbildung 8: Verknüpfungen der Datenbanktabelle *encounter\_rel*.

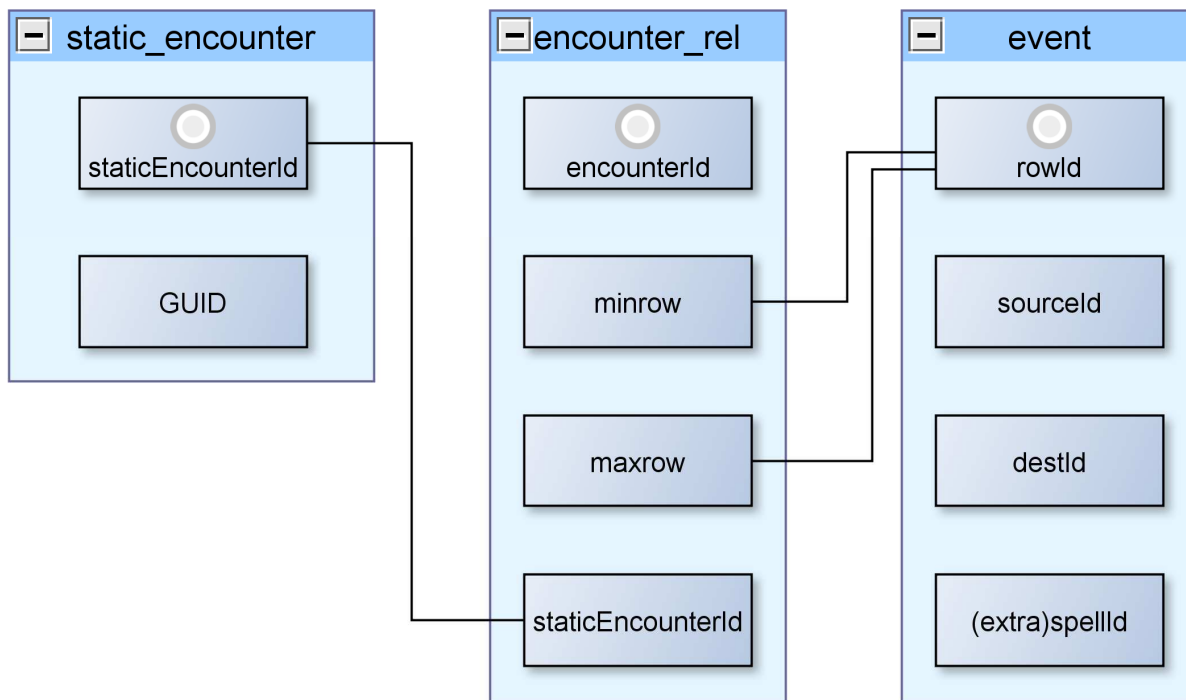


Abbildung 8: Verknüpfungen der Datenbanktabelle *encounter\_rel*

#### 4.3.2 Zuordnungsproblem Klasse

Die Klasse (siehe 2.2.6 Klassen) eines Spielers kann sich niemals ändern, daher kann sie unabhängig von den Begegnungen gespeichert werden. Obwohl jede Klasse drei verschiedene primäre Talentbäume (siehe 2.2.11.5 Übersicht) hat, eignen sich die Signaturfähigkeiten (siehe 2.2.10 Talente) aus allen drei Talentbäumen zur Klassifikation einer Klasse. Weitere Fähigkeiten, die nur exklusiv von einer Klasse benutzt werden können, aber keine Signaturfähigkeiten sind, würden sich auch zur Klassifikation eignen, sind aber bei der Menge der Ereignisse, die ein Combat Log normalerweise hat, nicht nötig.

Für die Zuordnung der Klasse wird geprüft, welche Identifikationsnummern (Feld *spellId*) von Signaturfähigkeiten aus der Datenbanktabelle *static\_spell\_classtree* in dem Feld *spellId* in der Datenbanktabelle *event* vorkommen. Dem entsprechenden Spieler (Feld *sourceId*) wird die Klasse (Feld *class*) aus der Datenbanktabelle *static\_spell\_classtree* zugeordnet. Die Struktur von Datenbanktabelle *class\_rel* ist in Tabelle 22: Datenbanktabelle *class\_rel* dargestellt.

Feldname	Typ	Beschreibung
<i>playerId</i>	<i>int(10) unsigned</i>	Identifikationsnummer des Spielers
<i>class</i>	<i>Aufzählung(10)</i>	Zugeordnete Klasse des Spielers

Tabelle 22: Datenbanktabelle *class\_rel*

Die Umsetzung der Zuordnung der Klasse wird durch folgende Datenbankabfrage ermöglicht:

```
INSERT INTO class_rel (playerId, class)
SELECT event.sourceId, static_spell_classtree.class
FROM event
STRAIGHT_JOIN static_spell_classtree ON (static_spell_classtree.spellId = event.spellId)
GROUP BY event.sourceId
```

### 4.3.3 Zuordnungsproblem Rolle

Im Gegensatz zur Klasse eines Spielers, kann sich die Rolle (siehe 2.2.11 Rollen) zwischen Begegnungen ändern, auch bei wiederholten Versuchen der gleichen Begegnung. Daher kann eine Zuordnung von Spieler zu Rolle nur pro Begegnung gemacht werden. Bestimmt wird eine Rolle in der Regel durch die Zuordnung zu einem primären Talentbaum (siehe 2.2.10 Talente) durch den Einsatz von Signaturfähigkeiten.

Zunächst werden in der Datenbanktabelle *role\_rel* (siehe Tabelle 23: Datenbanktabelle *role\_rel*) alle beteiligten menschlichen Spieler (Feld *idType* = 0 in der Datenbanktabelle *player*), sowie deren Aktivitätszeitraum (Felder *mintime* und *maxtime*) eingetragen. Dies geschieht pro Begegnung aus der Datenbanktabelle *encounter\_rel* (Feld *encounterId*) und dem Vorkommen eines Ereignisses mit dem Spieler in der Datenbanktabelle *event* in den entsprechenden Zeilen der Begegnung (Felder *rowId*, sowie *minrow* und *maxrow*). Dabei werden noch keine Rollen zugeordnet, die Tabelle dient als Grundlage für die weiteren Schritte.

Zuerst wird nun mit den schon ermittelten Informationen und den bekannten Abhängigkeiten zwischen den Datenbanktabellen überprüft, welchem Spieler durch Einsatz von Signaturfähigkeiten ein primärer Talentbaum, und somit eindeutig eine Rolle, zugeordnet werden kann.

Bei sehr kurzen, und damit in der Regel erfolglosen, Begegnungen kann es vorkommen, dass keine Klassifizierung des primären Talentbaums erfolgen kann. Zusätzlich sind die Signaturfähigkeiten des primären Talentbaums „Elementar“ der Klasse „Schamane“ nicht sonderlich gut für eine Klassifikation geeignet. Im Gegensatz zu anderen Signaturfähigkeiten müssen sie nicht unbedingt ständig eingesetzt werden und sind eher von der Gesamtsituation abhängig. Daher wird für diesen primären Talentbaum eine Zuordnung anhand von anderen Kriterien benötigt. Hierfür kommen vor allem Zustandsveränderungen und Fähigkeiten, die nicht direkt vom Spieler ausgelöst werden, aus Sicht des Ursache-Wirkungs-Prinzips, aber dennoch genauso kodiert sind im Combat Log, in Frage.

Bei der Zuordnung des primären Talentbaums „Elementar“ wird die Fähigkeit „Lavasog“ (Identifikationsnummer 77762) benutzt, die in dem Ereignis „SPELL\_CAST\_SUCCES“ vorkommt. Von der Mechanik der Datenabfrage her, gibt es keinen Unterschied zu den Vergleichen mit den Signaturfähigkeiten. Daher könnte die Fähigkeit „Lavasog“ auch in der Datenbanktabelle *static\_spell\_classtree* stehen, allerdings ist es keine Fähigkeit, die der Spieler selbst direkt auslösen kann, und passt deswegen nicht in das Schema der Datenbanktabelle.

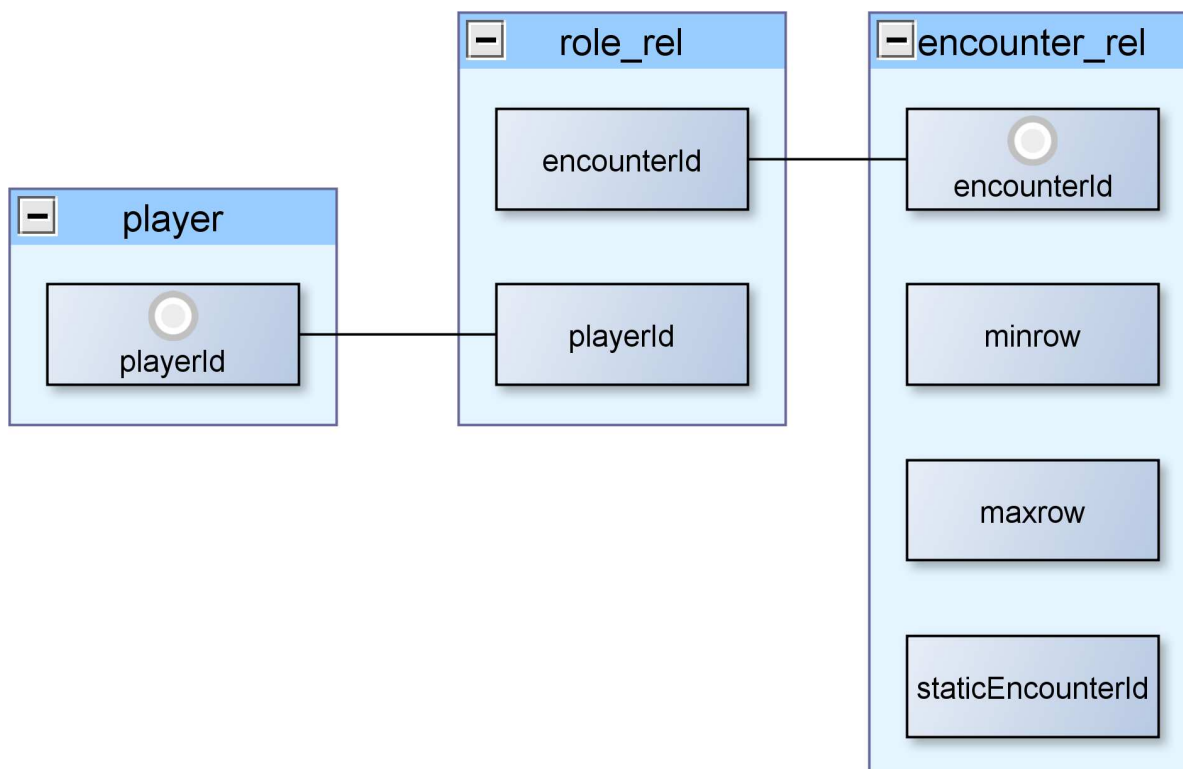
Die einzige Ausnahme bei der Zuordnung zwischen primärer Talentbaum und Rolle ist der primäre Talentbaum „Wilder Kampf“ der Klasse „Druide“. Dieser kann sowohl die Rolle des „Schadensfänger“ als auch des „Schadensausteiler(nah)“ ausfüllen. Die Rolle kann nur

anhand von statistischen Werten bestimmt werden. Prinzipiell wäre mit diesen Werten generell eine Klassifizierung der Rollen möglich, allerdings ist die Methode nach den primären Talentbäumen eindeutiger.

Zuerst wird auf die Rolle „Schadensfänger“ geprüft. Dies geschieht durch Ermittlung des durchschnittlichen eingehenden Schadens der Gruppe, während der betrachteten Begegnung. Der eingehende Schaden des untersuchten Spielers wird durch den Durchschnitt geteilt. Dabei kommt ein Quotient heraus, der angibt, wie viel mehr oder weniger Schaden der Spieler erlitten hat, als der Durchschnitt der Gruppe. Liegt dieser über 1,5 wird der Spieler als „Schadensfänger“ klassifiziert. Analog hierzu wird auch der Quotient für den ausgehenden Schaden gebildet. Liegt dieser über 0,5 wird der Spieler als „Schadensausteiler(nah)“ klassifiziert. Die niedrige Schranke ist nur möglich, weil die Klassifizierung als „Schadensfänger“ Vorrang hat. Sie ist allerdings auch nötig, weil die meisten Spieler „Schadensausteiler“ sind, und mit einem Quotient von 1 schwache „Schadensausteiler“ nicht klassifiziert werden würden.

Feldname	Typ	Beschreibung
encounterId	int(10) unsigned	Identifikationsnummer (eigenes Format)
playerId	int(10) unsigned	Identifikationsnummer des Spielers (eigenes Format)
classtree	Aufzählung(30)	Zugeordneter primärer Talentbaum
role	Aufzählung(4)	Zugeordnete Rolle
mintime	bigint(13) unsigned	Beginn der Aktivität des Spielers
maxtime	bigint(13) unsigned	Ende der Aktivität des Spielers

**Tabelle 23: Datenbanktabelle role\_rel**



**Abbildung 9: Verknüpfungen der Datenbanktabelle role\_rel**

Eine Übersicht der Verknüpfungen der Felder der Datenbanktabelle *role\_rel* zu anderen Datenbanktabellen gibt Abbildung 9: Verknüpfungen der Datenbanktabelle *role\_rel*.

Die Umsetzung der Zuordnung zwischen Spielern und Rolle wird durch folgende Datenbankabfrage ermöglicht:

```
UPDATE role_rel, event, encounter_rel, static_spell_classtree
SET role_rel.classtree = static_spell_classtree.classtree, role_rel.role = static_spell_classtree.role
WHERE (role_rel.encounterId = encounter_rel.encounterId) AND (event.rowId BETWEEN
encounter_rel.minrow AND encounter_rel.maxrow) AND (role_rel.playerId = event.sourceId)
AND (event.spellId = static_spell_classtree.spellId)
```

#### 4.3.4 Zuordnungsproblem Begleiter

Die Zuordnung von Begleitern zu den sie steuernden oder erschaffenden Charakteren ist in den Daten eines einzelnen Ereignisses leider nicht kodiert, wie es z.B. in den GUIDs denkbar wäre. Diese Zuordnung muss nun anders erfolgen, da auch die Begleiter zu der Gesamteffektivität eines Charakters beitragen. Durch bestimmte Ereignisse lassen sich viele Begleiter ihren Besitzern zuordnen.

Die Erschaffung von Begleitern ist ein spezielles Ereignis, dass den Suffix „\_SUMMON“ (Index 25) hat. Nach diesen kann nun gezielt in der Datenbanktabelle *event* gesucht werden und Beziehungen zwischen den Identifikationsnummern (Feldnamen *source* und *destId* über Feldname *playerId* aus der Datenbanktabelle *player*) von Begleitern (Feldname *petId*) und Besitzern (Feldname *masterId*) schaffen und in die Datenbanktabelle *pet\_rel* (siehe Tabelle 24: Datenbanktabelle *pet\_rel*) eintragen.

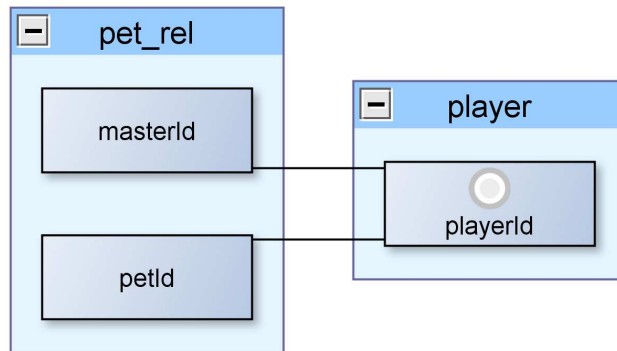
In speziellen Fällen werden auch Begleiter erschaffen, ohne dass das eben erwähnte Ereignis ausgelöst wird. Sofern diese Begleiter nicht mit bestimmten Ereignissen in Interaktion mit ihren Erschaffern stehen, ist diese Zuordnung nicht herstellbar. Bekannte Probleme stellen hierbei die Begleiter „Wächter der uralten Könige“ von allen drei verschiedenen primären Talentbäumen der Klasse Paladin und der Begleiter „Schattenhafte Erscheinung“ von dem primären Talentbaum „Schatten“ der Klasse Priester dar. Die nicht zugeordneten Begleiter führen dazu, dass die Effektivität dieser Klassen teilweise schlechter wiedergegeben wird, als sie eigentlich im Spiel ist.

Es kann allerdings auch vorkommen, dass Begleiter weitere Begleiter erschaffen, diese werden in der Datenbanktabelle *pet\_rel* dann transitiv den Besitzern des erschaffenden Begleiters zugeordnet.

Feldname	Typ	Beschreibung
<i>masterId</i>	int(10) unsigned	Identifikationsnummer des Besitzers (entspricht <i>playerId</i> aus der Datenbanktabelle <i>player</i> )
<i>petId</i>	int(10) unsigned	Identifikationsnummer des Begleiters (entspricht <i>playerId</i> aus der Datenbanktabelle <i>player</i> )

**Tabelle 24: Datenbanktabelle *pet\_rel***

Eine Übersicht der Verknüpfungen der Felder der Datenbanktabelle *pet\_rel* zu anderen Datenbanktabellen gibt Abbildung 10: Verknüpfungen der Datenbanktabelle *pet\_rel*.



**Abbildung 10: Verknüpfungen der Datenbanktabelle pet\_rel**

#### 4.3.5 Filterung

Wie in Abschnitt 3.4 Beschränkungen schon erwähnt, werden alle Ereignisse mit dem Suffix „\_CAST\_FAILED“ (Index 20) nur von dem aufzeichnenden Spieler aufgezeichnet, und sind somit bei den Analysen nicht sinnvoll verwendbar. Daher werden diese Ereignisse schon beim Parsen geblockt und nicht in die Datenbank eingetragen.

Ebenfalls beim Parsen geblockt werden alle Ereignisse mit dem Prefix „ENCHANT“ (Index 7), und daraus folgend auch die Ereignisse mit den Suffixen „\_APPLIED“ (Index 30) und „\_REMOVED“ (Index 31). Dies geschieht aus den in Abschnitt 3.4 Beschränkungen genannten Gründen. Zusätzlich kann das Feld itemId nicht korrekt einem Gegenstand zugeordnet werden, da mehrere Inkarnationen eines Gegenstandes mit der gleichen Basis-Identifikationsnummer existieren können.

Durch das Verpassen der optimalen Zeitpunkte für den Beginn und das Ende der Aufzeichnung, sowie durch eventuell vorgekommene Pausen, können sich Einträge von menschlichen Spielern in der Datenbank befinden, die nicht an den Hauptbegegnungen beteiligt waren. Solche Einträge sind unerwünscht und können mit den in diesem Kapitel beschriebenen Informationen gelöscht werden. Dabei werden alle Einträge in den Datenbanktabellen *event*, *player* und *pet\_rel* gelöscht, in denen menschliche (Feldname idType = 0) Spieler oder deren Begleiter vorkommen, die an keiner der zugeordneten Hauptbegegnungen teilgenommen haben. Dafür wird folgende Datenbankabfrage verwendet:

```

DELETE player, pet_rel, event
FROM player
LEFT JOIN role_rel ON (role_rel.playerId = player.playerId)
LEFT JOIN pet_rel ON (pet_rel.masterId = player.playerId)
JOIN event ON (event.sourceId = player.playerId OR event.destId = player.playerId OR event.sourceId =
pet_rel.petId OR event.destId = pet_rel.petId)
WHERE player.idType = 0 AND role_rel.playerId IS NULL
  
```

## 4.4 Gesamtübersicht der Datenbankverknüpfungen

Einen Überblick über alle Verknüpfungen zwischen allen nicht statischen Datenbanktabellen liefert Abbildung 11: Gesamtübersicht der Datenbankverknüpfungen. Dabei sind Schlüssel, die in anderen Datenbanktabellen Fremdschlüssel sind, speziell markiert.



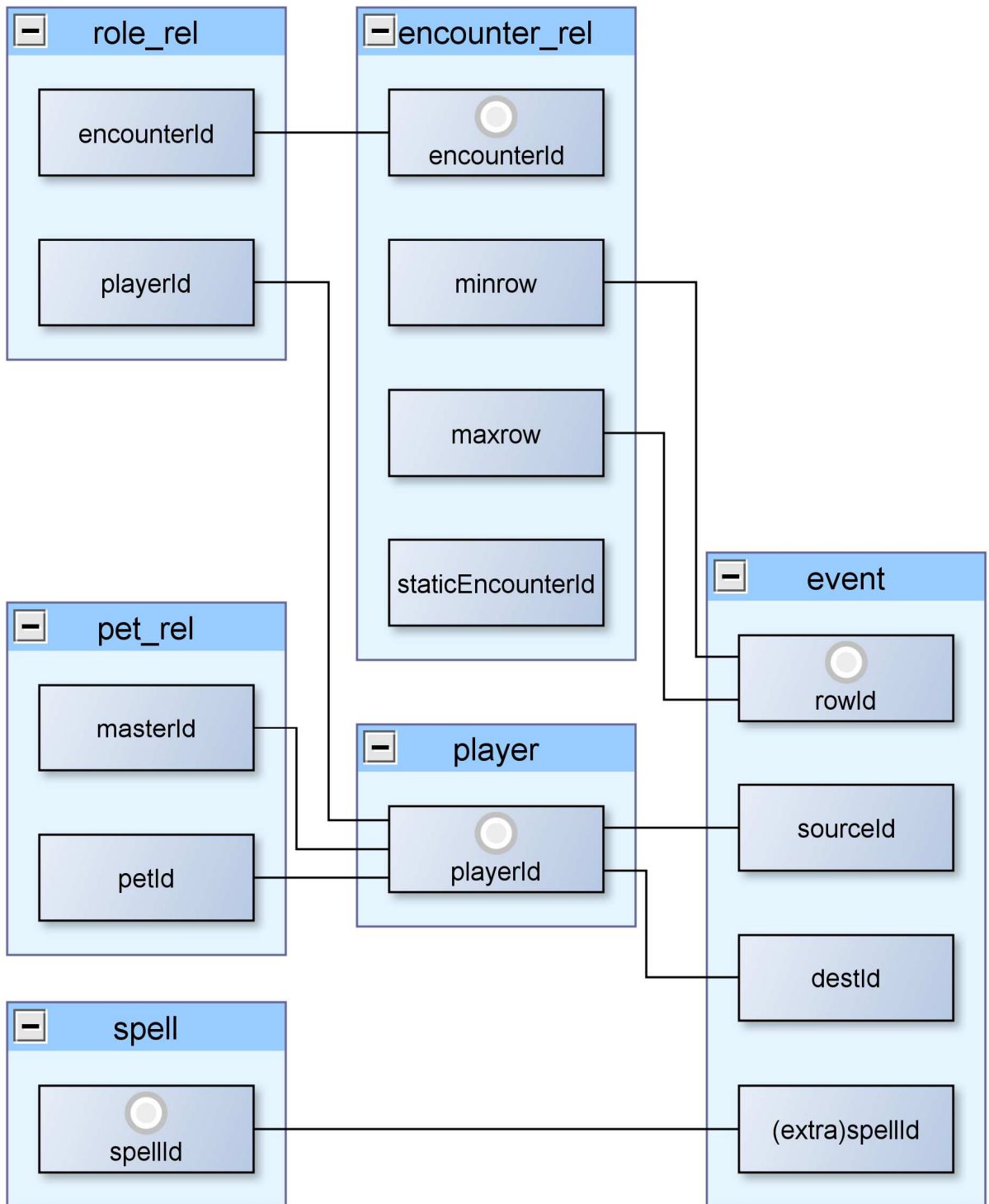


Abbildung 11: Gesamtübersicht der Datenbankverknüpfungen

## 5 Analysen und Präsentation

In diesem Kapitel wird beschrieben, welche Analysemöglichkeiten auf den extrahierten Daten mit dieser Arbeit ermöglicht werden. Begonnen wird mit einer kompakten Einführung in den allgemeinen Gebrauch solcher Dienste. Dies beinhaltet auch eine kleine Übersicht schon bestehender Dienste. Danach wird der Dienst dieser Arbeit im Detail vorgestellt. Dabei wird jede Web-Seite des Präsentationsteils einzeln und ausführlich vorgestellt, und auf die Erhebung bestimmter Analysen und spezielle Präsentationen eingegangen. Abgeschlossen wird das Kapitel mit den übergeordneten besonderen Funktionen, die keinem der

Abschnitte allein zugehörig sind. Die Einordnung in den Gesamtzusammenhang wird in Abbildung 12: Übersicht Präsentation veranschaulicht.

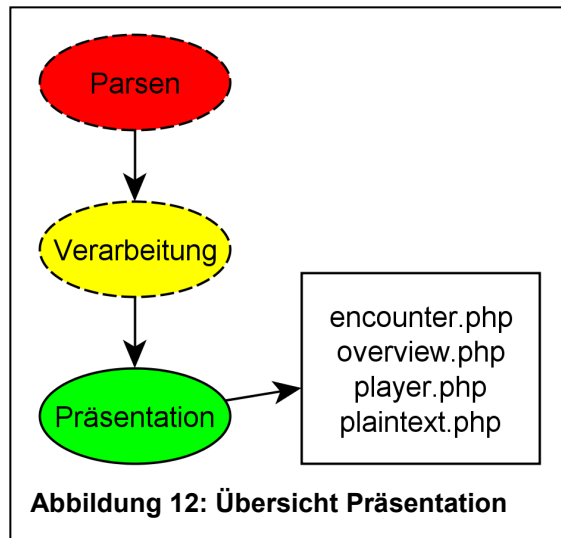


Abbildung 12: Übersicht Präsentation

### 5.1 Allgemeines

In diesem Abschnitt werden einige schon bestehende Webservices zur Auswertung von Combat Logs vorgestellt, und die Motivation für deren Benutzung erläutert. Es gibt zusätzlich auch nicht auf dem Internet basierende Dienste, auf die hier allerdings nicht weiter eingegangen wird.

Die Motivation zur Nutzung solcher Dienste besteht in dem Streben nach kontinuierlicher Verbesserung des individuellen Spielers oder des ganzen Teams, wie in 2.2.12 Streben nach Verbesserung schon beschrieben. Die Besonderheit der Webservices ist hierbei die visuelle Präsentation, Informationsvielfalt und dass von Web-Seiten, und damit in der Regel auch die eingeschränkten Bereiche, einfach die Verknüpfung mit anderen Benutzern geteilt werden kann. Zusätzlich hat bei diesen Diensten nur ein Administrator des Teams die Arbeit, alle Mitglieder des Teams aber den Nutzen des Dienstes.

Eine Übersicht der bekanntesten (wahrscheinlich auch die Einzigen, die aktuell aktiv und gepflegt sind) Webservices zur Auswertung von Combat Logs, inklusive einer Aufzählung ihrer besonderen Merkmale, findet sich in Tabelle 25: Übersicht bestehender Webservices zur Auswertung von Combat Logs.

Als Abgrenzung zwischen dem Dienst dieser Arbeit und den drei vorgestellten, werden die beliebige Darstellung von Fähigkeiten und Zustandsveränderungen als Graph für einzelne Charaktere, die Möglichkeit zur zeitlichen Manipulation und individuelle Schwachstellen betrachtet.

Der Webservice „Raidlogs“ [11] bietet keine weitere Möglichkeit zur zeitlichen Einschränkung, außer auf die erkannten Begegnungen. Zusätzlich gibt es keine Möglichkeit zur beliebigen Visualisierung von Fähigkeiten und Zustandsveränderungen in einem Graphen. Schwachstellen des Dienstes sind die geringe Anzahl an Zuordnungen von Begleitern und die geringe Anzahl an Informationen zu einzelnen Fähigkeiten.

Der Webservice „World of Logs“ [12] bietet die Möglichkeit zur freien zeitlichen Einschränkung, allerdings ist die nur über Markierungen in einem Graphen möglich, und somit nicht ganz genau möglich. Zusätzlich gibt es keine Möglichkeit zur beliebigen Visualisierung von Fähigkeiten und Zustandsveränderungen in einem Graphen. Die Schwachstelle sind die fehlenden Informationen zur Rolle und zum primären Talentbaum eines Spielers. Bei den übergeordneten Such-/Vergleichsoptionen kann aber eine Einschränkung nach diesen Informationen getroffen werden, allerdings werden diese Informationen auf der Ebene eines betrachteten Combat Logs nicht präsentiert.

Der Webservice „WoW Meter Online“ [13] bietet keine weitere Möglichkeit zur zeitlichen Einschränkung, außer auf die erkannten Begegnungen. Zusätzlich gibt es keine Möglichkeit zur beliebigen Visualisierung von Fähigkeiten und Zustandsveränderungen in einem Graphen. Eine einfache Schwachstelle sind die generell fehlenden Erklärungen zu den Tabellenspalten, was teilweise nicht intuitiv erfassbare (oder eindeutige) Werte zur Folge hat.

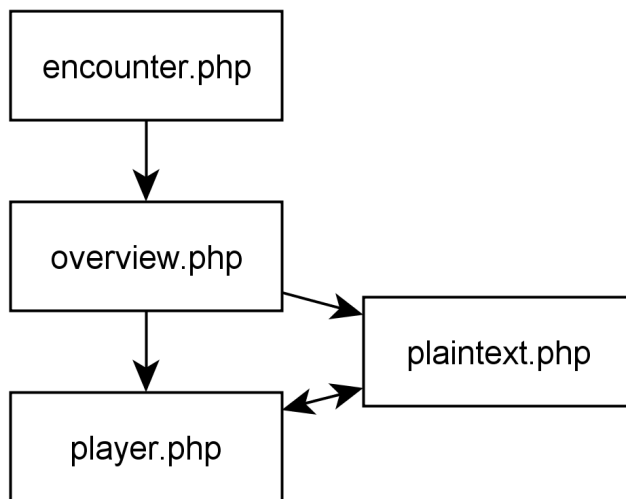
Webservice	besondere Merkmale
Raidlogs [11]	<ul style="list-style-type: none"> <li>▪ Kaum Zuordnung von Begleitern</li> <li>▪ Visuell beeindruckende Präsentation</li> <li>▪ Graphen klein, aber in der Regel mit direkter Zoom Manipulation</li> <li>▪ Keine Klartextausgabe</li> <li>▪ Vergleich von Daten gleicher Begegnungen und Klassen</li> <li>▪ Vergleich von verschiedenen Versuchen</li> <li>▪ Zuordnung von mehreren Combat Logs zu einer Gemeinschaft</li> <li>▪ Wenig Daten zu einzelnen Fähigkeiten</li> </ul>
World of Logs [12]	<ul style="list-style-type: none"> <li>▪ Begleiter werden zugeordnet (werden aber nicht auf der Detailansicht zu einzelnen Charakteren berücksichtigt)</li> <li>▪ Sehr viele Details zu Fähigkeiten und Zustandsveränderungen (durch einige Unterseiten sind die Informationen etwas verteilt)</li> <li>▪ Klartextausgabe</li> <li>▪ Vergleich von verschiedenen Versuchen</li> <li>▪ Graphen mit visuell ansprechender Präsentation (hierüber sind auch Manipulationen der zeitlichen Einschränkung möglich, auch über Begegnungsgrenzen hinaus)</li> <li>▪ Allgemein viele Manipulationsmöglichkeiten zur dynamischen Betrachtung der Daten (teilweise allerdings nicht direkt ersichtlich)</li> <li>▪ Zuordnung von mehreren Combat Logs zu einer Gemeinschaft</li> <li>▪ Keine Zuordnung von Rollen und primären Talentbäumen (zumindest nicht auf der Ebene eines betrachteten Combat Logs)</li> </ul>
WoW Meter Online [13]	<ul style="list-style-type: none"> <li>▪ Begleiter werden zugeordnet</li> <li>▪ Sehr viele Details zu Fähigkeiten und Zustandsveränderungen (durch viele Unterseiten sind diese Informationen teilweise schwer zu finden)</li> <li>▪ Klartextausgabe (nur bei wenigen der online gestellten Combat Logs möglich)</li> <li>▪ Graphen mit visuell ansprechender Präsentation</li> <li>▪ Bei einigen Tabellenspalten ist durch fehlende Erklärung nicht gleich klar, was sie zu bedeuten haben</li> <li>▪ Mehrere Varianten an Matrix-Darstellungen</li> <li>▪ Zuordnung von mehreren Combat Logs zu einer Gemeinschaft</li> </ul>

**Tabelle 25: Übersicht bestehender Webservices zur Auswertung von Combat Logs**

## 5.2 Web-Seiten

In diesem Abschnitt werden die Web-Seiten vorgestellt, die die eingelesenen und nachbearbeiteten Daten präsentieren und somit die eigentliche Benutzung des Dienstes darstellen. Der Standardeinstieg ist dabei die Gesamtübersicht aller Begegnungen (Web-

Seite encounter.php). Diese führt zu einer zeitlich eingeschränkten Übersicht aller Spieler (Web-Seite overview.php), und diese wiederum führt zur Einzelansicht von einem Spieler in dem vorher eingeschränkten zeitlichen Bereich (Web-Seite player.php). Von der zeitlich eingeschränkten Übersicht aller Spieler und der Einzelansicht von einem Spieler besteht die Möglichkeit zu einer Seite zu kommen, die die jeweiligen Ereignisse als einzelne verständliche Sätze ausgibt (Web-Seite plaintext.php). Einen Überblick der Web-Seiten und deren Verknüpfungen bietet Abbildung 13: Verknüpfung der Web-Seiten.



**Abbildung 13: Verknüpfung der Web-Seiten**

Eine ausführlichere Beschreibung der Web-Seiten folgt in den nächsten Abschnitten. Allgemein sind auf einigen der Web-Seiten Legenden und bei allen HTML-Tabellen Hinweistexte in den Spaltenüberschriften zur Erklärung der Begriffe zu finden. Durch Klicken einer HTML-Tabellenüberschrift ist es auch möglich die HTML-Tabelle nach der jeweiligen Spalte sortieren zu lassen. Dies wurde die Einbindung des JavaScripts von [14] umgesetzt.

### **5.2.1 Gesamtübersicht aller Begegnungen**

Die Gesamtübersicht aller Begegnungen (Web-Seite encounter.php) präsentiert allgemeine Informationen zu allen von diesem Dienst erkennbaren Hauptbegegnungen einer Combat Log-Datei. Diese Web-Seite ist der Standardeinstieg zu den Web-Seiten.

Am Anfang der Web-Seite werden einige statistische Daten zum zeitlichen Beginn, Ende und Dauer der Ereignisse in den Combat Log-Daten wiedergegeben. Danach folgt eine tabellarische Ausgabe der erkannten Hauptbegegnungen. Dabei entspricht jede Zeile der HTML-Tabelle einer Begegnung, deren Felder in Tabelle 26: HTML-Tabelle Gesamtübersicht aller Begegnungen beschrieben sind. Anschließend folgt eine Eingabemaske für die zeitliche Manipulation des Start- und Endzeitpunkts des betrachteten Zeitraums für die folgenden Web-Seiten. Abschließend ist eine Legende zu einzelnen Ausdrücken oder Abkürzungen aufgeführt.

Die ausgehenden Verknüpfungen dieser Web-Seite führen alle zu der zeitlich eingeschränkten Übersicht (Web-Seite overview.php). Hierbei führen die Verknüpfungen in der HTML-Tabelle in der Spalte Name zu einem genau auf die jeweilige Hauptbegegnung

zeitlich begrenzten Bereich. Dagegen führt die Eingabemaske für die zeitliche Manipulation zu einem beliebig, auch über Begegnungsgrenzen hinaus, eingeschränkten Bereich.

Spaltenname	Beschreibung
Startzeitpunkt	Startzeitpunkt der Begegnung in dem Format: Monat-Tag Stunde:Minute:Sekunde.Zehntel-/Hunderstel/Tausendstel-sekunde
Name	Name der Hauptbegegnung, der in der Datenbanktabelle <i>static_encounter</i> steht. Der Name dient als Verknüpfung zur genau auf diese Begegnung zeitlich begrenzten Übersicht.
Versuchsnummer	Die Versuchsnummer gibt an, um den wievielten Versuch einer Hauptbegegnung es sich handelt.
Erfolg	Erfolgsstatus der Hauptbegegnung mit den Werten „Wahr“ für erfolgreich und „Falsch“ für nicht erfolgreich.
Dauer	Dauer der Hauptbegegnung im Format: Stunde:Minute:Sekunde.Zehntel-/Hunderstel-/Tausendstelsekunde
Spieleranzahl	Anzahl der teilnehmenden Spieler, die mit dieser Begegnung verknüpft sind. Die Anzahl der teilnehmenden Spieler wird aus der Aufsummierung der Einträge in der Datenbanktabelle <i>role_rel</i> ermittelt.
Rollenverteilung (SF, H, SAN, SAF)	Rollenverteilung, die aus der Datenbanktabelle <i>role_rel</i> ausgelesen wird. (SF = Schadensfänger, H = Heiler, SAN = Schadensausteiler(nah), SAF = Schadensausteiler(fern))
Klassenverteilung (D, H, K, J, M, Pa, Pr, Sa, Su, T)	Klassenverteilung, die aus der Datenbanktabelle <i>role_rel</i> ausgelesen wird. (D = Druiden, H = Hexenmeister, K = Krieger, J = Jäger, M = Magier, Pa = Paladin, Pr = Priester, Sa = Schamane, Su = Schurke, T = Todesritter)

**Tabelle 26: HTML-Tabelle Gesamtübersicht aller Begegnungen**

Nachfolgend steht die Datenbankabfrage zur Ermittlung Daten für die HTML-Tabelle:

```
SELECT encounter_rel.encounterId, encounter_rel.succes, static_encounter.name, encounter_rel.minrow
AS minrow, encounter_rel.maxrow AS maxrow, encounter_rel.mintime AS mintime,
encounter_rel.maxtime AS maxtime, role_rel.playerId, class_rel.class, role_rel.role, encounter_rel.try
FROM encounter_rel STRAIGHT_JOIN static_encounter ON (encounter_rel.staticEncounterId =
static_encounter.staticEncounterId)
STRAIGHT_JOIN role_rel ON (encounter_rel.encounterId = role_rel.encounterId)
STRAIGHT_JOIN class_rel ON (role_rel.playerId = class_rel.playerId)
GROUP BY encounter_rel.encounterId, role_rel.playerId
```

### 5.2.2 Übersicht (zeitlich eingeschränkt)

Die zeitlich eingeschränkte Übersicht (Web-Seite *overview.php*) präsentiert spezifische Informationen zu den Spielern, also dem Team, in dem von der Gesamtübersicht aller Begegnungen (Web-Seite *encounter.php*) bestimmten Bereich. Die ausgegebenen Daten sind darauf ausgelegt, dass es sich bei dem eingeschränkten Bereich um eine Hauptbegegnung handelt, oder zumindest einen Unterbereich von einer, was allerdings auch die häufigste Verwendung der Web-Seite sein sollte. Diese Web-Seite ist nur von der Gesamtübersicht aller Begegnungen erreichbar.

Am Anfang der Web-Seite werden einige statistische Daten zum zeitlichen Beginn, Ende und Dauer der Ereignisse, und eventuell übergeordneter Hauptbegegnungen, des eingeschränkten Bereichs wiedergegeben. Danach folgt eine Verknüpfung zur Klartextausgabe der einzelnen Ereignisse des betrachteten Zeitraums. Anschließend folgt eine tabellarische Ausgabe der beteiligten menschlichen Spieler. Dabei entspricht jede Zeile

der HTML-Tabelle einem Spieler, deren Felder in Tabelle 27: HTML-Tabelle zeitlich eingeschränkte Übersicht beschrieben sind. Abschließend an der HTML-Tabelle ist eine Legende zu einzelnen Ausdrücken oder Abkürzungen aufgeführt.

Anschließend kommt der Bereich mit der Heil- und Schadensmatrix. Bei der Heilmatrix stehen in den Zeilen die Quellen und in den Spalten die Ziele der Heilungen. In den Zeilen der Matrix stehen die auf die Ziele (Spalte) entfallenden prozentualen Anteile an der Gesamtheilung der Quelle (Zeile). Quellen und Ziele beschränken sich dabei auf menschliche Spieler. Bei der Schadensmatrix stehen in den Spalten die Quellen und in den Zeilen die Ziele der Schäden. In den Spalten der Matrix stehen die auf die Ziele (Zeile) entfallenden prozentualen Anteile an dem verursachten Gesamtschaden der Quelle (Spalte). Quellen beschränken sich hierbei auf menschliche Spieler und Ziele beschränken sich auf nicht menschliche Spieler. Durch die in den Matrizen enthaltenen Informationen besteht die Möglichkeit individuelle Unterschiede in der Strategie, Effizienz, Vorteile, Besonderheiten und Verhalten der einzelnen Spieler zu erkennen, auch wenn sie derselben Rolle zugeordnet sind.

Die ausgehenden Verknüpfungen in der HTML-Tabelle in der Spalte Name führen zur Einzelansicht eines Spielers (Web-Seite `player.php`) in dem vorher eingeschränkten zeitlichen Bereich. Die Verknüpfung zur Klartextausgabe (Web-Seite `plaintext.php`) erfolgt mit den gleichen zeitlichen Einschränkungen.

Spaltenname	Beschreibung
Name	Name des Spielers, der in der Datenbanktabelle <i>player</i> steht. Der Name dient als Verknüpfung zur Einzelübersicht dieses Spielers in dem gleichen eingeschränkten Zeitraum wie diese Web-Seite.
primärer Talentbaum	Primärer Talentbaum (inklusive des Klassennamens) des Spielers, sofern es eine Hauptbegegnung ist. Ausgelesen aus der Datenbanktabelle <i>role_rel</i>
Rolle	Rolle des Spielers, sofern es eine Hauptbegegnung ist. Ausgelesen aus der Datenbanktabelle <i>role_rel</i>
Aktivität	Aktivität des Spielers, gemessen in Prozent, aus der Differenz von dem zeitlichen Abstand zwischen dem ersten und letzten Ereignis mit Beteiligung des Spielers in der Begegnung und der Dauer der Hauptbegegnung selbst.
Schaden( raus )	Summe der Wertigkeiten der effektiven Ereignisse bezüglich des ausgehenden Schadens des Spielers.
SPS( raus )	ausgehender Schaden pro Sekunde
aSPS( raus )	ausgehender Schaden pro Sekunde in der aktiven Zeit des Spielers (SPS(raus) / Aktivität)
Heilung	Summe der Wertigkeiten der effektiven Ereignisse bezüglich der Heilung des Spielers.
HPS	Heilung pro Sekunde
aHPS	Heilung pro Sekunde in der aktiven Zeit des Spielers (HPS / Aktivität)
Schaden( rein )	Summe der Wertigkeiten der effektiven Ereignisse bezüglich des eingehenden Schadens des Spielers.
SPS( rein )	eingehender Schaden pro Sekunde
aSPS( rein )	eingehender Schaden pro Sekunde in der aktiven Zeit des Spielers (SPS(rein) / Aktivität)

**Tabelle 27: HTML-Tabelle zeitlich eingeschränkte Übersicht**

Nachfolgend steht eine exemplarische Datenbankabfrage zur Ermittlung Daten für die HTML-Tabelle:

```
SELECT player.playerId, player.name, role_rel.classtree, role_rel.role, role_rel.maxtime,
role_rel.mintime, sum(event.amount) AS damageOut
FROM event
STRAIGHT_JOIN encounter_rel ON (encounter_rel.encounterId = $encounterId)
STRAIGHT_JOIN role_rel ON (encounter_rel.encounterId = role_rel.encounterId)
STRAIGHT_JOIN player ON (role_rel.playerId = player.playerId AND (event.sourceId = player.playerId
OR event.sourceId IN (SELECT pet_rel.petId FROM pet_rel WHERE pet_rel.masterId =
role_rel.playerId)))
WHERE (event.rowId BETWEEN $minrow AND $maxrow) AND event.suffix = 0
GROUP BY player.playerID
ORDER BY damageOut DESC
```

### **5.2.3 Einzelübersicht – Spieler**

Die Einzelübersicht Spieler (Web-Seite player.php) präsentiert alle Informationen für einen Spieler in einem zeitlich eingeschränkten Bereich. Diese Web-Seite ist von der zeitlich eingeschränkten Übersicht, sich selbst und der Klartextausgabe erreichbar. Die zeitliche Einschränkung wird dabei jeweils von der vorherigen Web-Seite übernommen.

Am Anfang der Web-Seite werden einige statistische Daten zum betrachteten Spieler, zeitlichen Beginn, Ende und Dauer der Ereignisse, und eventuell übergeordneter Hauptbegegnungen, des eingeschränkten Bereichs wiedergegeben. Danach folgen mehrere Verknüpfungen zur Klartextausgabe der einzelnen Ereignisse des betrachteten Zeitraums, an denen der Spieler beteiligt ist. Diese Verknüpfungen unterscheiden sich darin, ob nur Ereignisse bei denen der Spieler die Quelle, das Ziel oder beides ist, aufgelistet werden. Anschließend folgen mehrere tabellarische Ausgaben für die ein- und ausgehenden Fähigkeiten und Zustandsveränderungen. Abschließend kann mittels eines Graphen der zeitliche Verlauf von Fähigkeiten und Zustandsveränderungen visualisiert werden. Die tabellarischen Ausgaben, sowie der Graph, werden in den nächsten Abschnitten näher beschrieben.

Die ausgehenden Verknüpfungen in den HTML-Tabellen in der Spalte Info(ID) führen zu detaillierten Informationen zu Fähigkeiten bei dem Online-Dienst von Buffed [10]. Die ausgehende Verknüpfung zur Zeichnung des Graphen (Knopf „Graph zeichnen“ unter jeder HTML-Tabelle) führt zu der Web-Seite selbst zurück, allerdings mit zusätzlichen Informationen, die benötigt werden, um den Graph zu zeichnen. Weitere ausgehende Verknüpfungen führen zur Klartextausgabe (Web-Seite plaintext.php), wie oben schon erwähnt.

#### **5.2.3.1 Schaden (ausgehend)**

Im Abschnitt Schaden (ausgehend), der eine besondere Bedeutung für die Rollen Schadensaussteiler(nah) und Schadensaussteiler(fern) hat, werden tabellarisch Informationen zu den einzelnen Fähigkeiten des Spielers angegeben, mit denen er Schaden an Zielen erzeugt hat. Dabei entspricht jede Zeile der HTML-Tabelle einer Fähigkeit, deren Felder in Tabelle 28: HTML-Tabelle Schaden (ausgehend) beschrieben sind. Für etwaige dem Spieler zugehörige Begleiter werden getrennte tabellarische Ausgaben erzeugt. Abschließend wird

die Gesamteffizienz des Spielers zusammengefasst, die auch die zugehörigen Begleiter mit einbezieht.

Das Balkendiagramm in der Spalte Balken in der HTML-Tabelle ist eine Visualisierung der Verteilung der Kategorien der Angriffe des Spielers. Dabei orientiert sich die Darstellung an der Theorie der sogenannten „Attack Table“ [15]. Diese Theorie versucht zu beschreiben, wie von der Spielmechanik die Wahrscheinlichkeit der Kategorie eines Angriffs ermittelt wird, und vor allem in welcher Reihenfolge dies geschieht. Dabei werden bis zum Erreichen von einer Wahrscheinlichkeit von 100% die Wahrscheinlichkeiten der Ereignisse verfehlt, gestreift, geblockt, kritisch und krachend in dieser Reihenfolge berechnet, ein eventueller Rest ist als normaler Treffer anzusehen. Durch hohe Wahrscheinlichkeiten einzelner Kategorien ist es auch möglich, dass nachfolgende Kategorien gar nicht mehr beachtet werden, da die maximale Wahrscheinlichkeit schon erreicht ist. Unbeachtet bleiben hierbei Werte für widerstanden und absorbiert, da sie unabhängig von den vorher genannten Kategorien auftreten können.

Spaltenname	Beschreibung
Graph	Checkbox für die Auswahl der zu zeichnenden Daten in dem Graph
Name	Name der Fähigkeit
Schule	Schule der Fähigkeit
$\Sigma$ Schaden	Effektivität der Fähigkeit
$\emptyset$ Schaden	durchschnittlicher Schaden der Fähigkeit ( $\Sigma$ Schaden / Gesamtanzahl der Ereignisse)
aSPS	Effektiver Schaden pro Sekunde (aSPS = Amount / Aktive Zeit der Spielers)
%	Prozentualer Anteil am Gesamtschaden
#E	Anzahl der gesamten Ereignisse der Fähigkeit
Verfehlt	Anzahl der verfehlten Ereignisse, in Klammern steht der prozentuale Anteil
Gestreift	Anzahl der streifenden Ereignisse, in Klammern steht der prozentuale Anteil
Geblockt	Anzahl der geblockten Ereignisse, in Klammern steht der prozentuale Anteil
Kritisch	Anzahl der kritischen Ereignisse, in Klammern steht der prozentuale Anteil
Krachend	Anzahl der krachenden Ereignisse, in Klammern steht der prozentuale Anteil
Treffer	Anzahl der gewöhnlichen Treffer, in Klammern steht der prozentuale Anteil
Balken	Balkendiagramm der Verteilung der Kategorien der Angriffe des Spielers. Die zugehörigen Tabellenspalten sind entsprechend farblich markiert.
Widerstanden	Anzahl der widerstandenen Ereignisse, in Klammern steht der prozentuale Anteil
Absorbiert	Anzahl der absorbierten Ereignisse, in Klammern steht der prozentuale Anteil
$\Sigma$ Verfehlt	Summe der Wertigkeiten der verfehlten Ereignisse
$\Sigma$ Geblockt	Summe der Wertigkeiten der geblockten Ereignisse
$\Sigma$ Widerstanden	Summe der Wertigkeiten der widerstandenen Ereignisse
$\Sigma$ Absorbiert	Summe der Wertigkeiten der absorbierten Ereignisse
Info (ID)	externer Dienst

**Tabelle 28: HTML-Tabelle Schaden (ausgehend)**

Nachfolgend steht die Datenbankabfrage zur Ermittlung Daten für die HTML-Tabelle, dabei steht \$playerId für die Identifikationsnummer des Charakters, \$minrow für das erste Ereignis im betrachteten Zeitraum und \$maxrow für das letzte Ereignis im betrachteten Zeitraum:

```
SELECT event.spellId, spell.spellName, sum(event.amount) AS sumamount, count(event.amount) AS countamount, count(event.critical) AS countcrit, count(event.glancing) AS countglancing, count(event.crushing) AS countcrushing, sum(event.resisted) AS sumresisted, sum(event.blocked) AS
```



```

sumblocked, sum(event. absorbed) AS sumabsorbed, count(event.resisted) AS countresisted,
count(event.blocked) AS countblocked, count(event. absorbed) AS countabsorbed, count(event.missType)
AS countmissed, sum(event.amountMissed) AS summissed, event.sourceId AS sourceId,
spell.spellSchool AS spellSchool
FROM event
LEFT JOIN spell ON event.spellId = spell.spellId
WHERE event.suffix IN (0, 1) AND (event.sourceId = $playerId OR event.sourceId IN (SELECT
pet_rel.petId FROM pet_rel WHERE pet_rel.masterId = $playerId)) AND event.rowId BETWEEN
$minrow AND $maxrow
GROUP BY event.sourceId, event.spellId
ORDER BY event.sourceId ASC, sumamount DESC

```

### 5.2.3.2 Heilung

Im Abschnitt Heilung, der eine besondere Bedeutung für die Rolle Heiler hat, werden tabellarisch Informationen zu den einzelnen Fähigkeiten des Spielers angegeben, mit denen er Heilung an Zielen erzeugt hat. Dabei entspricht jede Zeile der HTML-Tabelle einer Fähigkeit, deren Felder in Tabelle 29: HTML-Tabelle Heilung beschrieben sind. Für etwaige dem Spieler zugehörige Begleiter werden getrennte tabellarische Ausgaben erzeugt. Abschließend wird die Gesamteffizienz des Spielers zusammengefasst, die auch die zugehörigen Begleiter mit einbezieht.

Auf eine entsprechende Aufschlüsselung von eingehenden Heilungen nach Fähigkeiten wird hier verzichtet, weil es in der Regel keine interessanten Informationen für Begegnungen liefern würde.

Spaltenname	Beschreibung
Graph	Checkbox für die Auswahl der zu zeichnenden Daten in dem Graph
Name	Name der Fähigkeit
Schule	Schule der Fähigkeit
$\Sigma$ Heilung	Effektivität, gibt die aufsummierten Wertigkeiten der Fähigkeit an
$\emptyset$ Heilung	durchschnittliche Heilung der Fähigkeit ( $\Sigma$ Heilung / Gesamtanzahl der Ereignisse)
aHPS	aktive Heilung pro Sekunde (aHPS = Amount / Aktive Zeit der Spielers)
%	Prozentualer Anteil an der Gesamtheilung des Spielers
#E	Anzahl der gesamten Ereignisse der Fähigkeit
Kritisch	Anzahl der kritischen Ereignisse, in Klammern steht der prozentuale Anteil
Überheilung	Anzahl der überheilten Ereignisse, in Klammern steht der prozentuale Anteil
$\Sigma$ Überheilung	Summe der Wertigkeiten der überheilten Ereignisse
Absorbiert	Anzahl der absorbierten Ereignisse, in Klammern steht der prozentuale Anteil
$\Sigma$ Absorbiert	Summe der Wertigkeiten der absorbierten Ereignisse
Info (ID)	externer Dienst

**Tabelle 29: HTML-Tabelle Heilung**

Nachfolgend steht die Datenbankabfrage zur Ermittlung Daten für die HTML-Tabelle, dabei steht \$playerId für die Identifikationsnummer des Charakters, \$minrow für das erste Ereignis im betrachteten Zeitraum und \$maxrow für das letzte Ereignis im betrachteten Zeitraum:

```

SELECT event.spellId, spell.spellName, sum(event.amount) AS sumamount, count(event.amount) AS
countamount, count(event.critical) AS countcrit, count(event.glancing) AS countglancing,
count(event.crushing) AS countcrushing, sum(event.resisted) AS sumresisted, sum(event.blocked) AS

```

```

sumblocked, sum(event.absorbed) AS sumabsorbed, count(event.resisted) AS countresisted,
count(event.blocked) AS countblocked, count(event.absorbed) AS countabsorbed,
count(event.amountMissed) AS countmissed, sum(event.amountMissed) AS summissed,
count(event.overhealing) AS countoverhealing, sum(event.overhealing) AS sumoverhealing,
event.sourceId AS sourceId, spell.spellSchool AS spellSchool
FROM event
LEFT JOIN spell ON (event.spellId = spell.spellID)
WHERE event.suffix IN (2) AND (event.sourceId = $playerId OR event.sourceId IN (SELECT
pet_rel.petId FROM pet_rel WHERE pet_rel.masterId = $playerId)) AND (event.rowId BETWEEN
$minrow AND $maxrow)
GROUP BY event.sourceId, event.spellId
ORDER BY event.sourceId ASC, sumamount DESC

```

### 5.2.3.3 Schaden (eingehend)

Im Abschnitt Schaden (eingehend), der eine besondere Bedeutung für die Rolle Schadensfänger hat, werden tabellarisch Informationen zu den einzelnen Fähigkeiten angegeben, die an dem Spieler Schaden erzeugt haben. Dabei entspricht jede Zeile der HTML-Tabelle einer Fähigkeit. Dabei entspricht der Aufbau der HTML-Tabelle genau dem in Abschnitt 5.2.3.1 Schaden (ausgehend) beschriebenen, und die Felder entsprechen auch denen in Tabelle 28: HTML-Tabelle Schaden (ausgehend). Abschließend wird die Gesamteffizienz der auf den Spieler gerichteten Angriffe zusammengefasst. Begleiter werden in diesem Abschnitt nicht berücksichtigt.

### 5.2.3.4 Zustandsveränderungen

Der Abschnitt Zustandsveränderungen (in Abschnitt 2.2.9 Zustandsveränderungen bereits erwähnt) besteht aus den vier Bereichen: Positive Zustandsveränderung (auf Charakter), Negative Zustandsveränderung (auf Charakter), Positive Zustandsveränderung (auf Ziel) und Negative Zustandsveränderung (auf Ziel). Die Bereiche unterscheiden sich nur in der Aufteilung zu jeweils negativen und positiven Zustandsveränderungen auf dem betrachteten Charakter oder seinen Zielen, allerdings werden auf Zielen nur die vom betrachteten Charakter ausgelösten Zustandsveränderungen berücksichtigt. Die Informationen werden hierbei jeweils in einer HTML-Tabelle wiedergegeben, wobei jede Zeile einer Fähigkeit entspricht, und deren Felder in Tabelle 30: HTML-Tabellen Zustandsveränderungen beschrieben sind. Begleiter werden generell nicht berücksichtigt bei der Ermittlung der Zustandsveränderungen.

Spaltenname	Beschreibung
Graph	Checkbox für die Auswahl der zu zeichnenden Daten in dem Graph
Name	Name der positiven/negativen Zustandsveränderung
#E	Anzahl der Ereignisse
Dauer	Dauer der positiven/negativen Zustandsveränderung
Info(ID)	externer Dienst

**Tabelle 30: HTML-Tabellen Zustandsveränderungen**

### 5.2.3.5 Graph

Im Abschnitt Graph können mit Hilfe eines Graphen verschiedene Fähigkeiten und Zustandsveränderungen visualisiert werden. Hierbei ist es auch möglich beliebig viele Kombinationen aus Fähigkeiten und Zustandsveränderungen gleichzeitig anzeigen zu lassen, zumindest solange die kombinierte Anzahl 20 nicht übersteigt. Alle gezeichneten Elemente haben verschiedene Farben, die in einer Legende oberhalb des Graphen beschrieben sind. Die Auswahl der zu zeichnenden Elemente erfolgt über Checkbox-Elemente in den HTML-Tabellen und bei der Zusammenfassung der Gesamteffizienz in den einzelnen Abschnitten. Deren Status wird bei einem Klick auf den Knopf „Graph zeichnen“ berücksichtigt, und bei dem darauf automatisch folgendem Wiederaufruf der HTML-Seite markiert.

Die horizontale Achse stellt die Zeitpunkte dar, relativ gesehen zum Anfang des betrachteten Zeitraums, und ist in regelmäßigen Abständen beschriftet. Die vertikale Achse stellt für Fähigkeiten die Wertigkeit da, und ist ebenfalls in regelmäßigen Abständen beschriftet. Zustandsveränderungen werden als Fläche dargestellt, deren Breite die Wirkungszeiträume markiert und die Höhe keine weitere Bedeutung hat. Fähigkeiten werden dabei als durchgehende Linie gezeichnet, die die Wertigkeiten über dem Verlauf der Zeit darstellt. Für ein optisch verwertbareres Bild werden die Daten zunächst über einem kleinen Zeitraum geglättet, und anschließend mittels einer Kurvenfunktion dargestellt.

Bei der Generierung der Daten für den Graphen wird außer den Identifikationsnummern der Fähigkeiten und ob es sich um eine Zustandsveränderung handelt kein weiterer Kontext berücksichtigt. Ausgenommen hiervon sind: Nahkampfschaden(eingehend), Nahkampfschaden(ausgehend), Gesamtschaden(eingehend), Gesamtschaden(ausgehend), Gesamtheilung.

Ein Beispiel für einen gezeichneten Graph findet sich in Abbildung 14: Beispiel für einen Graphen. Bei dem Graph handelt es sich um eine Zustandsveränderung (senkrechten Balken) und zwei Fähigkeiten (zwei Kurven). Die Zustandsveränderung bewirkt, dass beide Fähigkeiten effektiver werden, verbraucht sich aber bei der Benutzung einer der beiden Fähigkeiten. Daher ist es erstrebenswert die Zustandsveränderung für die stärkere Fähigkeit (Grüne Kurve) zu nutzen. Auf Grund von Mangel an Ressourcen kann es auch vorkommen, dass die stärkere Fähigkeit (grüne Kurve) nicht eingesetzt werden kann. In dem Graph ist nun erkennbar, wie gut es dem Charakter gelungen ist die Zustandsveränderung mit dem Einsatz der stärkeren Fähigkeit (Ausschläge der grünen Kurve) zu koordinieren. Es fällt dabei auf, dass häufig die schwächere Fähigkeit (blaue Kurve) die Zustandsveränderung verbraucht hat. Das bedeutet, dass der Spieler hier Optimierungspotential hat.

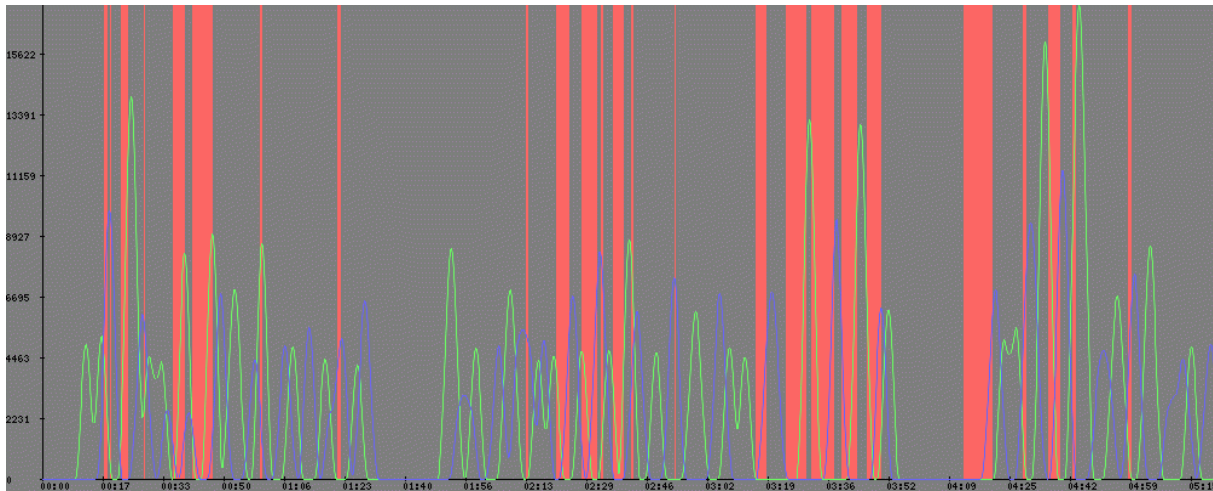


Abbildung 14: Beispiel für einen Graphen

#### 5.2.4 Klartextausgabe

Die Klartextausgabe (Web-Seite plaintext.php) präsentiert alle Informationen für den eingeschränkten Bereich als allgemein verständliche Sätze. Diese Web-Seite ist von der zeitlich eingeschränkten Übersicht und der Einzelübersicht Spieler erreichbar. Sämtliche Einschränkungen werden dabei jeweils von der vorherigen Web-Seite übernommen. Durch diese Kontextabhängigkeit der Klartextausgabe, ist es nur möglich Einschränkungen über die aufrufenden Web-Seiten zu machen.

Mögliche Einschränkungen für die Klartextausgabe sind: der zeitliche Bereich, der Charakter (auch Objekte möglich), nur als Quelle (bezogen auf einen Charakter), nur als Ziel (bezogen auf einen Charakter) und ohne weitere Einschränkungen für Quelle und Ziel.

Die Ausgabe der Web-Seite besteht aus mehreren allgemein verständlichen Sätzen, wobei jeder Satz einer Zeile in der Combat Log-Datei entspricht. Jeder Satz beginnt mit dem Zeitstempel des Ereignisses, dem das ausformulierte Ereignis dann folgt. Dabei wird die Satzstruktur von den Feldern prefix und suffix des Ereignisses bestimmt, wobei der Einfluss der beiden Felder unabhängig voneinander ist.

Die ausgehenden Verknüpfungen in den einzelnen Zeilen sind jeweils die Namen der Quelle und des Ziels, sofern vorhanden, und führen zu der Einzelübersicht Spieler. Diese Verknüpfung ermöglicht es von jedem Charakter (oder Objekt), vor allem Nicht-Spieler-Charaktere, die Einzelübersicht Spieler zu bekommen.

Es folgen einige von der Klartextausgabe generierte Beispielsätze:

04-03 19:34:43.984: Spieler Taran's Nahkampfangriff hat NSC Theralion nicht getroffen, weil pariert.

04-03 19:34:44.171: Spieler Lismera's Siegel der Einsicht(Heilig) hat Spieler Lismera 3,513 Mana erzeugt.

04-03 19:34:44.390: Spieler Mohiam's Nahkampfangriff hat NSC Theralion 2,968 streifenden Schaden(Physisch) zugefügt.

04-03 19:33:05.843: NSC Totem des heilenden Flusses's periodische Wirkung von Totem des heilenden Flusses hat Spieler Zondra für 1,554 geheilt.

04-03 19:33:14.843: Spieler Mohiam's Auslöschen(Physisch) hat NSC Valiona 32,280 kritischen Schaden(Physisch) zugefügt.

04-03 19:33:19.500: Spieler Lismera's Lichtblitz(Heilig) hat Spieler Hilflos für 30,665(davon 3,871 überheilt) kritisch geheilt.

### 5.3 Besondere Funktionen

Eine besondere Funktion des Dienstes ist das Adresssystem der Web-Seiten, das zur Einschränkung der präsentierten Informationen führt. Die Parameter der URLs und deren Bedeutung werden in Tabelle 31: Adresssystem beschrieben.

Name	Werte	Beschreibung
minrow	Integer	untere zeitliche Schranke als rowId von Datenbanktabelle <i>event</i>
maxrow	Integer	obere zeitliche Schranke als rowId von Datenbanktabelle <i>event</i>
playerid	Integer	Identifikationsnummer des betrachteten Charakter aus der Datenbanktabelle <i>player</i>
encounterid	Integer	Identifikationsnummer der betrachteten Begegnung aus der Datenbanktabelle <i>encounter_rel</i>
range	(s, c, b)	Eigenschaft des betrachteten Bereichs: s = sub: Unterbereich einer Begegnung c = complete: kompletter Bereich einer Begegnung f = free: frei gewählter Bereich – zeitliche Schranken nicht innerhalb einer Begegnung
context	(s, d, b)	betrachteter Kontext des Charakters (kommt nur für die Web-Seite plaintext.php vor): s = source: Charakter nur als Quelle d = dest: Charakter nur als Ziel b = both: Charakter als beides

**Tabelle 31: Adresssystem**

## 6 Zusammenfassung und Ausblick

In dieser Arbeit wurde die Entwicklung eines Webservices für die Auswertung und Analyse von Combat Logs des MMORPGs World of Warcraft® beschrieben. Angefangen wurde mit einer Einführung in das Spiel World of Warcraft®, währenddessen auch die Aspekte erklärt wurden, die für den Motivationsbereich dieser Arbeit wichtig sind. Nachfolgend wurde ausführlich die Struktur des Combat Logs erläutert, wobei auch auf die Bedeutung und Eigenheiten jedes Parameters eingegangen wurde. Anschließend wurde die zur Speicherung, der vom Parser generierten Daten, verwendete Datenstruktur der Datenbank beschrieben. Zusätzlich wurden statische Informationen vorgestellt, die im Folgeschritt für einen Teil der Nachbearbeitung der Daten nötig waren. In der Nachbearbeitung der Daten wurden darin enthaltene übergeordnete Informationen untersucht und ebenfalls in Datenbanken gespeichert, bei denen es sich teilweise um Klassifikationen handelte. Abschließend wurde die Präsentation des Webservices dargestellt. Dabei wurden die verschiedenen Visualisierungen und die dadurch vermittelten Informationen ausführlich beschrieben. Zuvor wurden schon bestehende Webservices anhand einiger Kriterien abgegrenzt, die ein Schwerpunkt bei der Entwicklung des Webservices dieser Arbeit waren.

Diese Kriterien, nämlich eine exakte Zeitmanipulation und die beliebige Anzeige von Fähigkeiten und Zustandsveränderungen als Graph, stellen Alleinstellungsmerkmale dieses Webservices gegenüber den bestehenden Webservices dar. Zusätzlich werden die Schwachstellen der anderen Webservices besser gelöst. Die HTML-Tabellen sind generell mit Hinweistexten versehen, die die genauere Bedeutung erläutern. Die Informationen zu den Fähigkeiten sind umfassend, und möglichst viele Begleiter werden ihren Besitzern zugeordnet. Rollen und primäre Talentbäume werden Spielern zugeordnet, sofern es in dem jeweiligen Kontext sinnvoll ist, außerdem war dies ein Schwerpunkt der Arbeit.

Das Hauptaugenmerk bei der Entwicklung dieses Webservices lag auf der Visualisierung der Informationen und dem eigentlichen Nutzungsbereich, daher entstanden Schwachstellen im administrativen Bereich. Die Dauer des Parsens beträgt in der Regel mehrere Minuten, und ist somit viel höher als die Standardbeschränkung von 30 Sekunden für die Ausführung eines PHP-Skripts. Die Nachbearbeitung der Daten kann ebenfalls länger als diese Standardbeschränkung dauern. Da immer nur ein Combat Log mit dem Webservice betrachtet werden kann, fehlt ein Vergleich zwischen mehreren Combat Logs. Daher fehlt auch eine Zuordnung von mehreren Combat Logs zu einer Gemeinschaft.

Eine umfassendere Verarbeitung der Daten wäre ohne die beschriebenen Nachteile des Combat Log-Formats (siehe 3.4 Beschränkungen) möglich. Gerade das Fehlen von Sonderereignissen für den Beginn und das Ende einer Begegnung, zumindest einer Hauptbegegnung, erschweren die Verarbeitung. Dass bestimmte Begleiter generell nicht ihren zugehörigen Besitzern zugeordnet werden können (siehe 4.3.4 Zuordnungsproblem Begleiter), führt zu einer Verfälschung der Gesamteffizienz des Spielers, der eigentlich der Besitzer des Begleiters ist. Aber gerade die Gesamteffizienz von Spielern ist ein wichtiger Punkt für die Benutzer solcher Webservices. Die mögliche Unvollständigkeit der Ereignisse und das Fehlen einer offiziellen Dokumentation stellen weitere Einschränkungen bei dem Umgang mit Combat Logs dar.

Die Klassifikation der vier Rollen (siehe 2.2.11 Rollen) sollte, ohne Beachtung des primären Talentbaums, mit aufsummierten Wertigkeiten bestimmter Ereignisse, bzw. deren normiertes

Verhältnis zu den entsprechenden Ereignissen des ganzen Teams, als Features möglich sein. Im Rahmen dieser Arbeit wurde dies als Teilansatz gelöst, um die Zuordnung des primären Talentbaums „Wilder Kampf“ der Klasse „Druide“ zu einer Rolle zu realisieren (siehe 4.3.3 Zuordnungsproblem Rolle). Desweiteren wurde mit diesen normierten Summen für die Ereignisse Schaden(ausgehend), Schaden(eingehend) und Heilung eine Zuteilung zu drei Rollen getestet, hierfür wurden die Rollen Schadensaussteiler(nah) und Schadensaussteiler(fern) zusammengefasst. Durch Festlegen von Grenzwerten zu den einzelnen normierten Summen und einer besonderen Reihenfolge bei der Überprüfung, wobei die erste Überschreitung eines Grenzwertes zu einer vorzeitigen Zuordnung dieser Rolle führte, wurden schon vielversprechende Ergebnisse für diesen Ansatz erzielt. Allerdings kann es bei diesem Ansatz zu Problemen bei der Zuordnung kommen, wenn ein Spieler mit schwach ausgeprägten Werten Teil eines Teams mit ansonst stark ausgeprägten Werten ist, und die normierten Summen des Spielers unter den Grenzwerten liegen.

Die Klassifikation der 30 primären Talentbäume (siehe 2.2.10 Talente) oder 10 Klassen (siehe 2.2.6 Klassen) sollte, ohne Beachtung der Signaturfähigkeiten, mit Ereignissen zu Zustandsveränderungen und dem Einsatz von Fähigkeiten als Features möglich sein. Allerdings ist die bloße Anzahl der verschiedenen interessanten Ereignisse so groß, dass sich eine Vorselektion der Features anbietet, anhand derer dann eine Klassifikation vorgenommen werden kann. Zur Beurteilung der Features können die Informationen aus der Datenbanktabelle *static\_spell\_classtree* verwendet werden. Die Ergebnisse können mit den Daten dieses Webservices evaluiert werden.

Da der primäre Talentbaum, und somit auch meistens die Rolle, zwischen Begegnungen gewechselt werden kann, macht eine Klassifikation nur innerhalb der Schranken von Begegnungen Sinn. Die Klasse hingegen kann nicht gewechselt werden, und deshalb mit den Daten des gesamten Combat Logs klassifiziert werden.

Bei der Analyse von einzelnen Spielern sind nur Ereignisse interessant, in denen der Spieler Quelle oder Ziel ist. Auf diesen eingeschränkten Daten, mit eventueller zusätzlicher Einschränkung durch Begegnungen, können Untersuchungen zum Verhalten und eventuell der Effizienz des Spielers gemacht werden. Hierbei sollten interessante Ergebnisse Informationen über den Einsatz von Fähigkeiten beinhalten. Der Einsatz von zwei verschiedenen Fähigkeiten nacheinander, der wiederholte Einsatz derselben Fähigkeit und der Einsatz einer Fähigkeit während einer bestimmten Zustandsveränderung sollten hierbei besonders interessant sein. Bedingt durch Abklingzeiten von Fähigkeiten (dies verhindert den schnellen wiederholten Einsatz hintereinander), sollten auch jeweils die zeitlichen Abstände berücksichtigt werden. In [16] wird ein vielversprechender Ansatz für solch eine Untersuchung vorgestellt, da auch eine Berücksichtigung der zeitlichen Abstände möglich ist. Zusätzlich sind räumliche Informationen in diesem Ansatz optional, was zu den nicht vorhandenen räumlichen Informationen im Combat Log passt.

Ein wichtiger Aspekt für Teams ist nicht nur die Effizienz in Begegnungen, sondern auch die Vorbereitung einer Wiederholung einer Begegnung, nach dem das Team eine Niederlage hatte. Nach einer Niederlage sollte das Team folgende Phasen durchlaufen: Bewegen zum Ausgangspunkt der nächsten Begegnung, positive Zustandsveränderungen wirken und alle Ressourcen füllen, Begegnung starten. Da die Bewegung der Spieler nicht in der Combat Log-Datei erfasst ist, kann logischerweise auch keine Auswertung hierzu direkt erfolgen, allerdings kann an der Spanne zwischen erfolglosem Versuch und Beginn der positiven Zustandsveränderungen erkannt werden, wie lange die Dauer der Bewegung war. Allerdings

ist es auch hierfür nicht möglich einen pauschalen Grenzwert anzugeben, da abhängig von der nächsten Begegnung die Dauer der nötigen Bewegung unterschiedlich sein kann. Die Phase der positiven Zustandsveränderungen sollte nur die Zeit erfassen, in der auch alle Spieler davon betroffen sind. Umgekehrt formuliert bedeutet das, erst wenn alle Spieler vor Ort sind, sind auch alle Spieler gleichzeitig von den positiven Zustandsveränderungen betroffen. Vorzeitige einzelne positive Zustandsveränderungen sollten den Beginn dieser Phase nicht auslösen. Interessant wäre ein Versuch diese Phasen durch Data Mining erkennen zu können. Hierfür scheint sich ein rückwärts gerichteter Ansatz zu empfehlen.

Begegnungen können nach weiteren Kriterien unterschieden werden. Sie können jeweils für zwei Teamgrößen (10 und 25) und zwei Schwierigkeitsstufen ausgelegt sein. Die Teamgröße hat nur Einfluss auf die Eigenschaften des Hauptgegners, die aber nicht in den Daten des Combat Logs stehen. Hinzu kommt, dass auch 10 oder weniger Spieler Begegnungen für die Teamgröße 25 bestreiten können. Umgekehrt geht das allerdings nicht. Die Schwierigkeitsstufe hat ebenfalls Einfluss auf die Eigenschaften des Hauptgegners, zusätzlich sollten sich aber auch die Identifikationsnummern der eingesetzten Fähigkeiten des Hauptgegners ändern. Ansätze mit Data Mining-Methoden zur Einteilung von Begegnungen nach diesen Kriterien könnten sehr interessant sein.

Aus zeitlichen Gründen wurde darauf verzichtet eine Web-Seite für einzelne Fähigkeiten zu entwickeln. Dadurch wäre es möglich den Einsatz einer Fähigkeit zwischen mehreren Charakteren zu vergleichen, was allerdings nur bei identischem primärem Talentbaum aussagekräftig ist. Auch ein Vergleich desselben Charakters bei unterschiedlichen Begegnungen wäre sehr sinnvoll. Dabei auftretende Unterschiede wären ein guter Ansatz zur Bestimmung von Kriterien bei der Unterscheidung der Effizienz eines Charakters.

Die Ergebnisse der Analyse von Hauptgegnern können zur Hilfestellung bei der jeweiligen Begegnung genutzt werden. Dies ist über Hilfsprogramme möglich, die in World of Warcraft® verwendet werden können.

Mit dem in [17] vorgestellten Ansatz lassen sich aktive Gemeinschaften bestimmen, wobei nach eigenen Angaben erfolgreiche Gemeinschaften in bestimmten Gebieten angesiedelt sind.

Zusätzlich lassen sich erfolgreiche Gemeinschaften mittels [18] ermitteln, die in der Regel aus guten Spielern besteht. Combat Logs von diesen Gemeinschaften sollten sich als Quelle für Daten guter Spieler erweisen.

Ausgehend von dem Webservice dieser Arbeit, sind Verbesserungen im administrativen Bereich und bei der Vergleichbarkeit von mehreren Combat Logs denkbar. Gleichzeitig sind Alleinstellungsmerkmale schon vorhanden, die gerade im Bereich der zeitlichen Manipulation noch weiter ausgebaut werden können. Zusätzlich ist die Kompaktheit der visuellen Präsentation, bedingt durch die Menge an Informationen, noch nicht optimal gelöst.



## Anhang A

Dies ist eine Auflistung der bekannten failedTypes:

"A more powerful spell is already active"  
"Another action is in progress"  
"Can only use outside"  
"Can only use while swimming"  
"Can't do that while asleep"  
"Can't do that while charmed"  
"Can't do that while confused"  
"Can't do that while fleeing"  
"Can't do that while horrified"  
"Can't do that while incapacitated"  
"Can't do that while moving"  
"Can't do that while silenced"  
"Can't do that while stunned"  
"Interrupted"  
"Invalid target"  
"Item is not ready yet"  
"Must be in Bear Form, Dire Bear Form"  
"Must have a Ranged Weapon equipped"  
"No path available"  
"No target"  
"Not enough energy"  
"Not enough mana"  
"Not enough rage"  
"Not yet recovered"  
"Nothing to dispel"  
"Out of range"  
"Target is friendly"  
"Target is hostile"  
"Target needs to be in front of you."  
"Target not in line of sight"  
"Target too close"  
"You are dead"  
"You are in combat"  
"You are in shapeshift form"  
"You are unable to move"  
"You can't do that yet"  
"You must be behind your target."  
"Your target is dead"

## Anhang B

Benutzungsmöglichkeit von Identifikationsnummern bei externen Diensten

Fähigkeiten (wobei nach s= die Identifikationsnummer (i.d.R spellId) angegeben werden kann):

<http://wowdata.buffed.de/?s=49143>

Nicht-Spieler-Charakter (wobei nach n= die Identifikationsnummer (i.d.R GUID, wenn kein Spieler) angegeben werden kann):

<http://wowdata.buffed.de/?n=45992>

## Anhang C

Screenshot der Web-Seite encounter.php

Startzeitpunkt	Name	#V	Erfolg	Dauer	#S	SF	H	SAN	SAF	D	H	K	J	M	Pa	Pr	Sa	Su	T
04-03 19:32:47.234	<a href="#">Theralon</a>	1	WAHR	05:27.422	10	1	3	3	3	2	0	1	0	1	2	1	2	0	1
04-03 20:07:25.671	<a href="#">Cho'gall</a>	1	WAHR	09:31.063	10	2	3	2	3	2	0	1	0	1	2	1	2	0	1
04-03 21:00:45.875	<a href="#">A'l'Akir</a>	1	FALSCH	01:24.593	10	1	3	3	3	2	0	1	0	1	2	1	2	0	1
04-03 21:04:24.421	<a href="#">A'l'Akir</a>	2	FALSCH	01:24.219	10	1	3	3	3	2	0	1	0	1	2	1	2	0	1
04-03 21:07:54.203	<a href="#">A'l'Akir</a>	3	FALSCH	06:06.468	10	1	3	3	3	2	0	1	0	1	2	1	2	0	1
04-03 21:16:48.125	<a href="#">A'l'Akir</a>	4	FALSCH	04:35.968	10	1	3	3	3	2	0	1	0	1	2	1	2	0	1
04-03 21:24:14.203	<a href="#">A'l'Akir</a>	5	FALSCH	05:53.140	10	1	3	3	3	2	0	1	0	1	2	1	2	0	1
04-03 21:34:42.453	<a href="#">A'l'Akir</a>	6	FALSCH	01:25.312	10	1	3	3	3	2	0	1	0	1	2	1	2	0	1
04-03 21:38:56.546	<a href="#">A'l'Akir</a>	7	FALSCH	05:05.907	10	1	3	3	3	2	0	1	0	1	2	1	2	0	1
04-03 21:48:45.625	<a href="#">A'l'Akir</a>	8	FALSCH	01:09.734	10	1	3	3	3	2	0	1	0	1	2	1	2	0	1
04-03 21:52:29.953	<a href="#">A'l'Akir</a>	9	FALSCH	05:11.031	10	1	3	3	3	2	0	1	0	1	2	1	2	0	1
04-03 22:00:13.593	<a href="#">A'l'Akir</a>	10	FALSCH	05:38.313	10	1	3	3	3	2	0	1	0	1	2	1	2	0	1
04-03 22:10:49.062	<a href="#">A'l'Akir</a>	11	FALSCH	05:01.859	10	1	3	3	3	2	0	1	0	1	2	1	2	0	1
04-03 22:19:22.468	<a href="#">A'l'Akir</a>	12	WAHR	05:48.844	10	1	3	3	3	2	0	1	0	1	2	1	2	0	1
										<b>Rolle</b>		<b>Klasse</b>							

## Anhang D

Screenshot einer Heilmatrix

Legende: In den Zeilen der Tabelle steht die prozentuale Verteilung der Heilung des Spieler an seinen Zielen.

	Taran	Zondra	Mooby	Tricia	Vierxdrei	Mohiam	Hilflos	Entrecote	Lismera	Zynist
Taran	100.00%									
Zondra	6.46%	12.85%	24.92%	5.24%	6.74%	9.81%	9.71%	8.17%	9.62%	6.47%
Mooby			100.00%							
Tricia	0.32%	17.03%		37.51%	0.73%		15.69%	12.70%	15.70%	0.31%
Vierxdrei					100.00%					
Mohiam						100.00%				
Hilflos	10.19%	6.82%	32.10%	3.23%	6.82%	4.40%	22.70%	5.91%	5.67%	2.16%
Lismera	7.40%	3.94%	34.96%	3.11%	7.96%	6.99%	3.09%	2.68%	23.68%	6.19%

## Anhang D

Screenshot der Web-Seite overview.php

Name	Talentaum	Rolle	Aktivität	Schaden(uns)	SPS(uns)	aSPS(uns)	Heilung	HPS	aHPS	Schaden(tein)	SPS(tein)	aSPS(tein)
<a href="#">Taran</a>	Druide - Wilder Kampf	Schadensaussteiler(nah)	98,96%	5,678,834	17,366,46	17,549,07	220,301	673,70	680,79	884,441	2,704,71	2,733,15
<a href="#">Vierckrei</a>	Krieger - Furor	Schadensaussteiler(nah)	98,32%	5,428,135	16,599,80	16,883,41	208,519	637,67	648,57	807,921	2,470,71	2,512,92
<a href="#">Tricia</a>	Priester - Schatten	Schadensaussteiler(fern)	99,66%	5,262,219	16,092,41	16,147,15	1,209,549	3,698,93	3,711,51	503,726	1,540,45	1,545,69
<a href="#">Mohiam</a>	Todesritter - Frost	Schadensaussteiler(nah)	98,32%	4,671,218	14,285,07	14,529,13	99,219	303,42	308,61	712,124	2,177,75	2,214,96
<a href="#">Zynist</a>	Magier - Feuer	Schadensaussteiler(fern)	98,55%	4,550,222	13,915,05	14,119,89	0	0,00	0,00	437,667	1,338,43	1,358,13
<a href="#">Entrecote</a>	Schamane - Elementar	Schadensaussteiler(fern)	99,59%	2,995,532	9,160,65	9,197,98	0	0,00	0,00	430,836	1,317,54	1,322,91
<a href="#">Mcooby</a>	Druide - Wilder Kampf	Schadenfänger	99,75%	2,710,537	8,289,10	8,310,14	274,244	838,67	840,80	2,644,236	8,086,35	8,106,87
<a href="#">Zondra</a>	Schamane - Wiederherstellung	Heiler	99,93%	355,404	1,086,86	1,087,59	3,847,778	11,766,91	11,774,78	866,617	2,650,20	2,651,98
<a href="#">Lismera</a>	Paladin - Heilig	Heiler	99,78%	83,973	256,80	257,36	3,081,496	9,423,54	9,444,27	953,393	2,915,57	2,921,99
<a href="#">Hilios</a>	Paladin - Heilig	Heiler	99,78%	72,877	222,87	223,36	3,709,272	11,343,34	11,368,30	724,336	2,215,09	2,219,97

# Anhang F

Screenshot der Web-Seite player.php

Legende zum Balkendiagramm: Rot = verfehlt, Orange = gestreift, Blau = geblockt, Grün = kritisch, Grau = krachend, Gelb = normale Treffer

Gr	Name	Schule	#Schaden	ØSchaden	asPS	%	#E	#Verfehlt	#Gestreift	#Geblockt	#Kritisch	#Krachend	#Treffer	Balken	#Widerst	#Absorb	Verf	Block	Widers	Absorb	Info (ID)
<input type="checkbox"/>	Nahkampf	Physisch	906,325	2,518	2.815,36	19,40%	360	87(24,17%)	88(24,44%)	4(1,11%)	42(11,67%)	0(0,00%)	139(38,61%)		0(0,00%)	0(0,00%)	0	3.900	0	0	0
<input type="checkbox"/>	Auslöschern	Physisch	801,555	17,425	2.489,90	17,16%	46	1(2,17%)	0(0,00%)	0(0,00%)	10(21,74%)	0(0,00%)	35(76,09%)		0(0,00%)	0(0,00%)	0	0	0	0	49020
<input type="checkbox"/>	Froststoß	Frost	667,265	11,505	2.072,75	14,28%	58	1(1,72%)	0(0,00%)	0(0,00%)	24(41,58%)	0(0,00%)	33(56,90%)		0(0,00%)	0(0,00%)	0	0	0	0	49143
<input type="checkbox"/>	Auslöschern (Schuldhaut)	Physisch	524,554	11,403	1.629,44	11,23%	46	0(0,00%)	0(0,00%)	0(0,00%)	12(26,09%)	0(0,00%)	34(73,91%)		0(0,00%)	0(0,00%)	0	0	0	0	66198
<input type="checkbox"/>	Froststoß (Schuldhaut)	Frost	413,869	7,136	1.285,62	8,86%	58	0(0,00%)	0(0,00%)	0(0,00%)	21(36,21%)	0(0,00%)	37(63,79%)		0(0,00%)	0(0,00%)	0	0	0	0	66196
<input type="checkbox"/>	Heulende Boe	Frost	329,237	15,678	1.022,72	7,05%	21	0(0,00%)	0(0,00%)	0(0,00%)	3(14,29%)	0(0,00%)	18(85,71%)		0(0,00%)	0(0,00%)	0	0	0	0	49184
<input type="checkbox"/>	Frostfieber	Frost	283,909	2,629	881,92	6,08%	108	0(0,00%)	0(0,00%)	0(0,00%)	18(16,67%)	0(0,00%)	90(83,33%)		0(0,00%)	0(0,00%)	0	0	0	0	55095
<input type="checkbox"/>	Blutseuche	Schatten	174,771	1,802	542,90	3,74%	97	0(0,00%)	0(0,00%)	0(0,00%)	9(9,28%)	0(0,00%)	88(90,72%)		0(0,00%)	0(0,00%)	0	0	0	0	55078
<input type="checkbox"/>	Blitzenschlag	Natur	126,737	7,455	393,69	2,71%	17	0(0,00%)	0(0,00%)	0(0,00%)	5(29,41%)	0(0,00%)	12(70,59%)		0(0,00%)	0(0,00%)	0	0	0	0	89087
<input type="checkbox"/>	Blutstoß	Physisch	125,984	5,727	391,35	2,70%	22	1(4,55%)	0(0,00%)	0(0,00%)	5(22,73%)	0(0,00%)	16(72,73%)		0(0,00%)	0(0,00%)	0	0	0	0	45902
<input type="checkbox"/>	Blutstoß (Schuldhaut)	Physisch	69,378	3,154	215,51	1,43%	22	0(0,00%)	0(0,00%)	0(0,00%)	1(4,55%)	0(0,00%)	21(95,45%)		0(0,00%)	0(0,00%)	0	0	0	0	66215
<input type="checkbox"/>	Seuchenstoß	Physisch	28,789	5,738	89,43	0,62%	5	0(0,00%)	0(0,00%)	0(0,00%)	2(40,00%)	0(0,00%)	3(60,00%)		0(0,00%)	0(0,00%)	0	0	0	0	45462
<input type="checkbox"/>	Seuchenstoß (Schuldhaut)	Physisch	12,988	2,598	40,35	0,28%	5	0(0,00%)	0(0,00%)	0(0,00%)	0(0,00%)	0(0,00%)	5(100,00%)		0(0,00%)	0(0,00%)	0	0	0	0	66216
<input type="checkbox"/>	Schneidendes Eis	Frost	9,382	34	29,14	0,20%	277	0(0,00%)	0(0,00%)	0(0,00%)	0(0,00%)	0(0,00%)	277(100,00%)		0(0,00%)	0(0,00%)	0	0	0	0	50401
<input type="checkbox"/>	Frostverletzlichkeit	Frost	0	0	0,00	0,00%	3	3(100,00%)	0(0,00%)	0(0,00%)	0(0,00%)	0(0,00%)	0(0,00%)		0(0,00%)	0(0,00%)	0	0	0	0	51714
<input type="checkbox"/>	Frostbeulen	Frost	0	0	0,00	0,00%	25	25(100,00%)	0(0,00%)	0(0,00%)	0(0,00%)	0(0,00%)	0(0,00%)		0(0,00%)	0(0,00%)	0	0	0	0	50435

Graph zeichnen



## Literaturverzeichnis

- [1] Homepage von World of Warcraft  
<http://eu.battle.net/wow/de/>  
Zuletzt aufgerufen am 13.07.2011
- [2] Homepage von Blizzard Entertainment  
<http://eu.blizzard.com/de-de/>  
Zuletzt aufgerufen am 13.07.2011
- [3] Forum von Elitist Jerks  
<http://elitistjerks.com/forums.php>  
Zuletzt aufgerufen am 13.07.2011
- [4] Buffed - Klassentreffen  
<http://wowdata.buffed.de/class/portal>  
Zuletzt aufgerufen am 13.07.2011
- [5] Homepage von WoWWiki  
<http://www.wowwiki.com/Portal:Main>  
Zuletzt aufgerufen am 13.07.2011
- [6] World of Warcraft AddOn Kit  
[http://us.blizzard.com/support/article.xml?locale=en\\_US&articleId=21466](http://us.blizzard.com/support/article.xml?locale=en_US&articleId=21466)  
Zuletzt aufgerufen am 13.07.2011
- [7] WoWWiki – API COMBAT LOG EVENT  
[http://www.wowwiki.com/API\\_COMBAT\\_LOG\\_EVENT](http://www.wowwiki.com/API_COMBAT_LOG_EVENT)  
Zuletzt aufgerufen am 13.07.2011
- [8] WoWWiki – API UnitGUID  
[http://www.wowwiki.com/API\\_UnitGUID](http://www.wowwiki.com/API_UnitGUID)  
Zuletzt aufgerufen am 13.07.2011
- [9] WoWWiki - UnitFlag  
<http://www.wowwiki.com/UnitFlag>  
Zuletzt aufgerufen am 13.07.2011
- [10] buffed.de – Portal für Online-Spiele  
<http://www.buffed.de/>  
Zuletzt aufgerufen am 13.07.2011
- [11] Raidlogs – WoW combat log analyser  
<http://www.raidlogs.com/>  
Zuletzt aufgerufen am 13.07.2011
- [12] World of Logs – Real Time Raid Analysis  
<http://www.worldoflogs.com/>  
Zuletzt aufgerufen am 13.07.2011
- [13] WoW Meter Online –Combatlog Replay  
<http://www.wowmeteronline.com/>  
Zuletzt aufgerufen am 13.07.2011
- [14] sortable: Make all your tables sortable  
<http://www.kryogenix.org/code/browser/sortable/>  
Zuletzt aufgerufen am 13.07.2011

- [15] WoWWiki – Attack table  
[http://www.wowwiki.com/Attack\\_table](http://www.wowwiki.com/Attack_table)  
Zuletzt aufgerufen am 13.07.2011
- [16] S. Breining, H.-P. Kriegel, M.Schubert, A.Züfle  
„Action Sequence Mining“  
*Workshop on Machine Learning and Data Mining in Games*, 2011
- [17] C. Thureau, K. Kersting, C. Bauckhage.  
„Convex Non-Negative Matrix Factorization in the Wild“  
*Proc. IEEE Int. Conf. on Data Mining*, 2009.
- [18] WoW Guild Rankings :: WoWProgress – World of Warcraft Rankings and History  
<http://www.wowprogress.com/>  
Zuletzt aufgerufen am 13.07.2011

## Abbildungsverzeichnis

Abbildung 1: Miniaturübersicht des Webservices _____	5
Abbildung 2: Fähigkeiten allgemein _____	9
Abbildung 3: Fähigkeiten gesamt _____	11
Abbildung 4: Use-Case der Rollen _____	12
Abbildung 5: Übersicht Parsen _____	16
Abbildung 6: Übersicht Verarbeitung _____	30
Abbildung 7: Verknüpfungen von den Datenbanktabellen event, player und spell _____	30
Abbildung 8: Verknüpfungen der Datenbanktabelle encounter_rel _____	36
Abbildung 9: Verknüpfungen der Datenbanktabelle role_rel _____	38
Abbildung 10: Verknüpfungen der Datenbanktabelle pet_rel _____	40
Abbildung 11: Gesamtübersicht der Datenbankverknüpfungen _____	41
Abbildung 12: Übersicht Präsentation _____	42
Abbildung 13: Verknüpfung der Web-Seiten _____	44
Abbildung 14: Beispiel für einen Graphen _____	52



## Tabellenverzeichnis

Tabelle 1: Rollen der einzelnen Klassen nach primärem Talentbaum	14
Tabelle 2: Umschreibungen von spezifischen Ausdrücken	15
Tabelle 3: Basisparameter	18
Tabelle 4: GUID-Arten	18
Tabelle 5: Zeichen	19
Tabelle 6: UnitFlag	20
Tabelle 7: Präfixe und Parameter	21
Tabelle 9: Environmental-Type	22
Tabelle 8: School-Type	22
Tabelle 10: Suffixe und Parameter	23
Tabelle 11: Miss-Type	26
Tabelle 12: Power-Type	27
Tabelle 13: Aura	27
Tabelle 14: Datenbanktabelle event	31
Tabelle 15: Datenbanktabelle player	32
Tabelle 16: Datenbanktabelle spell	32
Tabelle 17: Datenbanktabelle static_encounter	33
Tabelle 18: Beispiele für static_encounter	33
Tabelle 19: Datenbanktabelle static_spell_classtree	34
Tabelle 20: Beispiele für static_spell_classtree	34
Tabelle 21: Datenbanktabelle encounter_rel	35
Tabelle 22: Datenbanktabelle class_rel	36
Tabelle 23: Datenbanktabelle role_rel	38
Tabelle 24: Datenbanktabelle pet_rel	39
Tabelle 25: Übersicht bestehender Webservices zur Auswertung von Combat Logs	43
Tabelle 26: HTML-Tabelle Gesamtübersicht aller Begegnungen	45
Tabelle 27: HTML-Tabelle zeitlich eingeschränkte Übersicht	46
Tabelle 28: HTML-Tabelle Schaden (ausgehend)	48
Tabelle 29: HTML-Tabelle Heilung	49
Tabelle 30: HTML-Tabellen Zustandveränderungen	50
Tabelle 31: Adresssystem	53

## Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Diplomarbeit ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Björn Hirsch

Darmstadt, den 29.09.2011