LUIZ BERNARDO MARTINS KUMMER

A Method to Predict Risk Situations on Digital Game Usage Lifecycle

Master dissertation presented to the Graduate Program in Computer Science at the *Pontifícia Universidade Católica do Paraná* as a partial fulfillment of the requirements for the degree of Master in Computer Science.

CURITIBA

2017

LUIZ BERNARDO MARTINS KUMMER

A Method to Predict Risk Situations on Game Usage Lifecycle

Master dissertation presented to the Graduate Program in Computer Science at the *Pontifícia Universidade Católica do Paraná* as a partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Prof. Dr. Emerson Cabrera Paraiso

Co-advisor: Prof. Dr. Julio Cesar Nievola

CURITIBA

2017

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central

Kummer, Luiz Bernardo Martins

K96m A method to predict risk situations on game usage lifecycle / Luiz Bernardo
2017 Martins Kummer ; advisor: Emerson Cabrera Paraiso ; co-advisor: Julio Cesar Nievola. – 2017.

84 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2017

Bibliografia: f. 80-84

1. Jogos eletrônicos. 2. Mineração de dados. I. Paraiso, Emerson Cabrera. II. Nievola, Julio Cesar. III. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. IV. Título.

CDD 22. ed. – 794.8

Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática
PUCPR
GRUPO MARISTA

ATA DE SESSÃO PÚBLICA

DEFESA DE DISSERTAÇÃO DE MESTRADO Nº 01/2017

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA – PPGIa
PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ - PUCPR

Em sessão pública realizada às 08h30 de 03 de Março de 2017, no Sala 305 – Escola de Negócios, ocorreu a defesa da dissertação de mestrado intitulada **"A Method to Predict Risk Situations on Digital Game Usage Lifecycle"** apresentada pelo aluno **Luiz Bernardo M. Kummer**, como requisito parcial para a obtenção do título de **Mestre em Informática**, na área de concentração **Ciência da Computação,** perante a banca examinadora composta pelos seguintes membros:

Prof. Dr. Emerson Cabrera Paraiso (Orientador)- PUCPR

Prof. Dr. Julio C. Nievola (co-orientador) – PUCPR

Prof. Dr. Carlos Silla Junior – PUCPR

Prof. Dr. Marcos Didonet Del Fabro - UFPR

Após a apresentação da dissertação pelo aluno e correspondente arguição, a banca examinadora emitiu o seguinte parecer sobre a tese:

Membro	Parecer	
Prof. Dr. Emerson Cabrera Paraiso	(X) Aprovada	( ) Reprovada
Prof. Dr. Julio C. Nievola	(X) Aprovada	( ) Reprovada
Prof. Dr. Carlos Silla Junior	(x) Aprovada	( ) Reprovada
Prof. Dr. Marcos Didonet Del Fabro	(X) Aprovada	( ) Reprovada

Portanto, conforme as normas regimentais do PPGIa e da PUCPR, a tese foi considerada:

(X) **APROVADO**
(aprovação condicionada ao atendimento integral das correções e melhorias recomendadas pela banca examinadora, conforme anexo, dentro do prazo regimental)

( ) **REPROVADO**

E, para constar, lavrou-se a presente ata que vai assinada por todos os membros da banca examinadora.
Curitiba, 03 de Março de 2017.

Prof. Dr. Emerson Cabrera Paraiso

Prof. Dr. Julio C. Nievola

Carlos N. Silla Jr.

Prof. Dr. Carlos Silla Junior

Prof. Dr. Marcos Didonet Del Fabro

Programa de Pós-Graduação em Informática
PPGIa PUCPR

Table of Contents

List of Figures

List of Tables

List of Abbreviations

SLR	*Systematic Literature Review*
WOWAH	*World of Warcraft Avatar History Dataset*
DAU	*Daily active users*
MAU	*Monthly active users*
UT	*Utilization Time*
DB	*Database*
ID	*Identifier*

Abstract

Digital games are usually software-profit, and its special feature is the ability to offer its users the immersion. Immersion is the pleasure provided by a media experience, being a sensual ability to transport the participant into a simulated and illusory reality, that is a degree that sets and measure how much the player is bonded and immersed in the game. In this work, we advocate that immersion can be represented by the commitment of a player to a game. After doing a systematic literature review, we could identify that game companies are very concerned with getting new players and keeping, as long as possible, the current players because more players means more money. Keeping the active player base is mainly done in two ways: releasing new versions of the same game with some improvements, or launching a new game. This decision is made by analyzing the usage lifecycle of the game, which can be defined by metrics such as DAU (daily active users) or MAU (monthly active users). However, decisions are mainly based on empirical experience than on the basis of an indicator. Some researches in the game domain studied the feeling of the player over time related to a game or a game genre (e.g., running, adventure, war, etc.). These studies define phases of players' interest, and some factors that motivate and demotivate to continue playing a given game. Our work proposal is to create an indicator to support decision-making on the game lifecycle, using metrics of a proposed commitment measure, which we believe is the main aspect of the usage of a game. Commitment represents a deeper analysis in the player behavior, instead of MAU that only illustrates "*how many players play*", commitment shows "*how motivate are the players*".The proposed risk indicator can show how the players are adhering in the attachment side, being it an approach of immersion. This new model would provide a methodological basis rather than an empirical one, to making decisions based on how the game is in the market, and also how to act to extend as long as possible its lifecycle and profit. The proposed method was applied in a real dataset, the final results identify risky situations where the classic approach do not.

Keywords : game usage lifecycle, player profile, player commitment, data mining, risk indicator.

Resumo

Jogos digitais são softwares, majoritariamente com fins lucrativos, tendo como característica especial a capacidade de oferecer aos seus usuários a imersão. Imersão é: o prazer que é proporcionado por uma experiência midiática, sendo uma habilidade de sensualmente transportar o participante para dentro de uma realidade simulada e ilusória. Em outras palavras, é um grau que define o quanto o jogador está aderente e imerso ao jogo. Esta imersão pode ser representada pelo comprometimento de um jogador com um jogo. Após a aplicação de uma revisão sistemática da literatura, pudemos identificar que produtoras de jogos se preocupam muito em obter novos jogadores e em manter pelo maior tempo possível os jogadores atuais, pois quanto mais jogadores, mais dinheiro. Esta manutenção da base de jogadores ativo é feita principalmente de duas formas: lançando novas versões de um mesmo jogo apresentando algumas melhorias, ou o lançamento de um jogo novo. Esta decisão é feita analisando o ciclo de vida de utilização do jogo, o qual pode ser definido por métricas como DAU (*daily active users*) ou MAU (*monthly active users*). Entretanto, as decisões são tomadas mais pela experiência empírica, do que com base em um indicador. Algumas pesquisas na área de jogos são voltadas ao fator motivacional do jogador ao longo do tempo relacionado a um jogo ou a um gênero de jogo (por exemplo: corrida, aventura, guerra, etc.). Estas pesquisas rotulam fases de interesse dos jogadores ao longo do tempo, e alguns fatores que os motivam e desmotivam a continuar jogando um jogo. Nesta pesquisa propomos a criação de um indicador de risco para o apoio a tomada de decisão sobre o ciclo de vida de jogos, tomando como base métricas de uma medida proposta chamada comprometimento, a qual acreditamos representar o principal aspecto de utilização de um jogo, pois irá mostrar o quão os jogadores estão aderentes no lado motivacional. Este novo método proporciona uma base metodológica ao invés de empírica, para a tomada de decisões das produtoras de jogos a respeito de como o jogo esta "vivo" no mercado, e também de como agir para prolongar pelo maior tempo possível o ciclo de vida e seus lucros. O método proposto foi aplicado a uma base de dados real, os resultados finais identificaram situações de risco onde a métrica clássica não identificou.

Palavras-Chave: ciclo de vida de utilização de jogos, perfil de jogadores, comprometimento de jogador e Mineração de Dados.

Chapter 1. Introduction

The usage of a digital game (only "game" hereinafter) starts with its availability in the market and ends with unprofitability or lack of use (Moore [1995], Speller [2012]). That usage can be interpreted as the lifecycle of the game. As a game is played usage data is generated (e.g., log, login list, match result). Game users (only "player" hereinafter) have a "flame" of interest, which motivates them to use a game, this interest changes over time, as shown by Zhu, Li and Zhao [2010] and Cook [2007].

There are games with other goals beyond entertainment, such as serious games, accessibility games and educational games, but in this research the focus is in entertainment games.

The target audience of a game does not choose the game to solve a problem, like in other kinds of software, but they chose a game for fun, in other words, the players are volunteers, even if they pay or not. Games are simulations with some mechanisms of risk and award, and that capacity, which a game have to attract and captivate the players is influenced by the immersion generated. Defined by Salen and Zimmerman [2003] immersion is "the pleasure provided by a media experience, being a sensually ability to transport the participant into a simulated and illusory reality". We can assimilate the degree of immersion to the degree of acceptance of a game by its players. A game content is consumed because a player like it (accepted), and that motivation in playing is kept by the immersion provided ("simulated and illusory reality"). Players look for the most degree of fun. Being the immersion a kind of sentiment, we propose an approach to measure a part of this, which is expressed in the usage, and we named it as commitment. We define commitment as "the attachment of a player to a given game", being measured by the time spent playing and the score obtained (players' ability).

Zhu, Li and Zhao [2010] interviewed players about their sentiments over time related to multiplayer on-line games (games played on the internet with many players at the same time). More details in Chapter 3.

Cook [2007] studied the game genre lifecycle. Genres are defined as group of games sharing some similar mechanism of risk and award, for example: race games, war games and adventure games. At this point we can define a game with two lifecycles, one is the usage and another is the conceptual (genre lifecycle). Cook identified some behaviors of game producers and players over the genre lifecycle, as is shown in Figure 1.

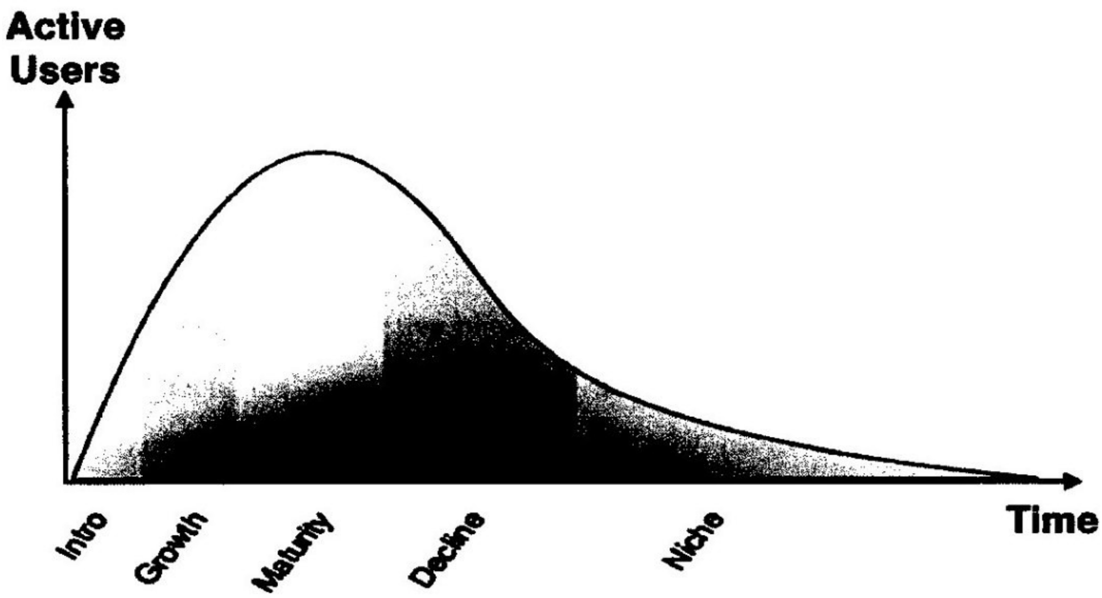

Figure 1 - Game genre lifecycle (Adapted from Cook [2007]).

Each stage was defined as:

- Introduction (*Intro*): game mechanisms are innovative and generate curiosity.
- Growth: the public accepted the genre and more games of that kind were developed.
- Maturity: great game producers adopted the genre.
- Decline: fewer games are developed over time. The genre attracts fewer players than before.
- Niche: there is no financial return, great game producers leave the genre and some games are maintained for love and not for money.

We assume that the behavior of usage lifecycle fits to the genre lifecycle, because the player discover the game or genre, fall in love, get bored and then move on to other forms of entertainment (Cook [2007]). The niche stage is the final stage and the most dangerous, because the number of active players drops constantly to levels that leads the game to become unprofitable.

Speller [2012] studied the simulation of the game usage lifecycle. He was motivated by a "good" problem, which is derived from the great success of on-line games. This success created some doubts in game producers' mind about how the usage behavior would be in the future (more details in Chapter 3).

Based on the research of Speller [2012] and on the interviews provided by Ludgate [2011], Graft [2009], Sheffield and Alexander [2008] some management actions done by game producers were identified:

- Begin of commercialization: introduction in the market and a strong appeal to get a great number of players, by advertisements. This chase for new players occurs for the entire lifecycle of the game.
- Growth: the number of active players is growing, game is more profitable. Begins an analysis about the rate of new players and the rate of abandonment. If the rate of abandonment is greater than the rate of new players this is a signal of unmet expectations, demanding the game producer to make some decisions to solve the problem. This can occur by bugs, lack of instructions about how to play the game or because what is shown on the advertisement does not occurs in the game.
- Retention: the game has the abandonment rate greater than the new players rate. That shows the game content was consumed and do not bring more fun to the players. On this stage, the game producer releases some new versions of the game aiming in disposing new contents to encourage its active players and new players. As players remain more time active, they are candidates to play and consume the new content disposed, generating a better profit.
- Decline: players do not have any more interest in the game. The game is unprofitable and no action of game producer can motivate the players again. All action made on the previous stages have the purpose of postponing the arrival of this stage.

The producer observes the historical behavior of the usage through some metrics like MAU (monthly active users). Looking at it, the producer wants to identify good or bad (risk) situations. When a risk situation is identified (e.g., the number of active players is not more profitable), the producer immediately tries to solve the situation or at least minimize it.

The availability of games in the market can be done in three main ways: on shelf, monthly payment or for free. Shelf games are sold in virtual and physical stores. In this case, the players pay for the product (game) before they can use it. Monthly payment games have the characteristic of charging for monthly usage. Games called Free are disposed in the market with no initial payment, the player plays with some restrictions. In this case, inside of the game exists a virtual market, which provides some paid services, being the source of the profit. Monthly payment games can also have this kind of virtual market. Speller [2012] showed that although free games have more players than other kinds, just 1% to 3% of players pay to play, in other words, just 1% or 3% of active players are profitable.

Shelf games have the greatest part of its profit on the beginning of commercialization in the market (weeks), as show in Figure 2. However, monthly payment and free games have their profit over the months (as show in Figure 3). In the case of monthly payment games, excluding the eventual promotions (game free for a period of time), every player pay, different of free games where 1% to 3% pay.

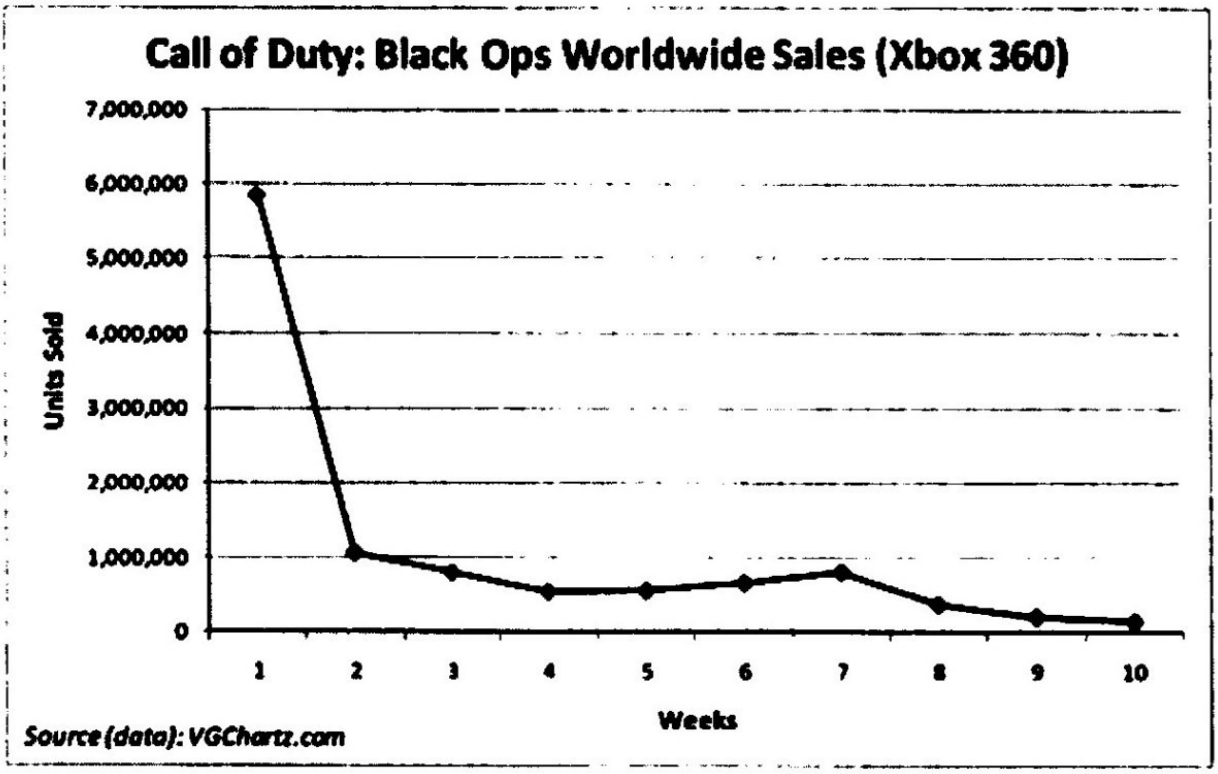

Figure 2 - Sales of a shelf game over a period of 10 weeks (Extracted from Speller [2012]).

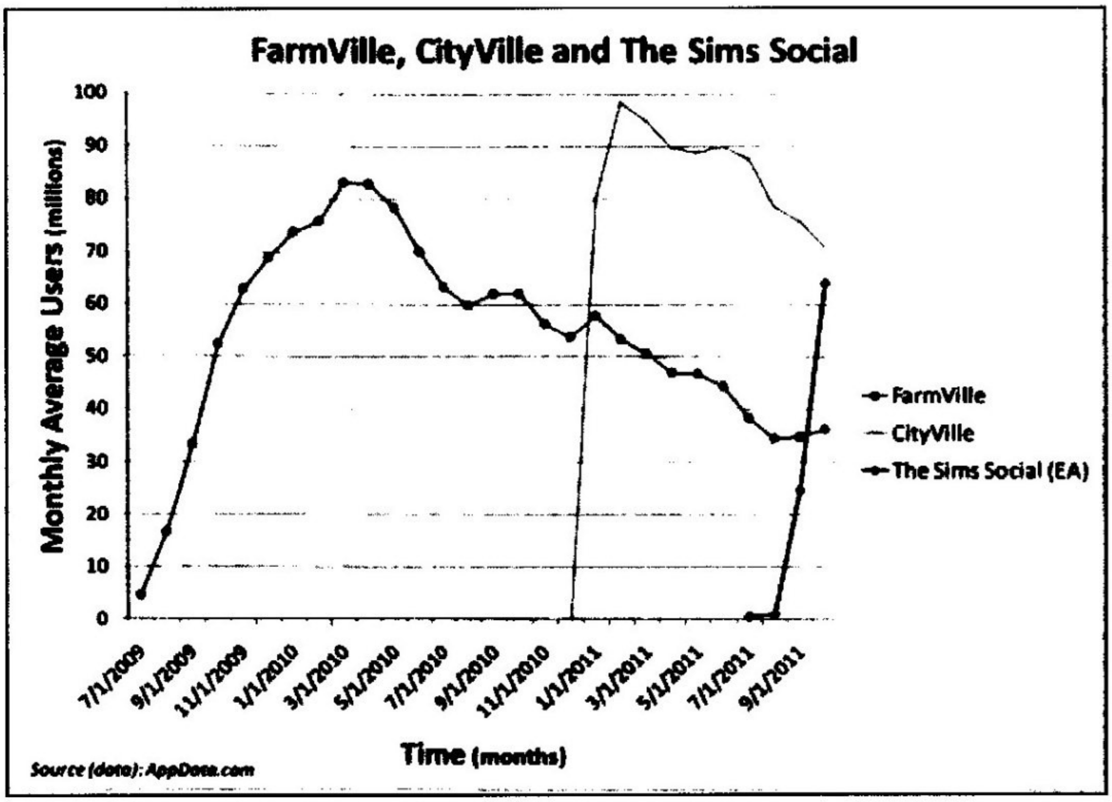

Figure 3 - Example of monthly payment and free game usage over time (Extracted from Speller [2012]).

Given the objective of game producers to engage more and more players, Stewart [2007] showed some strategies of development and test for games (more details in Chapter 3).

After the game is released to the market, the usage lifecycle begins, and that will be the object of study in this research.

1.1. Motivation

The previous paragraphs presented the influence of the management of game lifecycle in the success or failure of a game. The analysis did over the lifecycle allows the game producer to identify good or risk situations, supporting the decision-making for more effective and targeted solutions. Companies like HoneyTracks [2012] offer assistance services to help game producers with the management of the usage lifecycle. Some metrics of interest of the game producer are collected and then solutions are projected to improve the situation (e.g., improve rentability). However, to the best of our knowledge, there is no study, published in the literature, interested in the motivational usage of a game by a player related to the stage of the lifecycle a game is. Decision-making done by the game producer were based on empirical experience. A model which analyzes the game lifecycle considering the aspect of immersion (commitment in the case of this research) does not exist, and that is what we are proposing, because we advocate that motivation is an important factor to follow in the game usage lifecycle. In order to deal with the player behavior analyzes, we intend to apply Machine Learning techniques to model players based on usage data and then predict future behaviors.

1.2. Objective

The main goal of this research is to develop a new method to allow the analysis of the game usage lifecycle, disposing as a final result a risk indicator based on a proposed commitment measure. The method is based on Machine Learning to model the player behavior and label degrees of commitment based on it. To achieve this, the following steps were made:

Specific objectives:

- Cluster usage data to identify player behaviors based on the commitment context (time spent playing and score achieved).
- Obtain for each timestamp (e.g., monthly), a model that predicts its player behavior.
- Create an observation window based on models previous created to predict the player behavior as a historical point of view.
- Compute changes on the commitment over time.
- Create a Risk Indicator based on the commitment variation.

1.3. Working Hypothesis

The research hypotheses are:

H 1)

That it is possible to extract commitment data based on game usage data.

H 1.1)

That it is possible to induce a model influenced by commitment data.

H 1.1.1)

After the induction of the model, it can infer good and risk situations.

H 1.1.2)

A model with commitment metric presents a better assertiveness in identifying risk situations than MAU.

1.4. Contribution and Technology Transfer

The main contribution of this research is the proposition of a method capable of predicting risk situations, using commitment metrics.

The proposed method has potencial to help game producers in the market, because it can measure the motivational usage of its games. After an expert validation, the method can be implanted in a real situation.

1.5. Scope

This research is focused on the proposition of a risk indicator about the digital game usage lifecycle, games with usage data collect procedure and are oriented in entertainment (because of the motivational usage). We applied the method to a MMORPG (Massively multiplayer online role-playing game), but we advocate that if a game contemplates the method assumptions, the method can be applied with no restrictions based on game genres or type of availability (e.g., on shelf, monthly payment or for free).

1.6. Organization of the Text

After the introductory Chapter, a theoretical background will be detailed in Chapter 2. Posteriorly the state of art will be shown (Chapter 3), specifying how the systematic literature review was done and describing the related works. In Chapter 4 the method to predict risk situations in its conceptual form will be detailed, on the next chapter the methodological procedures are shown, specifying how Data Mining techniques are applied and evaluated. Chapter 6 have the application of the method on a game usage data and its equivalents results and analyzes. Chapter 7 finishes with final conclusions and future works to be done.

Chapter 2.
Theoretical Background

This chapter presents the background related to Games and Data Mining.

2.1. Digital Games

Cook [2007] defined a digital game as a software with mechanisms of risk and award, with the capacity of providing a "flame" of interest to its players. The dynamics of risk and award is also used in training games and educational games, even if they are not focused only in entertainment. Some games are called "Serious Games", this kind of game has focus in the development of a skill, like: fire fighting or piloting (e.g., plane, helicopter). Educational games use the aspect of award to teach academic knowledge to its players, an example can be a Mathematics game, where the player earns points because he or she calculated correctly. Another type of game is focused to attend people with disabilities, which are called Accessibility Games. The idea is to integrate people with various difficulties who cannot use entertainment games. Some events are focused on developing this kind of game, like the "Accessibility Jam"[1] to deaf, dumb and blind people.

Games which do not have focus on entertainment, and consequently, not having players with a voluntary and motivational use, cannot be used for the proposal of this research. The research focus is on the identification of how a player is adherent to a game by a voluntary and motivational way, and use that information to predict risk situations related to that, improving the actual analyzes about game usage lifecycle, not only looking for a quantity of active users (Speller, 2012), but looking at how committed are the active players, allowing a deeper analysis with the new information.

An "ordinary" software can be used by a user who wants to solve a problem. This can be for an internal company activity or for a personal use, for example, pay a bill in an internet banking. A game differs from this kind of software because it can provide entertainment through a simulation.

When we think in the motivational use of a commercial software and a game, it is possible to identify a difference. In a commercial software (e.g., internet banking) its users are induced to use it, that does not happen in a game, where its players have a voluntary usage. As happens with

[1] Web site: http://jams.gamejolt.io/accessibilityjam

books and movies, games also have characteristics which attract more of a certain kind of player than another. Players want a hobby which is funny and nice to them.

Moore [1995] represented a software lifecycle as its corresponding usage over time. In his model (presented in Figure 3) exists "The Chasm", which represents the acceptance or not of a software in the market. Degrees of maturity are also shown, being a kind of software life stages. The idea is an initial growth of use, and after a peak, a gradual decay. The causes which contribute to the decay can be, for instance, a competing software.

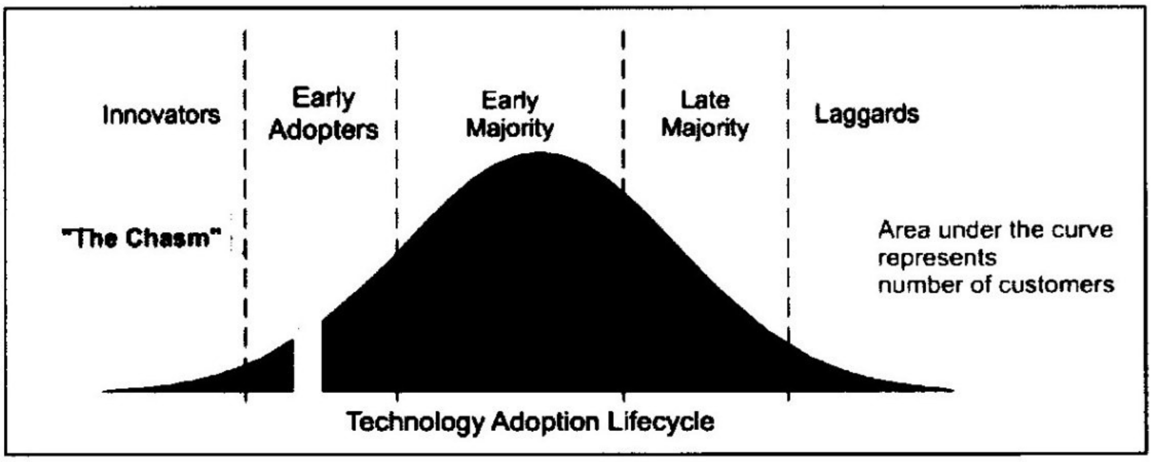

Figure 3 - Software lifecycle (Extracted from Moore [1995]).

Digital games also have a lifecycle. However, they differ from the traditional Moore's lifecycle because games have the characteristic of entertainment, which affects the game usage time. Figure 4 is an example of a game usage lifecycle.

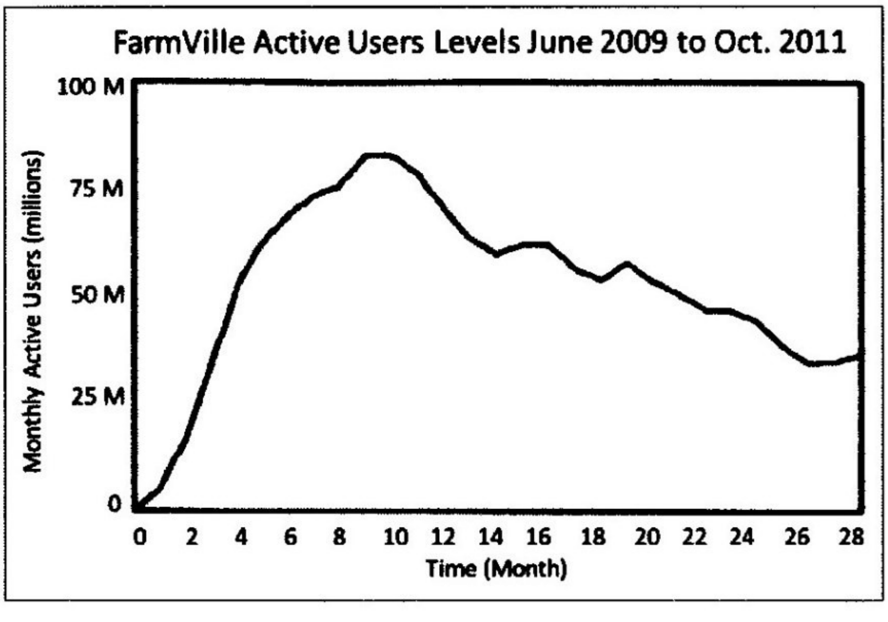

Figure 4 - Example of a game usage lifecycle (Extracted from Speller [2012]).

The analysis of Figure 4 allows to identify similarities between Moore's model and a game lifecycle. Both of them have a growth stage, a peak and after that a gradual decay.

With the goal of better understand what is the usage of a game, we define what is usage time. Usage time (UT) is the period which a game is used by a player. During the UT some events occur, which allows data collection. We have an interest in usage data, because we will use it to identify the commitment of players. We cannot measure the immersion sentiment directly, but we can measure data generated by this sentiment, being commitment an approach of the immersion.

Data which represent a usage of a game is any data which record that a player played a game at a given time. For example, a final result of a match, an on-line players list, login record or log. Some digital games do not have usage data (e.g., off-line games), making impossible to apply any aspect of this research.

The score is a game feature presented in many games. This kind of data represents the performance acquired by a player in a given timestamp. For example: the final result of a soccer match or the position in a race. Some games do not have scores, because the focus is not in acquiring greatest performances, but the focus is in the story accomplishment.

2.1.1. Risk Situations and Decision-making

A risk situation in a game is any situation which requires some actions to be made by the game producer to keep the game alive as long as possible. Some examples of risk situations are:

- Abandoning the game early, it means not passing over the Moore's Chasm (Figure 3) related to a frustration sentiment (Zhu, Li and Zhao [2010]).
- Identifying disinterest in players over the entire lifecycle, disinterest can happen due to an old content already consumed or a new content that does not please (Zhu, Li and Zhao [2010]).
- Identifying a situation where the abandonment rate is greater than the new players rate (Speller [2012]).

According to the behavior identified in the lifecycle, the game producer takes some decisions to try to solve the problem. The producers manage the usage lifecycle of a game from the beginning to the end of it. One example of a decision taken by the entire lifecycle is the "chase" for new players, which can be done by advertisements (Sheffield and Alexander [2008], Speller [2012]). An example of a decision taken after a risk situation is identified is when a game is in the beginning of the abandonment stage. In this situation, the game remains in a profitable base of active players, however, the profit is gradually reducing over time. In this situation, it is usually taken one of three possible decisions:

- Generation of new contents from the same game (e.g., more levels and challenges).
- Generation of a new game (upgrading the mechanisms of risk and award).
- Accepting the end of the game.

When the decision chosen is the generation of a new game, a situation called "self-cannibalism" may occur, as identified by Speller [2012]. In this situation, the new game consumes the players of the old one (Figure 5 shows that).

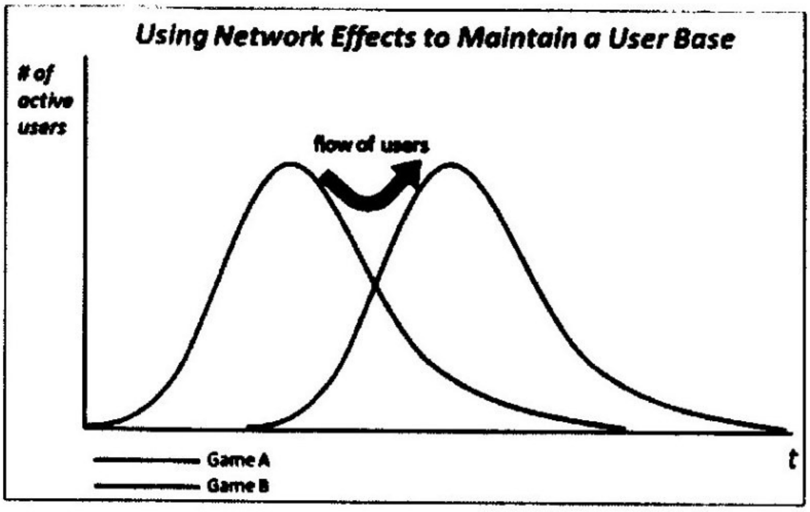

Figure 5 - Example of self-cannibalism (Extracted from Speller [2012]).

This situation is convenient for the producers, because it is possible to "ensure" somehow the usage of the new game by the players of the old one. The bad side of this idea is contributing to the end of the previous game, not keeping two profitable games at the same time.

In another way, there are some games with too much success, as showed by Speller [2012], Sheffield and Alexander [2008]. In these cases the producers want to expand the business, usually choosing to dispose their games in other platforms (e.g., PC, video-game, smartphone or tablet).

The usage of a game is linked to the entertainment it can give to the players. In other words, the quantity of game content that was not consumed yet.

2.1.2. Consumption of a Game

As a book has its contents consumed for each page read, a game has the same logic. For each level reached, challenge completed or record obtained, the game content is consumed. When a player is "satisfied" in consuming the game content, he begins to lose his motivation (Zhu, Li and Zhao [2010]).

The research of Wang and Mayer-schönberger [2010] identified that the speed consummation of a game can be associated with the degrees of consummation on the internal virtual game market. The products and services on the market were available after a payment with real money (games can have an internal money not bond to the real money), those products and services facilitate the game consume, the bad side of that is the players who pay tend to leave the game earlier than the players who do not use the internal virtual market. Then, however the game was more rentable, the UT of its players were reduced.

The speed consumption of a game can be monitored by the commitment of its players. That point is where this research focuses. We identify for each player his or her degree of commitment and based on it, some more information is extracted to induce a model to predict risk situations, resulting in a risk indicator. This indicator portrays, besides the acceptance of a game, how much the game was consumed, because the indicator represents how attached the players are.

To develop the model to predict the risk indicator, some theoretical elements must be studied and detailed, especially about knowledge discovery in databases (KDD).

2.2. KDD

KDD is a multidisciplinary field, related to Artificial Intelligence, Statistics, Machine Learning, Pattern Recognition, Database (DB) and Data Mining. The KDD motivation consists in identifying "hidden" and useful information, which is interesting for some application (e.g., sales in a supermarket or a disease study). This new information can be previously known or not.

In the course of time, studies and algorithms were developed, applied and evaluated. The process to do the knowledge discovery is usually divided in the following steps: data selection, preprocessing, transformation, data mining, evaluation and interpretation of results. More details can be found in Tan, Steinbach and Kumar [2005]. An introductory explanation about the KDD process and its main steps are described below, focusing on the aspects applied in this research.

2.2.1. Data Selection

The data selection is executed depending on each situation, there is not a rule for that. For example, a supermarket wants to study the buying behavior of its clients, to do that some sales information was collected (e.g., Sale ID, product names, values and quantities).

This step is important, because the data selected will be used for all the KDD process. If relevant information does not exist in the selected data, the final result may not be satisfactory or do not present a useful information. A study to understand what each data represent is done, and then a justification is elaborated to use or not it in the KDD process. Based on the supermarket example, if the product names are left, the final result can be useless, because no references to products exists.

Instance is a term used to the set of features which identify an individual. In the supermarket example, features can be the information about a sale (instance), as: product name, value and category (e.g., food).

2.2.2. Preprocessing and Transformation

In some situations, the original data format is not appropriate to the data mining algorithms (next step in KDD process), and because of it, is necessary to preprocess the data. Data collecting can be done in many ways, for example: manual action, sensors or a query in a DB, and that collection can present failures. Missing values (e.g., a product without a sale value) or incompatible values (e.g., an ID in a date attribute) can exist. In these situations, a preprocessing is run to fix each inconsistency.

Another situation occurs when the data must be transformed. The weather prediction is an example of that, the prediction result is not a number, but a nominal value like cold, warm or hot and the data collected is a number corresponding to the temperature. In this case, a transformation of the numeric values is done to a nominal value following a rule, like: cold when the temperate is less than 10, warm when it is between 11 and 25 and hot when it is above 25. After that, the original information is maintained in a different granularity, allowing the use of data mining algorithms. In some cases, preprocessing and transformation is not required (optional), it depends on the format of the original data.

2.2.3. Data Mining

In this step some algorithms are applied to the preprocessed and transformed data (if they were required). There are different modalities of algorithms, each one with a specific objective (linked to the called mining activity (Tan, Steinbach and Kumar [2005])). Classifiers label instances of future data, with a class learned in historical data. Regressors are very similar to classifiers, but in this case the algorithm does not label a class for an instance, instead it produces as a result a numeric value to the instance. Clusters aim to identify groups through some measures applied to the instances. Also, there exists algorithms to identify association rules, where results are generated based on the historical record, for example: if a client buy bread and milk, he tends to buy cheese too.

Independent of the algorithm modality chosen for the data mining activity, many of them have something in common (KDD). Some of them (e.g., classifiers and regressors) predict future results based on the model induced using previous data. The process to study the historical data and create a model is called Induction: the model is induced from the data. The use of that model is called Prediction: the model predicts results for future data.

Every model created represents one hypothesis about how to understand the data. For the same data many models can exist, in other words, many hypotheses (points of view). Besides that characteristic, some metrics exist to evaluate the quality of an induced model. These metrics can change according to the modality of the algorithm chosen.

In this research the following modalities were used: classification, regression, clustering and ensemble (combined use of classifiers).

2.2.3.1. Classifiers

Classifiers aim at labeling instances with a class. A class is a nominal value which already exists in the historical data used to induce the model. For example, a class could be a label that represents a person as a good or a bad payer, where good and bad are the possible classes for this problem.

There are many strategies to predict classes to the instances, being possible to divide them into two groups: a white box strategy and a black box strategy (Witten and Frank [2005]). White box algorithms allow a human to understand how the model works, illustrating how the result was generated. On the other hand, black box algorithms do not have this capacity. Depending on the problem, white box algorithms can be used to show to the interested person (usually the one who make decisions) how the result was generated, because it is common that the result represents a different approach or point of view.

Decision Trees and Rules (Tan, Steinbach and Kumar [2005]) are examples of white box knowledge representation. In Decision Trees, each leaf node is a class and each non leaf node represent a point of decision which uses the values of the instance attributes (features) to follow one of the possible options. Decision Trees can be translated to Rules and vice versa, an example of rule could be the following: If a patient has more than 39 Celsius degrees of temperature, then he or she has fever.

Neural Networks and SVM (Support Vector Machine) (Tan, Steinbach and Kumar [2005]) are examples of black box knowledge representation. The Neural Network model is created by training with many iterations in the same database, at each iteration some neurons entries have their weights changed, depending on whether the result was correct or not. SVM uses the addition of a hyperplane to divide the instances in groups, and which one represents a class of the problem. In both situations, the algorithm's internal operation is mathematically represented, being this approach more difficult to a human understand.

Regardless of the chosen algorithm modality for the classification activity, a model is induced using the historical data. This data can be divided into training (induction) and testing (prediction) sets. The classifier objective is not to predict with 100% of accuracy the historical data, but to use the historical data to induce a model able to predict the future data with the best accuracy as possible. The data behavior changes over time, therefore, it is a mistake to believe in a model with great accuracy to the historical data, because the same model can have a worse accuracy with future data, this situation is called overfitting the model. To avoid this problem, the historical data can be divided in two groups, one for training and another for testing. This decision usually is done in two main ways, a simple percentage division (e.g., 60% to train and 40% to test, also called Holdout) or cross-validation. Cross-validation divide the historical data in n parts (usually called folds), where n-1 parts are used to train and the remaining part to test. It has a defined number of iterations, depending on the number of parts, the iteration stops when all parts were tested one part at a time. After this process, the final result is calculated as the average result of all iterations.

The classifier final evaluation can be done through many metrics. In this research the percentage of correctly classified instances (accuracy) is used, because it is possible to identify the model correctness in predicting correctly the classes for the test instances.

Many classifier models can be induced to treat the same problem, but they can be different from each other (according to the different strategies adopted in the internal algorithm implementation), each one of them has a hypothesis solution to the problem and an accuracy associated. The Ensemble consists in the idea of using the result of many classifiers to label an instance. This labeling activity has some policies which define how an instance will be labeled (configurable in the algorithm). An example of this policy is the majority vote, where the more frequent class is returned as a result of the classification.

The evaluation of the ensemble is the same as applied to the classifiers. The ensemble creation process (induction stage) is adjustable, there are two ways to set the classifiers inside the ensemble. One consists in the ensemble creates new classifiers based on the data set, allowing to induce many models from different kind of classifiers (e.g., Decision Trees and Rules). Another way is the use of already induced models, being possible to combine these models to the models created in the ensemble creation process.

2.2.3.2. Regressors

A regressor is very similar to a classifier, the main difference is the result, for classifiers the result is a nominal value (class), and for regressors it is a numerical value. As occurs in Decision Trees for classifiers, the same concept is applied to the regression. Instead of a leaf node representing a class, it represents a numerical value (or an equation). The division of the historical data in train and test follows the same rules explained in the previous section.

The final evaluation of a regressor is different compared to the classifier evaluation, because the concept of correctly classified class does not exist. Instead of it, the concept is how the result accompanies the "expected" result, this characteristic can be measured by the correlation variable, which will be used in this research to evaluate the regressors.

2.2.3.3. Clustering

The concept of the clustering task is different from the classification and regression tasks (Tan, Steinbach and Kumar [2005]). Here, the historical data do not have any label which defines for each instance its corresponding group (similar to its corresponding class). The objective of the clustering task is in defining a group for each instance based on its attributes, identifying similar behaviors. There are many strategies to identify groups. In this research, we used the K nearest neighbor strategy. This algorithm is called K-means and have the capacity to set the number of groups you want to find (K value). Each instance is plotted in an n-dimensional hiper space, where *n* corresponds to the number of instance attributes. The algorithm works in the following way: some centroids are plotted randomly in the n-dimensional hiper space (the number of centroids corresponds to the number of groups to identify), each instance is associated to the nearest centroid, after that the positions of centroids are recalculated based on the average position of its instances. This process is executed until a limit of iterations is reached.

In some situations a classification approach is desired, but the data set does not have classes labeled. When it happens, the cluster task can be executed first to identify the groups, after that, the groups can be assumed as classes, allowing the classification task. As this process does not aim at predicting future behavior through the clustering approach, 100% of historical data is used in the training to find the groups. In the other hand, if the result of clustering is the final result of a given problem (do not using the cluster activity as a previous data treatment), as occurs to the previous classification example, cross-validation and some metrics can be applied to evaluate the model.

2.2.4. Evaluation and Interpretation of Results

In data mining, the data can be divided into training and test data. The training data is used to induct a model, and the test data to evaluate it. For example, classifiers uses data already labeled to induce a model, this process enables a validation which consists in checking if the predicted class is the same of the original data. This accuracy represents how assertive the classifier is. However, a classifier which predicts well on the training data (near 100%) can be bad at predicting the future data, because the data behavior changes over time. To deal with that situation, the models are induced in a way that they are not very specific to the training data. To do this the data are usually divided in the following ways: holdout and cross-validation (more details in Section 2.2.3.1).

After the data mining tasks, new information was disposed and must be analyzed to ascertain its credibility. KDD aims at discovering "hidden" information, and the interested person behind it must evaluate the new information, preferably together with a specialist. Unexpected and interesting results may appear. Accepting the result as credible, the interpretation over the results starts and some decisions to deal with the identified situation can be made (e.g., to enjoy a good situation or to minimize a bad one).

2.3. Conclusions

The players' voluntary use is influenced by the game content. That content is consumed over time, modifying the motivational usage by players (Cook [2007]) (Zhu, Li and Zhao [2010]). The motivation can be measured analyzing the time spent playing and their obtained scores, in other words, the motivation can be measured by a commitment approach. Game producers analyze the usage data to manage the game lifecycle, making decisions when favorable and unfavorable situations are identified. However, these decisions are not based on commitment (Speller [2012]). The decisions are based in the empirical knowledge instead a systematic method.

The use of KDD allows an approximation about imprecise values (e.g., players' motivational factor) by the trend discovery. Through the tasks of clustering, classification and regression, it is possible to identify distinct data behavior and form groups with it, displaying as a result, new kind of an information not previously available (e.g., quantity of players with high commitment).

Chapter 3.
State of the Art

This chapter presents a Systematic Literature Review (SLR) related to game usage lifecycle. The chapter starts by presenting the methodology used to implement this SLR. Besides the articles provided by SLR, other articles that help to understand secondary aspects of this research are described too.

3.1. Methodological Procedures

In this SLR, we searched for works on game usage lifecycle, from academic research to interviews with game producers. The review amplitude is not used on its entire completeness in this document, because some contents are not relevant to the objective of this research. The SLR was used as a knowledge base and provides the identification of some research gaps.

3.1.1. Research Protocol

The research protocol specifies how the SLR is conducted. On it is described: research objectives, research questions, keywords, search databases, publication period, search fields, inclusion and exclusion criteria of articles. This SLR was implemented in 5 main steps:

1. Definition of objectives and research questions.
2. Definition of search databases and keywords.
3. Analysis of keyword effectiveness and search databases credibility.
4. Keyword improvement and application of inclusion and exclusion criteria.
5. Reading of articles.
6. Application of inclusion and exclusion criteria.

3.1.1.1. Research Objective

The objective of this SLR is to identify models, properties and interests involved in digital game usage lifecycle.

3.1.1.2. Research Questions

RQ1

How is the lifecycle currently defined?

RQ2

What are the lifecycle stages?

RQ3

Do stages vary according to the game genre?

RQ4

What are the interests involved in the game lifecycle?

RQ5

After a game is available in the market, does some monitoring about the lifecycle exists?

RQ6

Does a measure of which stage a game is in exists?

3.1.1.3. Keywords

After the reading of some articles about the theme of this research, the following keywords were chosen: “game lifecycle”, “game life cycle stages” and “game product lifecycle”. We also added these Portuguese keywords: “ciclo de vida de jogos", “estágios do ciclo de vida de jogos” and "ciclo de utilização de jogos".

3.1.1.4. Search Databases

Initially the following search databases were used for this research: ACM Digital Library, IEEE Xplore, ScienceDirect, SpringerLink, GDC (game developer conference), SBGames (Brazilian games symposium), Google Scholar and Gamasutra (a blog about games). The databases of ACM, IEEE, ScienceDirect and SpringerLink were chosen because they have Computer Science articles. The databases of GDC, Gamasutra and SBGames were chosen because they are focused in the game field. Google Scholar was chosen as a great collector, because besides articles, it also has term papers, thesis, dissertations and registered patents.

After a qualitative analysis, GDC and SBGames were removed. GDC was removed because it does not have the same severity usually applied to academic research, not presenting peer review process and the articles which provide the information for the conference presentations, being its main focus in news about game development. SBGames was removed because its articles are already in the IEEE database. The Gamasutra is a blog in a game field and has many news, some of them are academic researches. Although it does not have the same severity applied to academic research, we kept this database as a motivational factor, because it has interviews with game producers and academic articles, which have the same focus of this research. Figure 6 illustrates the search databases used.

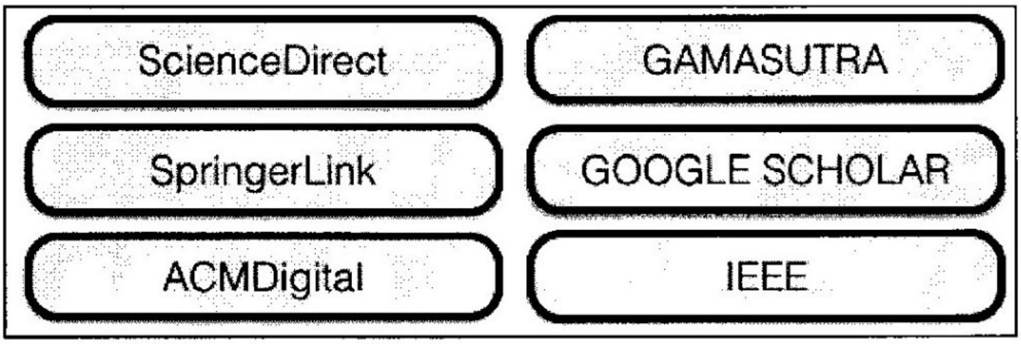

Figure 6 - Search databases used in the SLR.

3.1.1.5. Inclusion and Exclusion Criteria

The exclusion criteria are: game development cycle, articles not in Portuguese or English, and articles which do not have a link with the digital game usage lifecycle.

The inclusion criteria are: any article which adds knowledge about digital game usage lifecycle.

3.1.1.6. Search Method

There was no limit on the publication period. Search in full-text initially using the combination AND for keywords, for example, the keyword "game lifecycle" is the occurrence of the word "game" AND the word "lifecycle" in any part of the text.

After a first search we obtained a total of 6,876 articles, not counting the 35,600 registers returned in Google Scholar (these numbers for only one keyword). Analyzing a sample of articles a lack of articles focused on the theme of this research was identified, and then we changed the search method. The new strategy consists in search keywords in its completeness, for example, the article must contain the sentence "game lifecycle" in any part of the text, not more "game" in one part and "lifecycle" in another one.

The new strategy obtained 81 articles for all the six keywords and presented assertiveness about the content. That search shown us that this research theme is incipient in the academic community. It motivated us to explore more this field.

3.1.1.7. Articles Found

After applying the research protocol, the results was obtained, as show in Figure 7:

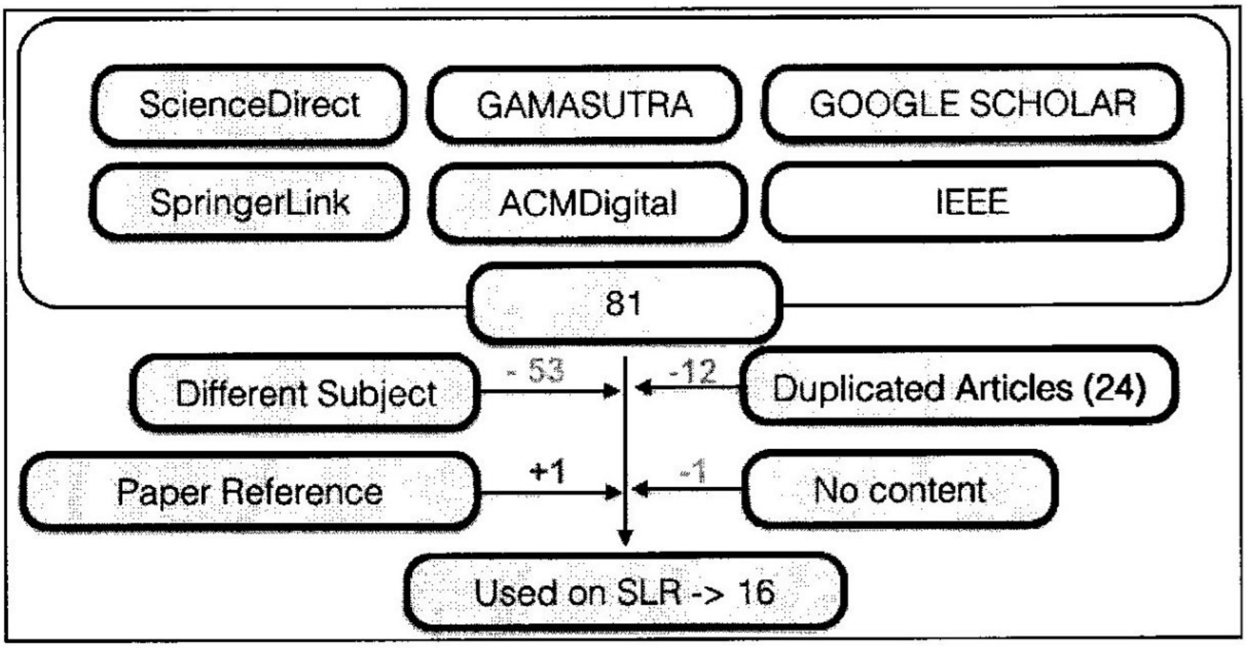

Figure 7 - Summary of articles found after apply the exclusion criteria.

The six search databases provided 81 articles, which were read and then applied the inclusion and exclusion criteria. Of the 81 articles, 53 had a content different from our focus, 12 were duplicated (caused by Google Scholar), one had no content and one was added as an interesting reference about software lifecycle (that article was used as a reference of one article returned in the search), totalizing 16 articles accepted in the SLR.

Besides the articles returned from the SLR, others articles were read as additional knowledge (including another SLR). Next the main articles related to this research are presented, some of them were detailed in the Chapter 2, as a part of theoretical background.

3.2. Usage Lifecycle

Moore [1995] described the lifecycle (usage) of softwares. A usage lifecycle starts with the availability in the market and ends when there are no users. Morre [1995] identified stages of interest as: availability in the market with new features, the acceptance or not by its users (The Chasm showed previously in Figure 3), if the software was accepted then its usage grows over time, a gradual decay of usage starts to happen and then the end with no more usage.

In the study made by Speller [2012] (Figure 8), it is possible to identify in games the same behavior of softwares: acceptance, grow of players, a gradual decay and the trend of abandonment in many different games.

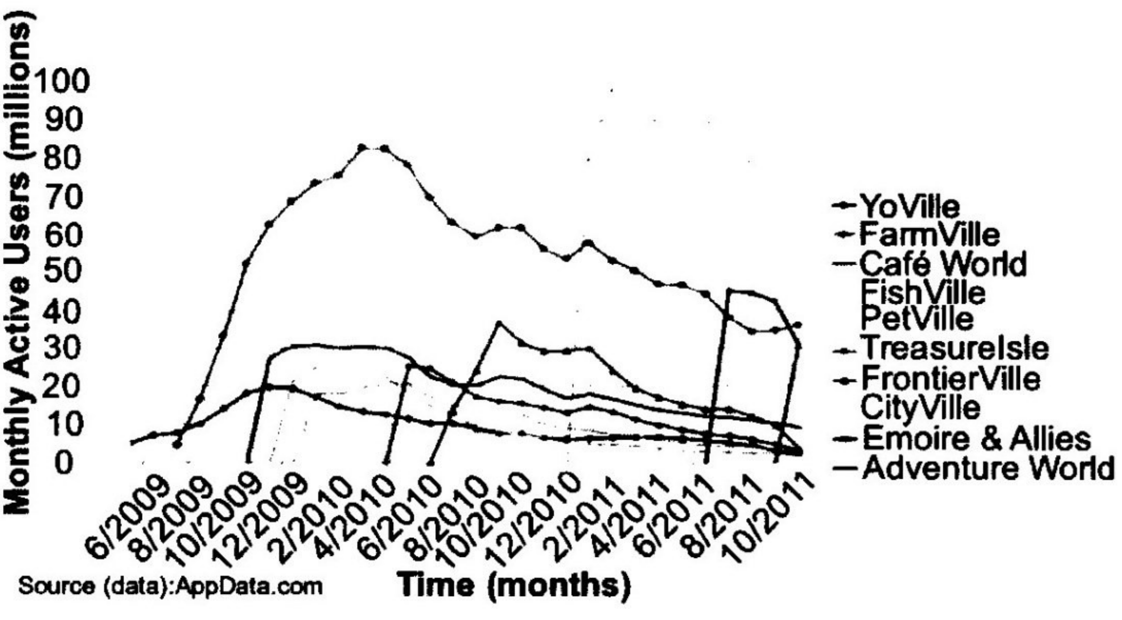

Figure 8 - Lifecycle of many games of a producer called Zynga (Extracted from Speller[2012]).

Speller [2012] was motivated by a problem found by some game producers, which have games with too much success, then the future behavior of the usage lifecycle becomes unknown. Speller identified many usage metrics such as: DAU (daily active users), MAU (monthly active users), new players rate and abandonment rate, using them to make a dynamic system capable to predict the behavior of the usage lifecycle for the next months, according to Figure 9:

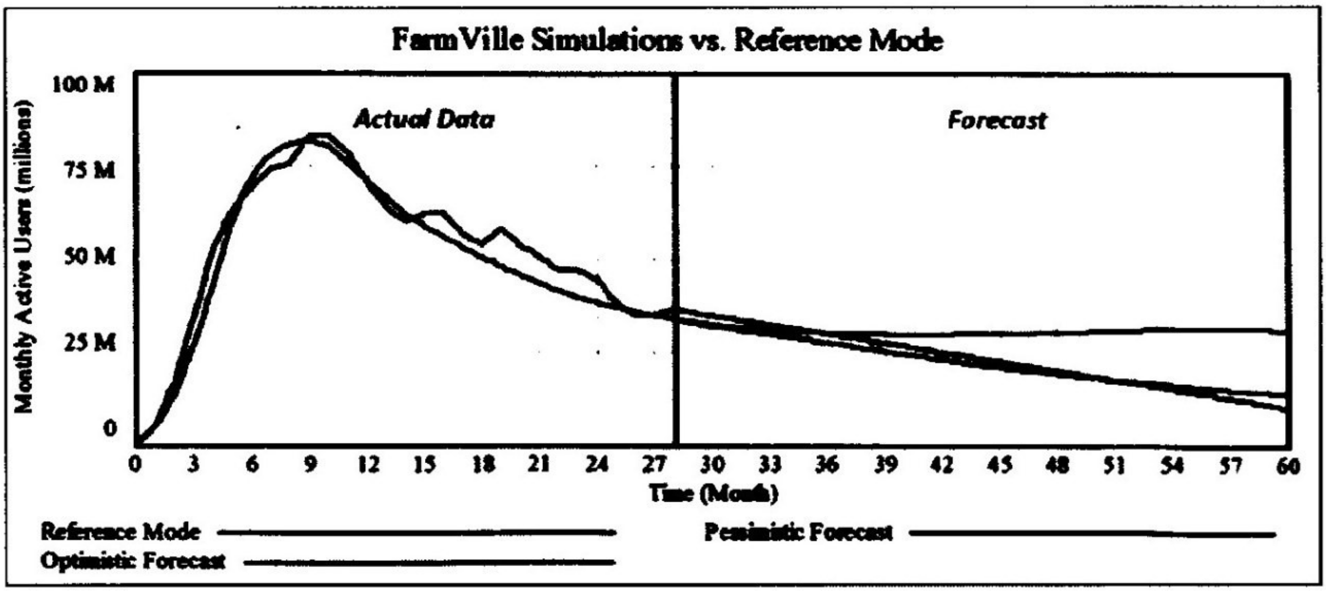

Figure 9 - Lifecycle Behavior Predicting System (Extracted from Speller [2012]).

Tarng, Chen and Huang [2009] made a model to predict the possible abandonment rate for next months; they did it by analyzing the players' behavior over time, identifying a gradual decay of use (decay of UT). Another approach applied by Castro and Tsuzuki [2015], was to monitor the login rate and then create a model to predict the player departure (abandonment).

An interesting fact about these works is the identification of metrics capable to predict the lifecycle behavior.

3.3. Digital Game Genre

Cook [2007] defined a game genre as a set of characteristics of award and risk mechanisms. Games which use in a similar form the same mechanisms were considered games of the same genre. Racing games are an example of games with the same mechanism of award and risk, which consists in winning a race using driving skills. Cook collected the quantity of games produced over time in many genres and identified a behavior which portrays the players' interest . In Figure 10 is possible to identify a growth of production and then a gradual decay over time, similar to the game usage lifecycle detailed previously.

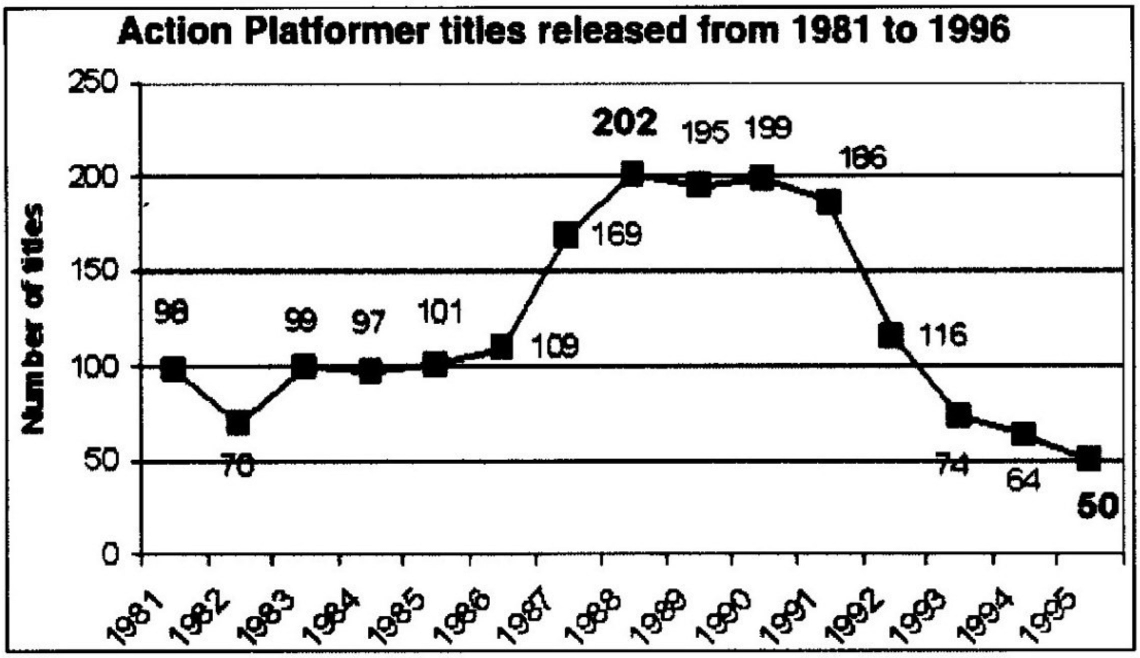

Figure 10 - Quantity of action platformer (genre) games produced over time (Extracted from Cook [2007]).

The game genre lifecycle presents similarities with a game usage lifecycle. There are a growth of production and a decay after a peak. It is possible to define that a game has two lifecycles, the genre and the usage.

Garda [2013] studied the appearance of new game genres. She identified that a game in its end of lifecycle allows the appearance of new genres, changing the mechanisms of award and risk. Those changes aim at improving the aspects which influenced negatively the players motivational factor. When a game producer launch a new version of a game with the mechanisms of award and risk changed (e.g., a flight simulation game to a flight combat game), it was changing the genre of its game, that new genre may be unprecedented or not. New genres can also emerge without basing in other genres.

3.4. Players Profile

Zhu, Li and Zhao [2010] did some interviews with players of a multiplayer on-line game about their motivational factor over time. They identified the following stages, from the most motivated to the least motivated:

- Try: Stage of discovering and curiosity. The first impression is very important to captivate the player.

- Tasting: Player spent more time playing, exploring his options and accumulating "profit" (e.g., items, levels, friends, objectives, etc.). On this stage the player already learned the basic mechanism and start to chase more complex mechanisms.
- Retention: On this stage the player is not more in "love" with the game. Although the game is not more interesting, the player is still playing because his friends play. For the game producer this is a critical stage, because it is necessary to make decisions to reanimate its players, otherwise, the abandonment will occur.
- Abandonment: The interest about the game drops and the player did not spend more time playing.

Cook [2007] also identified player profiles related to the conceptual game genre usage. They are:

- Initial learning: Player learns the game mechanisms, according to the ease of use and the understanding about the game, the player accepts or not the new genre.
- Master: The player dominates and understands how the game mechanisms work.
- Tool: The player uses the game mechanisms more as a tool to achieve his or her objectives.
- Burnout: The game mechanisms do not please the player, and do not provide any more objectives to achieve.

Cook [2007] identified a relation between the player ability and the game genre lifecycle:

- New Players: A new experience of learning and fun.
- Mature Players: Knowledgeable of game mechanisms. Efficient in achieving objectives. The player is a follower of the genre. If a game of interest does not provide new versions, the player looks for other similar games in the same genre.
- Niche Players: Lack of interest. The player loses his ability. They can be characterized in three types: Fire keeper: the player does not give up the genre, and stays playing. Lapsed player: new live objectives prevents the player to play, and his or her abilities drops. Players with no network support: players who found the game for the first time with no references, although this game can be obsolete at the time.

The players' motivational factor was also an object of study for Stewart [2007]. He showed some strategies of development and test of games, with focus in a development directed to the player and not only to the game designer. Ease of use, the learning curve and the interests of the target audience were not discarded being very important to captivate new players. A difficult game to use and understand affects directly the players' motivation.

Um, Kim and Choi [2007] developed a difficulty dynamic regulation system, which uses the player performance to provide the most appropriate degree of challenge, adapting the game to the player.

Alexander [2007] researched about the usage of games by children, identifying the electronic devices used by boys and girls on their youth.

Observing the change of the players' behavior over time, for the conceptual model (genre) and also for the specific game usage, is possible to better understand the full behavior present in the game lifecycle.

3.5. Artificial Intelligence and Data Mining in Games

Besides the researches done about game usage lifecycle, the game field is in constant evolution, opening in that way opportunities to apply Artificial Intelligence techniques (Correa and Pastor [2012]). Some internal behaviors are still a great challenge nowadays (Silver et al. [2016]) and many Data Mining techniques can be helpful to deal with these challenges (Galway et al., [2008]).

Some games have a usage data collect procedure with a detailed step-by-step process of a match (e.g., a list of timestamps containing each one all the events occurred), this detailed data enable model players and then create virtual players based on it, through the application of Data Mining techniques.

In a strategy game called StarCraft[2], Hostetler and colleagues [2012], Weber and Mateas [2009], Leece and Jhala [2014] studied the inference and prediction of strategies adopted in matches. Pingen [2014] made a counter-strategy system based on the enemy strategy identified. Synnaeve and Bessieère [2012] used clustering techniques to detect tactical information and predict attacks. Lewis, Trinh and Kirsh [2011] compared the necessary skills to obtain success in the game with the necessary skills to obtain success in risk activities in the real world. Robertson and Watson

[2] On-line RTS game. It is considered an electronic sport game. Web site: http://us.blizzard.com/pt-br/games/sc/ .

[2014] collected many StarCraft usage databases joining them in one SQL Database and made it available for future researches. Hsieh and Sun [2008] built a virtual player (bot) with the capacity to interpret and understand the enemy strategy. Cho and Kim [2013] compared the behavior between real players and virtual players.

Another game called World of Warcraft[3] was also an object of study due to its usage data. Chen, Pao and Chang [2008] developed a method to identify bots (players controlled by scripts). Tarng, Chen and Huang [2009] created a model to predict the abandonment of players. Lee and Chen [2010] studied the hardware needed to support the game, adopting strategies of energy and hardware saving according to the usage.

As an abstract way, Drachen and colleagues [2012] studied the identification of player profiles through clustering algorithms. They used K-means and Simplex Volume Maximization to identify clusters of behaviors in two games: an RPG and a FPS (first person shooter). Drachen and colleagues identified very specific profiles to each game. Another approach applied by Drachen and colleagues [2014] was to compare the distinct clusters found by distinct clustering algorithms to the same data. They analyzed Archetypal Analysis, K-means, C-means, non-negative Matrix Factorization and Principal Component Analysis. The conclusion was that the K-means algorithm is better to identify the players' profiles than others algorithms, because it is less susceptible to outliers (in games is very common exists players who present a distinct behavior, usually it happens in the case of great fans of a game). A way that they identified the number of clusters was changing the K value and analyzing the resultant profiles until predominant behaviors were identified.

Among all the works cited above, only the works of Tarng, Chen and Huang [2009], Castro and Tsuzuki [2015] and Drachen and colleagues [2012][2014] follows the same purpose of this research, however it does not focus on the commitment feature (an immersion approach considering the time and score) which is our focus. These related works shown the scarcity of researches in our field of interest, which is related to identifying risk situation in game lifecycle.

3.6. Conclusions

On the practical side, the lifecycle prediction of Speller [2012] and the abandonment prediction of Zhu, Li and Zhao [2010] and Castro and Tsuzuki [2015] go in the same way of the interests of this research, because they identified and applied usage metrics to analyze and

[3] Massive multiplayer on-line RPG. It is not considered an electronic sport, but has great success in its genre. Web site: http://us.battle.net/wow/pt/ .

presuppose behavior about the game lifecycle. Drachen and colleagues [2012][2014] shown some good practices and some challenges in the player profile clustering, which we can make use in this research.

On the knowledge base side, Cook [2007], Zhu, Li and Zhao [2010] showed the players' behavior over time, it helps us to better analyze the commitment which we want to extract.

None of the related works studied here presents a systematic way to analyze how a game is in its motivational side, but only on its usage side (MAU's idea). Some works deal with usage and another with motivational factor, but none of them join the two aspects and that is the purpose of this research.

Chapter 4.
Risk Prediction Method

In this chapter we present the proposed method to predict risk situations in the game context.

4.1. Method Assumptions

Usage data allow us to identify who played, when he or she played and his or her correspondent score. In this research, we will use this data to approach a measure of immersion, which we call commitment. Immersion affects the player pleasure sentiment when he or she plays, however we cannot measure that sentiment directly, but we can measure the actions made by the player based on his or her immersion sentiment. We suppose that if a player likes the game (good immersion) his or her performance and time played will grow, improving his or her commitment to that game (attachment). Understanding how a player play (score) and how much time he or she spends playing (UT) we will define degrees of commitment.

The game candidate to the method must have the following assumptions:

- The entertainment as an objective(voluntary use).
- An availability of usage data having for each player:
    - the score obtained;
    - the usage date (when the player played);
    - the player identification (ID);
- A way that allows the player to improve his abilities over time (e.g., in a soccer game, a player can win for 1x0, and after some time, win for 10x0 in the same conditions).

4.2. Method Steps

There are three main steps in the proposed method (Figure 11): extraction of commitment data, prediction of commitment and Risk Computation. There are: an unsupervised stage, a supervised stage and a computation stage. Each step of the process is detailed next.

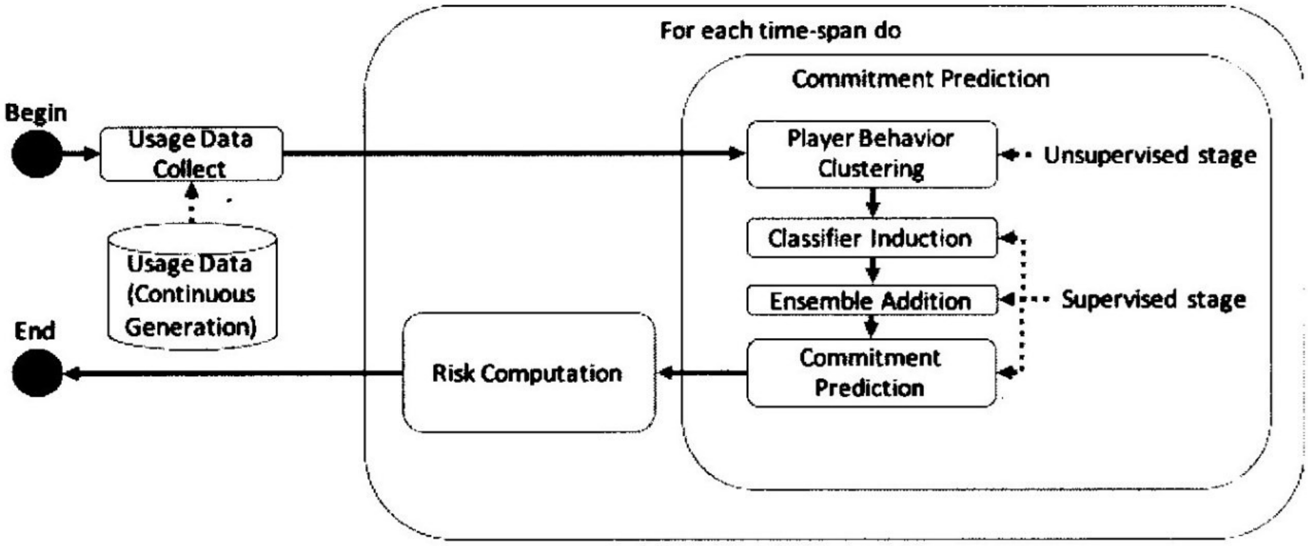

Figure 11 - Risk Prediction Method.

4.2.1. Usage Data Collect

The usage data is the database of the method, based on it the commitment measure is extracted. That measure allows the extraction of commitment metrics, which are used to create the model to predict risk situations. A valid usage data is any data which contains the player ID, the date when he or she played and the score obtained. The score is a game feature which measure the player's ability, for example: the result of a match, a level in an RPG, or the final time of a race.

In order to have the data entry of the method in a correct form, the vector v_i must be constructed:

$$v_i = \{id_i, d_i, s_{i\text{-}min}, s_{i\text{-}max}, \Delta si\}$$

where id_i is the player identification, d_i is the number of days the player *i* played the game in a given period of time, $s_{i\text{-}min}$ is the minimum score achieved in d_i, $s_{i\text{-}max}$ is the maximum score achieved in d_i and Δsi is $s_{i\text{-}max}$ - $s_{i\text{-}min}$. The method times-span granularity can be daily or monthly. For daily times-span, d_i corresponds to the quantity of hours played in a day. To each times-span, each player has a correspondent v_i vector if he or she played. The same player has *n* v_i vectors referring to the *n* times-spans that he or she played.

As an additional knowledge, based on vector v_i it is possible to extract usage metrics, like MAU. MAU is the distinct number of players who played the game in the given times-span (e.g., monthly) and can be represented as shown in Equation 1:

$$MAU = \sum_{i=1}^{n} p_i \qquad (1)$$

where n is the total number of players, p_i corresponds to one distinct player i and has a value equal to 0 when the player did not play, and, 1 otherwise.

Other metrics can also be obtained: new players rate (corresponds to the quantity of new players in the times-span), abandonment rate (corresponds to the quantity of players who left the game in the last times-span) and return rate (corresponds to the quantity of players in the abandonment state who returned to play again). These usage metrics help in better understand the usage of a game.

4.2.2. Commitment Prediction

Commitment data represents in our concept how the player plays, not only if he or she played (usage concept). Therefore, we understand that the time spent and the score obtained in a given times-span (e.g., monthly or daily) corresponds to attachment to the game. To extract the commitment data a few steps are necessary. Initially the identification of players' behavior groups is done, then with each player labeled as a member of a group, classifiers are induced and finally an ensemble of these classifiers is created and applied to discover, for each player, his or her corresponding degree of commitment based on the historical behavior. We understand that, if a player likes the game, he or she will spend more time playing and consequently improve his or her obtained score, therefore, players with a higher score and time spent have a higher degree of commitment. We defined three degrees of commitment: low, average and high. In Chapter 5, a justification of this K value is given (K value represents the number of groups to identify).

To identify the players' group, the K-means algorithm is used. A K value of 3 clusters was set (more details in Section 5.2), which corresponds to the three degrees of commitment (low, average and high). The algorithm plots in a four-dimensional plan (four corresponds to the numbers of v_i attributes used, just excluding the id_i) all the players vectors (v_i) and based on the d_i, $s_{i\text{-}min}$, $s_{i\text{-}max}$ and Δsi the three groups are identified. After that, the identified groups are named as low, average and high according to the mean max level as follows: $s_{max\text{-}low} < s_{max\text{-}avg} < s_{max\text{-}high}$. With the groups named, each member of the groups receives a label which follows the group name (e.g., players of the low commitment group will be labeled as "low"). For each times-span a player has a corresponding degree of commitment. Nothing prevents a player to improve or reduce his degree of commitment over time, he or she can be low for two months and then change to average next month.

Based on the behavior labeled to each player, the classifier induction becomes possible, moving from an unsupervised stage to a supervised one. These classifiers are able to predict future players' profiles based on the historical behavior.

For example, for each month the three degrees of commitment were identified through the clustering step, and then a classifier is induced based on it. After performing some experiments (more details in Chapter 6), the classifier chosen is the Decision Tree. One of the facts that supports this decision is that Decision Trees provide an analysis by the decision maker of how the result was generated (white box idea). Classifiers such as SVM and Neural Network are black box algorithms, not allowing a good understanding about its internal functions.

As for each month we can have different behaviors, for example, a school vacation month or a month with a game upgrade, it is possible to have to the same degree of commitment different interpretations over time. For example, the high committed players in a "common" month play for more than 22 days and have a score greater than 55, but in an upgrade month (when the game receives new contents), they play more than 27 days and have a score greater than 59. These changes of behavior can be upwards or downwards. These changes are what characterize the players' behavior over time.

To take advantage of the behavior diversity which can occur month by month, all of the classifiers correspondent to each month can be used in conjunction, providing a manner to classify players according to their historical behavior. It can be done through the use of an ensemble, creating an observation window according to the number of classifiers within it.

The ensemble is characterized by executing all classifiers over a specific instance. As an internal configuration, the majority vote policy is used, it means that a register is labeled as the most voted class according to the results of all classifiers. After labeling all players, it is possible to compute the commitment for each times-span using Equations 2, 3 and 4:

$$Low = \sum_{i=1}^{n} p_{low} \qquad (2)$$

$$Average = \sum_{i=1}^{n} P_{avg} \qquad (3)$$

$$High = \sum_{i=1}^{n} P_{high} \qquad (4)$$

Each classifier inside the ensemble represents a distinct times-span. As time goes, more classifiers are induced and added to the ensemble. It is possible to assimilate the number of classifiers with the number of months observed. The size of the observation window set to the method is progressive. It means that the size of the observation window will grow for each times-span added, with no limit. In Chapter 6, experiments of limited window size are described (six and 12 more recent months).

The observation window has two main ways to be applied by a game producer. One is when the method to predict risk situations is used from the beginning of commercialization, on this case the observation window will start with the size one after the first month of usage. Another one is when the game is in the market for a while, and then the method is used, in this case the observation window will have the number of months passed from the beginning of commercialization. In both cases, the observation window will increase by 1 when another month or day finishes (it depends on the times-span granularity chosen).

By analyzing the player profile called Niche identified by Cook [2007], these players are characterized as still playing a game with a high commitment degree, even when the rate of new players are almost zero and the game is almost no profitable. The niche stage is the final stage of usage, and its identification as soon as possible is of great value to game producers, because it allows them to take some emergency actions (as shown in Chapter 2). Identify the niche stage in a game is possible with the use of an ensemble (more details in Section 6.2). Based on the fact that the players' behavior changes over time tending to a niche behavior, an observation window which contemplates all the behaviors presented from the beginning of commercialization can identify when a niche stage is occurring (Kummer et al. [2016]). For example, assuming that the current month is a niche month, if only the present month is observed, the three degrees of commitment will be identified with similar numbers of members between them (due to the clustering algorithm behavior), and that does not characterize the niche, because the niche is characterized by a month with the number of high committed players greater than the other degrees. Using an observation window is possible to identify this situation, because the players will be labeled according to the historical behavior, rather than the specific month behavior (derivate from the specific month classifier), allowing in that way the niche stage to be identified. Using the observation window, is possible to identify a month with only one kind of degree of commitment (e.g., a month with only high commitment players according to the historical behavior).

The execution of the ensemble provides as a result the degree of commitment to a given player, as: low, average or high. The commitment can be interpreted as an approach of immersion, because the actions done by the immersion pleasure of the player are expressed in the usage data, representing how the player played. Based on the commitment labeled to each player, it is possible to generate new metrics besides the quantity of players by month are in each degree of commitment (e.g., January 2016 has 5,000 active players(MAU idea), among them 1,000 are low, 1,500 are average and 2,500 are high committed players). The new metrics generated are the rate of conversion between the degrees of commitment, as described in Table 1. If a player likes the game, he or she will play longer and improve his or her score, improving his or her commitment over time. The opposite situation occurs too, because a player who do not like the game anymore will play less time and reduce or stop (e.g., a level in an RPG) his or her score. Based on this idea it is possible to analyze for each player if he or she changed his or her commitment degree from the previous times-span to the current one with a positive or negative influence in the overall commitment of a game.

Table 1 - Changes of commitment degree (commitment metrics).

Metric	Label	Influence
Low to Average	LA	Positive
Average to High	AH	Positive
Low to High	LH	Positive
Average to Low	AL	Negative
High to Average	HA	Negative
High to Low	HL	Negative

4.2.3. Risk Computation

The final Risk Indicator (RI) is computed through a regressor model based on the quantities of: players in low, average and high commitment (Equations 2, 3 and 4) and the rate of conversion between the degrees of commitment (Table 1). The final result is a value between 0 and 1, where a favorable situation will have values close to 1 and an unfavorable situation a value close to 0. What defines these limits of 0 and 1 is the historical behavior. In order to train the regressor model is necessary to give to each month its correspondent Risk Indicator. To calculate it Equation 5 is computed:

$$RI_j^{\cdot} = \frac{(LA + AH + LH) - (AL + HA + HL)}{max(RI_n)} \qquad (5)$$

where *max (RIn)* is the largest *RI* already computed. For the first times-span, *RI* assumes 1. For a negative RI, a normalization is done to put it in a range of 0 and 1. After computing the *RI* value a regressor model is induced.

Having the regressor model ready the game producer can apply the method. In order to do it, the game producer chooses a desired times-span, applies the model and uses the result to evaluate if some decision must be taken. The result can be favorable, where no actions are required, or unfavorable, where some action is required to try to solve the problem. The kinds of possible decisions to act or not is a responsibility of the game producer, the method stops in the moment that the risk indicator is generated.

4.3. Conclusions

Over the game lifecycle, the game content is consumed affecting the players' motivation factor. For this review game producers make some game upgrades to "feed" that motivation with new content. Using the historical usage data, and identifying the months when upgrades were done, it is possible to identify characteristics in the usage and commitment metrics to predict risk situations in a systematic way.

The idea of the proposed method is for example, month by month usage data is collected, a classifier is induced based on the clustering result and added to the ensemble, then commitment metrics are computed (based on the commitment measure). The method execution generates a risk indicator which allows continuous analyzes and management of the game lifecycle based on the changes of the players' motivation over time. It offers an understanding based on a systematic method, not more through an empirical way as it was before, where motivational aspects were analyzed in a subjective way through usage metrics, like MAU (Speller [2012]).

Chapter 5.
Methodological Procedures

This chapter describes how the proposed method was implemented and evaluated.

5.1. Implementation of the Method

The method starts with the data entry in the correct format and finishes when the risk indicator is generated. All usage data used on the method is stored in a MySql database (DB). The usage metrics extraction is done through SQL queries in the DB. As described in Chapter 4, the following variables can be extracted: MAU, the quantity of new players by month, quantity of players who abandoned the game by month and the quantity of players who returned to play by month (the resulting data will be described in Chapter 6).

Differently of usage metrics, the commitment data is obtained through the application of Data Mining techniques. The sequence is: clustering to identify the commitment groups, classifier induction based on the clustering result, ensemble creation using the classifiers previously created and the application of the ensemble to identify for each player his or her commitment according to the ensemble observation window (details in Section 4.2.2). To automate the steps of clustering, classification, ensemble creation and commitment prediction, a tool in Java which uses Weka[4] classes was developed.

On the clustering task, the K-means algorithm through the implementation SimpleKMeans is used. For the classification task, Decision Trees with the J48 implementation (C4.5 algorithm) (Tan, Steinbach and Kumar [2005]) is used. For the regressor which will return the risk indicator is used the M5P algorithm. For the ensemble is used the Vote algorithm. Next, the Data Mining techniques evaluation is detailed, justifying why these configurations were chosen.

5.2. Method Evaluation

The experimental protocol of the cluster task is: limit of 500 iterations (one iteration consists in positioning the centroids and associate the nearest instances, as described in Chapter 2), the

[4] Tool with many Data Mining algorithm implementations. Web site: http://www.cs.waikato.ac.nz/ml/weka/ .

testing data is the training data (this is used because the clustering activity aims at preparing the data for the next steps, not being it a traditional data mining activity where the result must be evaluated through a cross-validation for example). The distance calculation is done by the Euclidian Distance (Equation 6), as the values used are continuous numbers (e.g., level in an RPG and quantity of days played), the distance can be measured by a straight line. Equation 6 describes the euclidian distance between 2 points, to compute that distance between more than 2 points a new pair of xi,yi can be added to the squared root.

$$EuclidianDistance = \sqrt{((x_1 - x_2)^2 + (y_1 - y_2)^2)} \quad (6)$$

In a bi-dimensional plan, a point is given by two coordinates in the axes X and Y, which are represented by x and y variables. In the equation above, $x1$ and $y1$ are coordinates of point 1 and $x2$ and $y2$ are coordinates of point 2. Applying the equation, one obtains the straight line distance between point 1 and point 2.

One aspect in the clustering step is the number of clusters to be set in the algorithm. In order to set this value an experiment was done. The number of clusters (K value) is chosen following the same rule as applied by Drachen and colleagues [2014]. The data used is the v_i vector (details in Chapter 4) without the id_i. Initially a K value of three is set (an assumption that at least three distinct groups exists) and then the K value is increased by one continuously for each new iteration, until not relevant profiles are identified. The dataset used is WOWAH (more details are described in the next Chapter).

The experiment starts observing in a bi-dimensional plan the clustering result for three centroids. We focused on two types of plan, in both the Y axis (vertical) has the quantity of days played in the month, in one the X axis (horizontal) has the quantity of levels improved in the month, and in the other one the X axis has the max level of the player in the month. We have chosen to analyze this perspective because it shows the players' profile (the time spent and the score obtained). However, the clustering step is done using the following v_i attributes: quantity of days played, initial level, final level and the quantity of levels improved in the month. Initially the first month is chosen as a data source. In Figures 12 and 13, the result of the execution with three centroids are illustrated.

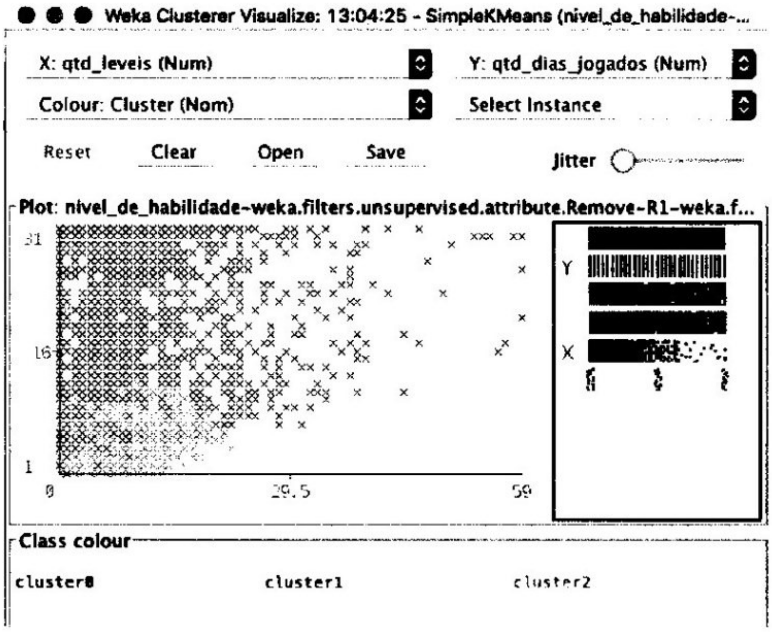

Figure 12 - Clustering with three centroids (X-quantity of levels ; Y-quantity of days played).

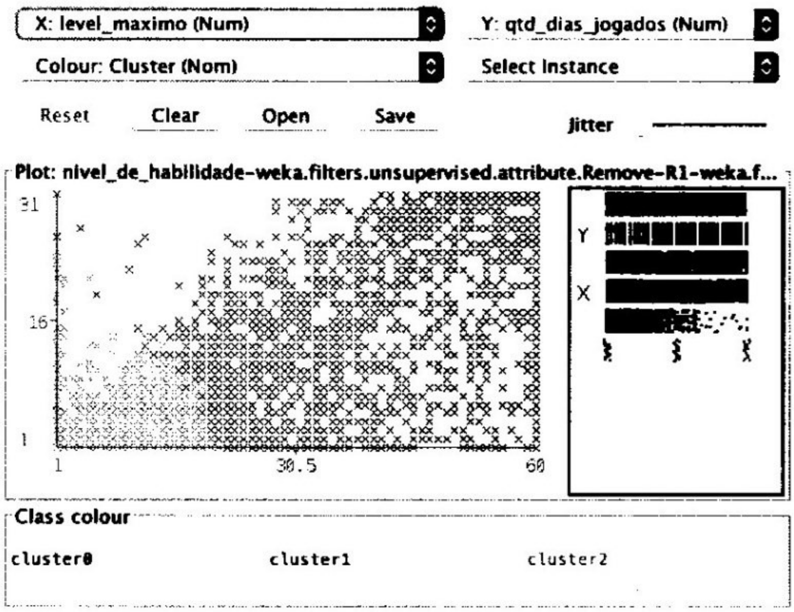

Figure 13 - Clustering with three centroids (X-max level ; Y-quantity of days played).

Analyzing the result it is possible to identify three distinct behaviors:

- Cluster 0: Players play from 1/3 of the total of days in the month to the whole month, do not evolve as quickly as Cluster 1 do, but has the highest score players.
- Cluster 1: Players play about half of the days in the month, they are on an average level and can evolve quickly. As more days are played, more levels are reached.

- Cluster 2: Players with 1/3 of the max level of the game (60 in this case) and evolve its score more slowly than other clusters, they play few days in a month (maximum of two weeks).

The analysis continues changing the number of clusters to 4 and 5 (aiming at identifying other possibles distinct behaviors). The results are described in Figures 14, 15, 16 and 17:

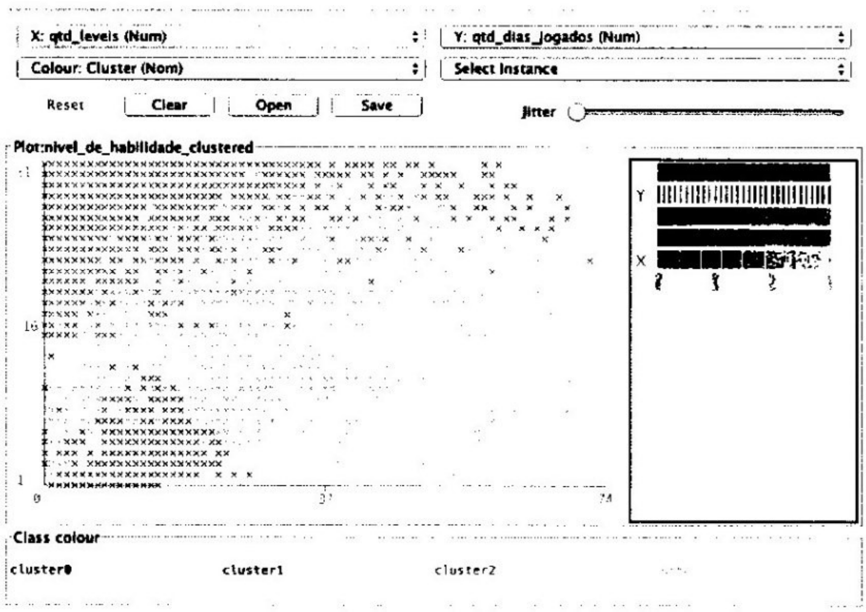

Figure 14 - Clustering with four centroids (X-quantity of levels ; Y-quantity of days played).

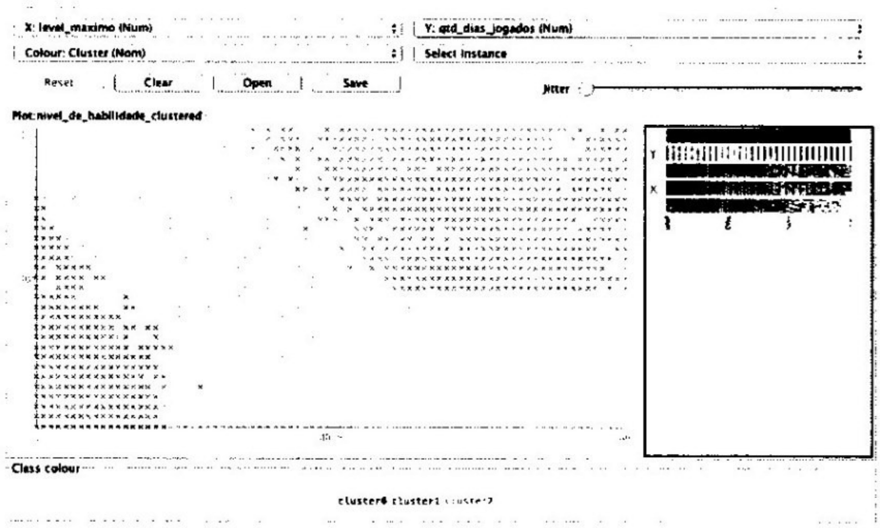

Figure 15 - Clustering with four centroids (X-max level ; Y-quantity of days played).

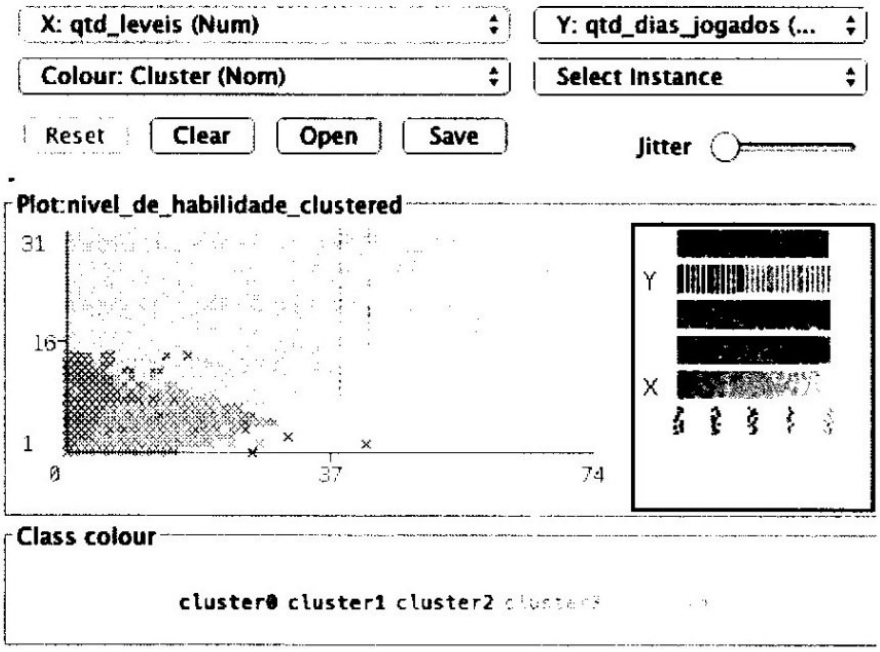

Figure 16 - Clustering with five centroids (X-quantity of levels ; Y-quantity of days played).

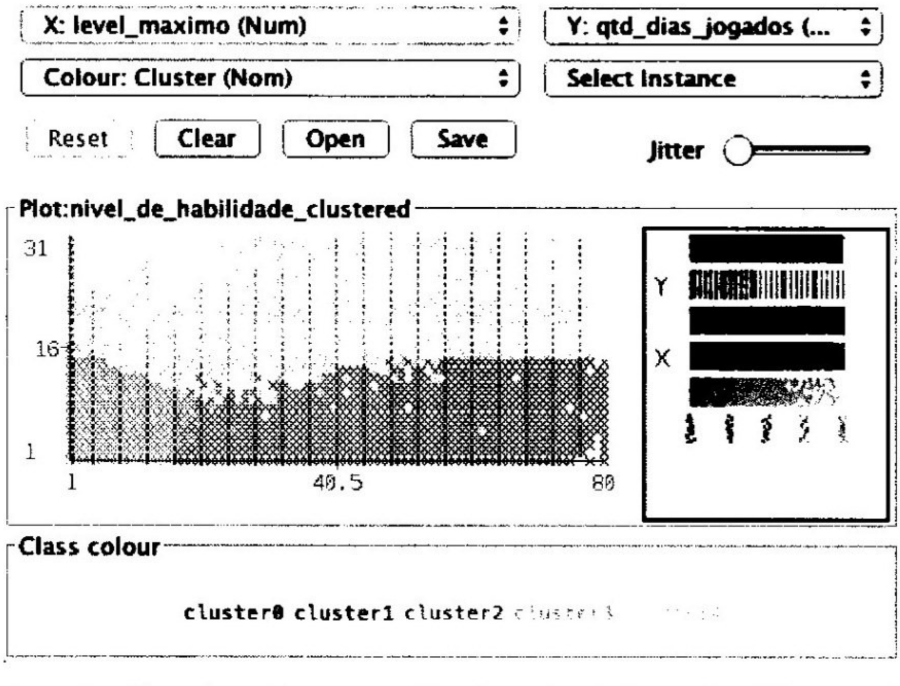

Figure 17 - Clustering with five centroids (X-max level ; Y-quantity of days played).

Evaluating the relation "Quantity of days X Max Level" is possible to identify a distinct group segregation, but it is less clear in the relation "Quantity of days X Quantity of levels", even adding one or two centroids (was tested until nine), the predominant behavior still being the original three clusters behaviors. The added groups were subgroups of the existing groups, and did not reveal any distinct behavior.

The same analysis was done changing the data for the next six months, and the conclusion was the same. After this analysis, three distinct profiles were identified, which we called low, average and high degrees of commitment, being it the result of this experiment.

A low committed player is the player who started to play, having about 1/3 of the total level, grows slowly compared to the others profiles and play about the maximum of two weeks per month. The game World of Warcraft (WOWAH dataset source) has a free monthly payment until the player achieves the level 20, from this point is necessary to make a payment, however, there are paid players in levels lowers than 20. Associating the software lifecycle defined by Moore [1995], this profile represents the player who is going to pass above "The Chasm" of acceptance. In the profile identified by Cook [2007], Zhu, Li and Zhao [2010], this kind of player would be in the curiosity and initial learning phases, where they will discover the game mechanisms of risk and award, igniting or not the "flame" of interest in continuing playing.

Average committed players can be interpreted as a stage after the low committed one. On this situation, the players are not in the initial levels any more, they start to spend more time playing and can grow fast. Associating with the lifecycle, this represents the successfully pass of Moore's "Chasm" and also an entry in a new motivational stage of use, showed by Cook [2007], Zhu, Li and Zhao [2010] as the tasting phase. Now the player aims in accumulating friends, levels and items for example, executing more complex activities, consuming the game content.

High committed players can be interpreted as a stage after the average committed one. Now the player is very close to achieve the highest score, if he has not already done, and consequently grows in a lower rate compared to the others profiles. The fact of achieving the max level is not a desmotivacional factor, because this profile has the greatest time spent playing. Zhu, Li and Zhao [2010] define this stage as the beginning of a risk situation, because the game content was almost, if it has not done yet, consumed by the player and the "flame" of interest starts to extinguish. For the game producer, this stage demands a player retention, it is usually done with game upgrades containing new internal mechanism of risk and award, attracting the player again with some news to know (e.g., new challenges).

Cook [2007] cites a profile called niche, which is derived from what we called high commitment. As the time passes, as well as the Moore's [1995] lifecycle and the example of Speller [2012], the number of active players gradually fall, however, the players who stay playing are great fans and do not abandon it very easily. For the game producer, the game has no financial return and new investments are unmotivated. A game is considered in a niche stage when its active players are mostly composed of high committed players (Kummer et al. [2016]). This is a possible end of the game lifecycle. Zhu, Li and Zhao [2010] call this stage as abandonment stage, because no matter the number of upgrades done, the players will not play in a profitable quantity.

The experimental protocol of classification task is the following: C4.5 default configuration, no pruning adjustment and 10 fold cross-validation. The metric is also the percentage of correctly classified instances (accuracy). For the regressor used to generate the risk indicator the protocol is the following: M5P default configuration, no pruning adjustment and 10 fold cross-validation. The metric is the correlation, because it represents an approximation of how the value returned are similar to the expected value (Equation 5 on Chapter 4). The algorithm chosen was the M5P, because it contains on its leaves an equation, different from RepTree that contains a specific value. This difference enables M5P compute more detailed values than the RepTree algorithm.

For both tasks of clustering and classification, it was chosen to maintain the models the most faithful to the data as possible (it is called 'overfitting', more details in Chapter 2, section 2.2.3.1), allowing the models created to be as close as possible to the players' behavior demonstrated in the time-span. As we desire to label players with the predominant behavior, it is very important to capture each specific behavior presented in the usage data, in order to compare them and choose the predominant.

The disadvantage of using faithful models is the prejudice of the data prediction. The data behavior changes over time, and a good model today can be worse in the future, because of that the models usually were not induced very close to the historical data. But when we use an ensemble with a lot of faithful classifiers to predict a register that disadvantage becomes an advantage, because a faithful classifier is only one in a total of "*n*" classifiers, this allows the use of all specific characteristics presented in every month of observation to identify the commitment of a given player. The objective of the ensemble in this method is labeling players with the most predominant behavior of the historical data. The accuracy comparison in this method is based on the clustering result of a distinct time-span (e.g., month), when the ensemble predominant behavior is different from the time-span specific behavior, the ensemble accuracy for this time-span will decrease, but it does not mean a bad thing (more details in Section 6.4).

5.3. Application of the Method

The indicator generated serves as a way to support the producers' decision-making, independently if the identified situation is a risk situation or not, the option to act or not is their responsibility. The method ends at that point. The indicator range between 0 and 1 represents the worst and the best situation of the historical data, these values are set according to bad and good situations which occurred in the data. Next some examples of interpretation of these values are described: if the method returns 1, it means that the commitment of its players for the given time-span is favorable for the continuity of the game lifecycle. In another way, if the method returns 0.5, it means that the commitment of its players for the given time-span is in a half degree as that once was. If the result is 0.1, this means that the commitment is 10% as that once was. This indicator shows the increase or decrease of players' commitment to a given time-span. All the time-spans have its own risk value generated by Equation 5 (in Section 4.2.3). All these values are normalized based on the max value identified, in that way, there always will have at least one timestamp with a risk value of 1, which represents the best commitment of the players in the series (time-spans).

The risk indicator as a result of the method is one of many indicators that can be used by the game producers. As an example of these other indicators we can mention the profitability and MAU.

5.4. Conclusions

This chapter focuses on describing and justifying the experimental protocol related to the Data Mining techniques. Some decisions are made guided by other authors which faced similar situations (as Drachen and colleagues [2012][2014] in the clustering aspect). The experiments done allowed us to identify configurations to set on the proposed method Data Mining algorithms. Next the final results derived from the method application are described.

Chapter 6.
Experimental Results

This chapter presents the experimental results of this research. The process to extract usage and commitment data is detailed. The original usage data was adapted to the method, the granularity adopted (time-span) is monthly and then the usage metrics were obtained. Despite those usage metrics (e.g., MAU, new players rate,...) were not used in the final Risk Indicator, we kept them because they can help to better understand the player behavior. The data set chosen is WOWAH, which is described in Section 6.1.

6.1. Usage Data Sources

It is important to mention that usage data is usually held in the producers' "safe box", being it a confidential information. We identified three mechanisms which provide game data.

The first was the game League of Legends[5]. The API provides access to many kinds of data, among them are the usage data (beta tests). The second was from the game Star Craft. That game allows in the end of a match a generation of a file containing the match replay. This replay is a file with extension ".rep" and can be processed in specifics APIs, which allow the extraction of usage data. Among the three mechanisms found, this is the oldest one.

The third mechanism found was chosen for this research, because, among the available options, it was the one which presented the biggest collection period(three continuous years) and has all the needed characteristics (score, date of use and player id) to apply the proposed method. This data set is a research of Lee, Chen, Cheng, and Lei [2011] called WOWAH (World of Warcraft Avatar History Data set). They collected for a period of three years, every 10 minutes a list of on-line players of the game World of Warcraft. The game profitability is made through monthly payments. The data set is available at: http://mmnet.iis.sinica.edu.tw/dl/wowah/ . The usage data follow the format described Figures 18 and 19.

[5] On-line RTS game. It is considered an electronic sport. Web site: http://br.leagueoflegends.com .

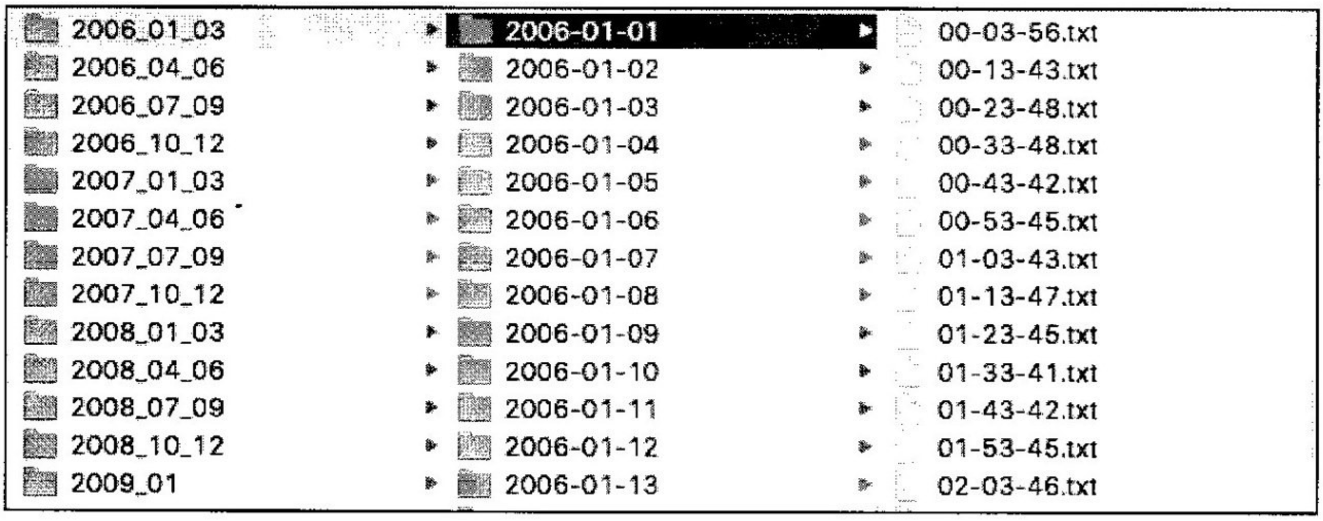

Figure 18 - WOWAH data set.

The period from January 2006 to January 2009, for every 10 minutes exists a corresponding file containing the list of on-line players. The file has the following format:

```
00-13-43.txt

Persistent_Storage = {
        [1] = "0, 01/01/06 00:09:38, 1,407, , 1, Orc, Hunter, Durotar, no, 0",
        [2] = "0, 01/01/06 00:09:38, 1,0, , 5, Orc, Warrior, Durotar, no, 0",
        [3] = "0, 01/01/06 00:09:44, 2,6, , 18, Orc, Warlock, Orgrimmar, no, 0",
        [4] = "0, 01/01/06 00:09:44, 2,2, , 13, Orc, Shaman, Durotar, no, 0",
        [5] = "0, 01/01/06 00:09:44, 2,3,0, 14, Orc, Warrior, Durotar, no, 0",
        [6] = "0, 01/01/06 00:09:44, 2,4, , 14, Orc, Shaman, Durotar, no, 0",
        [7] = "0, 01/01/06 00:09:44, 2,5, , 16, Orc, Hunter, The Barrens, yes, 0",
        [8] = "0, 01/01/06 00:09:44, 2,7, , 17, Orc, Hunter, Silverpine Forest, yes, 0",
        [9] = "0, 01/01/06 00:09:49, 3,8,0, 26, Orc, Warrior, Stonetalon Mountains, yes, 0",
        [10] = "0, 01/01/06 00:09:49, 3,9,1, 27, Orc, Hunter, Stonetalon Mountains, no, 0",
        [11] = "0, 01/01/06 00:09:49, 3,10, , 24, Orc, Hunter, The Barrens, no, 0",
        [12] = "0, 01/01/06 00:09:49, 3,11,2, 25, Orc, Hunter, The Barrens, yes, 0",
        [13] = "0, 01/01/06 00:09:49, 3,12, , 21, Orc, Hunter, Thunder Bluff, no, 0",
        [14] = "0, 01/01/06 00:09:54, 4,13,3, 33, Orc, Warrior, Dustwallow Marsh, no, 0",
        [15] = "0, 01/01/06 00:09:54, 4,14,4, 35, Orc, Warrior, Orgrimmar, no, 0",
        [16] = "0, 01/01/06 00:09:54, 4,408, , 31, Orc, Hunter, Hillsbrad Foothills, no, 0",
        [17] = "0, 01/01/06 00:09:54, 4,16,6, 32, Orc, Warlock, Ashenvale, yes, 0",
        [18] = "0, 01/01/06 00:09:54, 4,17,7, 35, Orc, Hunter, Stranglethorn Vale, no, 0",
        [19] = "0, 01/01/06 00:09:54, 4,18,7, 39, Orc, Hunter, Stranglethorn Vale, no, 0",
        [20] = "0, 01/01/06 00:09:54, 4,15,5, 40, Orc, Hunter, Scarlet Monastery, yes, 0",
        [21] = "0, 01/01/06 00:09:54, 4,19, , 36, Orc, Rogue, Thunder Bluff, no, 0",
        [22] = "0, 01/01/06 00:09:54, 4,20,8, 39, Orc, Warlock, Thunder Bluff, no, 0",
        [23] = "0, 01/01/06 00:10:00, 5,21,9, 45, Orc, Hunter, Wailing Caverns, yes, 0",
        [24] = "0, 01/01/06 00:10:00, 5,22,10, 48, Orc, Warrior, Tanaris, no, 0",
        [25] = "0, 01/01/06 00:10:00, 5,23,2, 41, Orc, Hunter, Orgrimmar, yes, 0",
        [26] = "0, 01/01/06 00:10:00, 5,29, , 41, Orc, Hunter, Orgrimmar, no, 0",
        [27] = "0, 01/01/06 00:10:00, 5,24,11, 42, Orc, Shaman, Orgrimmar, yes, 0",
        [28] = "0, 01/01/06 00:10:00, 5,25,12, 42, Orc, Shaman, Orgrimmar, no, 0",
```

Figure 19 - WOWAH file example (usage data).

The attributes for each instance are the following (attributes are delimited by comma marks in each line):

1. The value used by the WOWAH researchers for internal analysis. It does not have utility in this research.
2. Data gathering time. Example: 01/01/06 00:09:38.
3. Collect ID. It does not have utility in this research.
4. Avatar_id is the player's identification. Example: 126.
5. Guild_id is the player group identification. It does not have utility in this research.
6. Level is the actual score obtained by the player. In games like World of Warcraft, the level never goes down, it can stay at the same value or go up. In the period of three years, the limit values for level were from 1 to 80. However, this upper value varied over time. From 01/2006 to 03/2007 it was 60, from 04/2007 to 10/2008 it was 70 and from 11/2008 upward it was 80. These changes occurred through game upgrades.
7. Avatar breed is the tribe whom the avatar belongs. It does not have utility in this research.
8. Avatar class is the avatar's profession, it can represents a military or commercial role (the essence of an RPG (role-playing game)). It does not have utility in this research.
9. Zone corresponds to the region, where the player is in. It does not have utility in this research.
10. The Value used by the WOWAH researchers for internal analysis. It does not have utility in this research.
11. The Value used by the WOWAH researchers for internal analysis. It does not have utility in this research.

Among all these characteristics, for this research the following attributes are used: data gathering time, avatar_id and level (score). As these data are in the granularity of hours and minutes, a preprocessing was done to change the granularity to month (more details in Section 6.2).

In order to obtain more knowledge about the data sources, we searched for related works related to them. For League of Legends no related works were found. For StarCraft nine works with diversified focus were identified and for WOWAH three works were found. All works are described in Chapter 3.

6.2. Data Preprocessing

Initially the original data were evaluated (Figure 20) and then the attributes of interest were chosen. A problem is that these data have just the data gathering time, player identification (avatar_id) and his or her level (score), in an hour/minute granularity. In order to obtain more attributes, like the quantity of levels increased, we opted to change the granularity to month, instead of a view of every 10 minutes. It was done through scripts applied in the MySql database.

```
Persistent_Storage = {
        [1] = "0, 12/31/05 23:59:46, 1,0, , 5, Orc, Warrior, Durotar, no, 0",
        [2] = "0, 12/31/05 23:59:46, 1,1, , 9, Orc, Shaman, Durotar, yes, 0",
        [3] = "0, 12/31/05 23:59:52, 2,2, , 13, Orc, Shaman, Durotar, no, 0",
        [4] = "0, 12/31/05 23:59:52, 2,3,0, 14, Orc, Warrior, Durotar, no, 0",
        [5] = "0, 12/31/05 23:59:52, 2,4, , 14, Orc, Shaman, Durotar, yes, 0",
        [6] = "0, 12/31/05 23:59:52, 2,5, , 16, Orc, Hunter, The Barrens, yes, 0",
        [7] = "0, 12/31/05 23:59:52, 2,6, , 18, Orc, Warlock, The Barrens, no, 0",
```

Figure 20 - WOWAH original data.

In the new arrangement more data of interest was obtained. Based on this month perspective was extracted for each player: the quantity of days played, initial level, final level and the quantity of levels increased, as shown in Table 2:

Table 2 - Preprocessed WOWAH data.

avatar_id	year_month	qty_days_played	min_month_level	max_month_level	qty_levels
0	2006-01	3	5	12	7
1	2006-01	2	9	10	1
2	2006-01	14	13	17	4
2	2006-02	9	17	18	1
2	2008-12	1	18	18	0
3	2006-01	4	14	17	3
4	2006-01	4	14	17	3
4	2006-03	1	17	17	0

For a total of 36,513,647 instances of the original data, it changed to a total of 282,780 instances on the preprocessed data.

6.3. Usage Metric Extraction

Based on the monthly data the following usage metrics were extracted: MAU, new players rate, abandonment rate and return rate as follows.

- MAU: It is the quantity of active players per month. Speller [2012] showed the use of this metric. In Figure 21, the MAU distribution in WOWAH data set is illustrated. In 04/2007 and 10/2008 exists peaks of usage derived from game upgrade which changed the max level of the game.

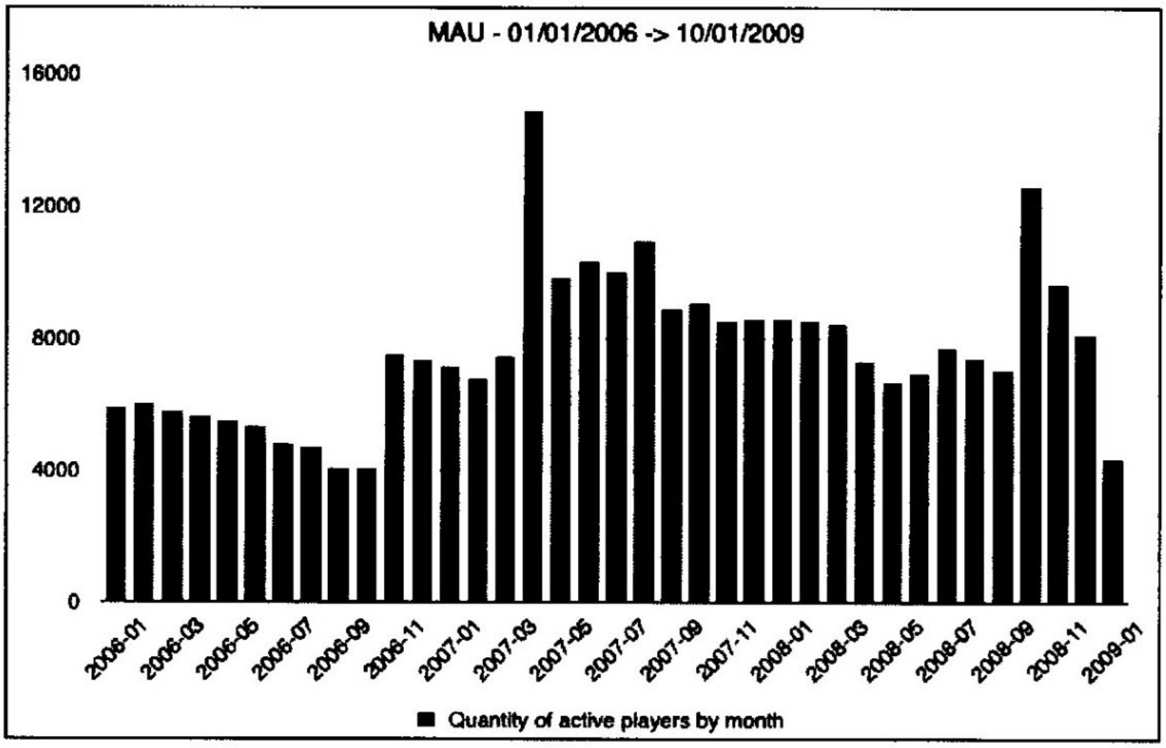

Figure 21 - MAU in WOWAH data set.

- New players rate: It is the quantity of new players per month. Speller [2012] showed the use of this metric. Figure 22 illustrates its distribution over the WOWAH data set. As occurred with MAU, the new players rate has peaks related to months with game upgrades, it shows that upgrades attract new players.

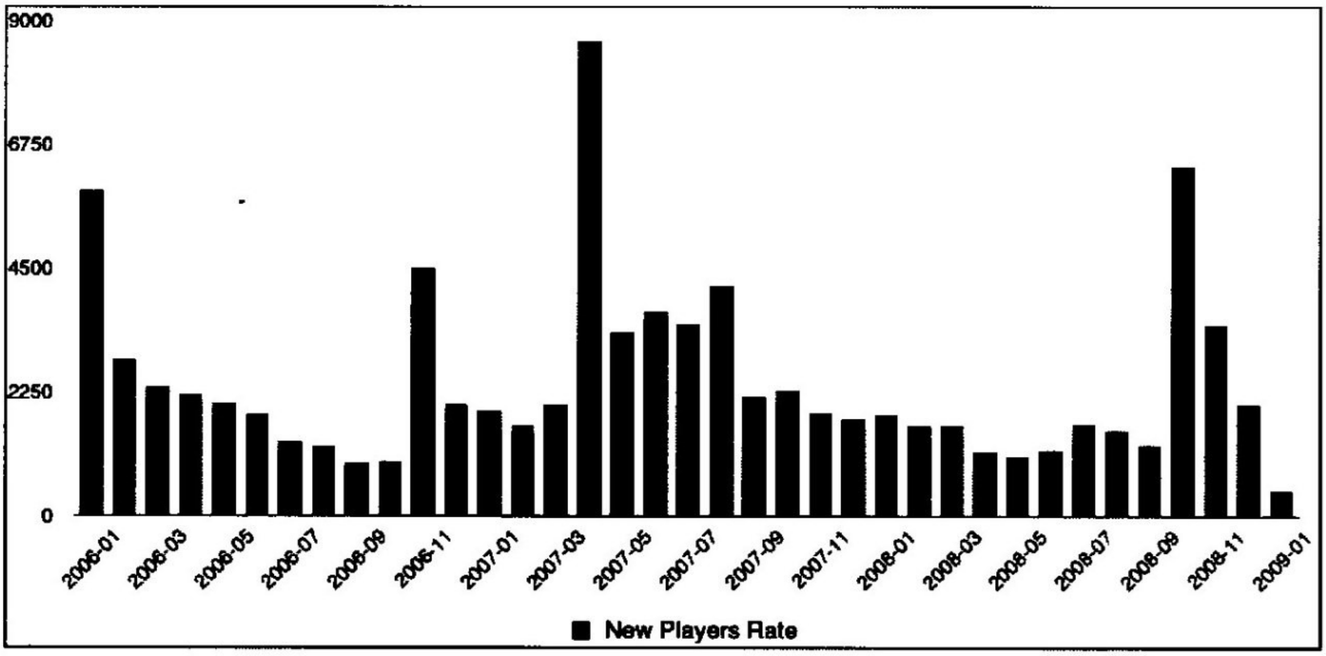

Figure 22 - New players rate in WOWAH data set.

- Abandonment rate: It is the quantity of players who stopped to play. Tarng, Chen and Huang [2009] used this concept in the prediction of the abandonment of players. However, in this research, it is assumed that a given player abandoned the game when he or she did not play in the previous month. In Figure 23 it is possible to identify that after two months of great upgrades there exists great abandonment rates. This is a signal of lack of motivation with the new game content.

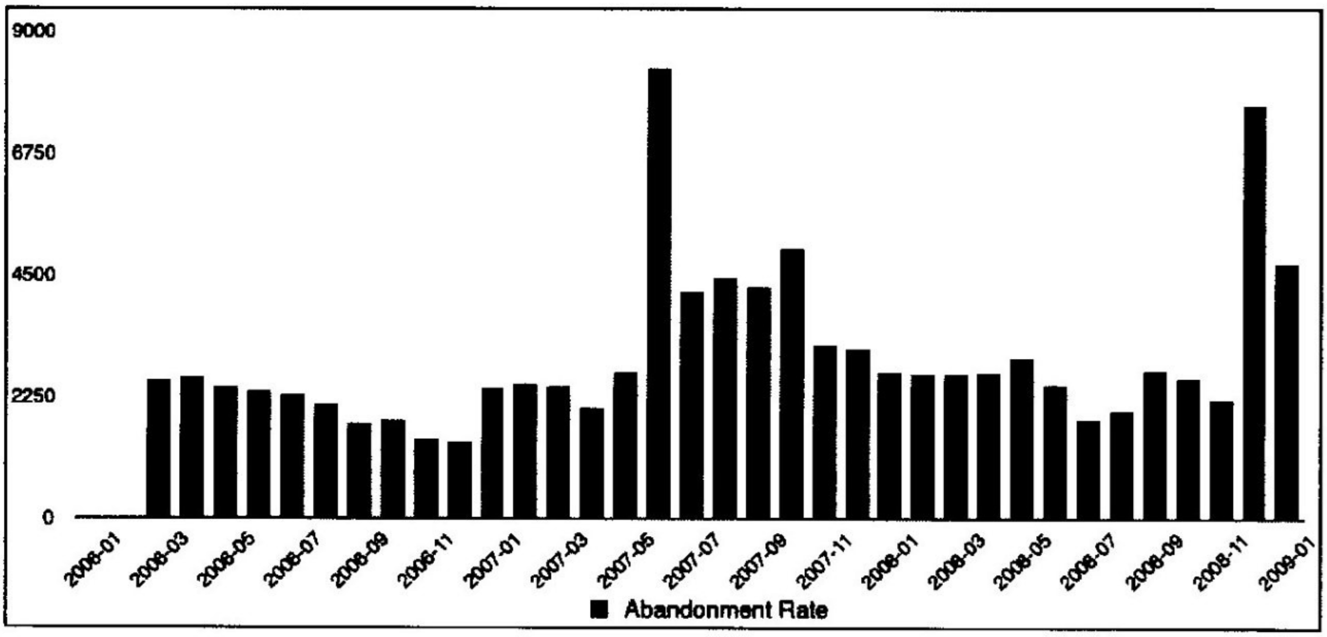

Figure 23 - Abandonment rate in WOWAH dataset.

- Return rate: it is the quantity of players who has abandoned the game and then returned to play again. This metric is a contribution of this research. Table 3 presents an example of this metric.

Table 3 - List of months without playing.

avatar_id	year_month	months_without_play
0	2006-01	0
1	2006-01	0
2	2006-01	0
2	2006-02	0
2	2008-12	33
3	2006-01	0
4	2006-01	0
4	2006-03	1
5	2006-01	0

Studying the return rate over time, it was possible to identify that among the group of players who abandoned the game (one month without playing), there exists a greater probability of players to return to play after one month than when two or more months have passed. Figure 24 shows this return rate behavior over time.

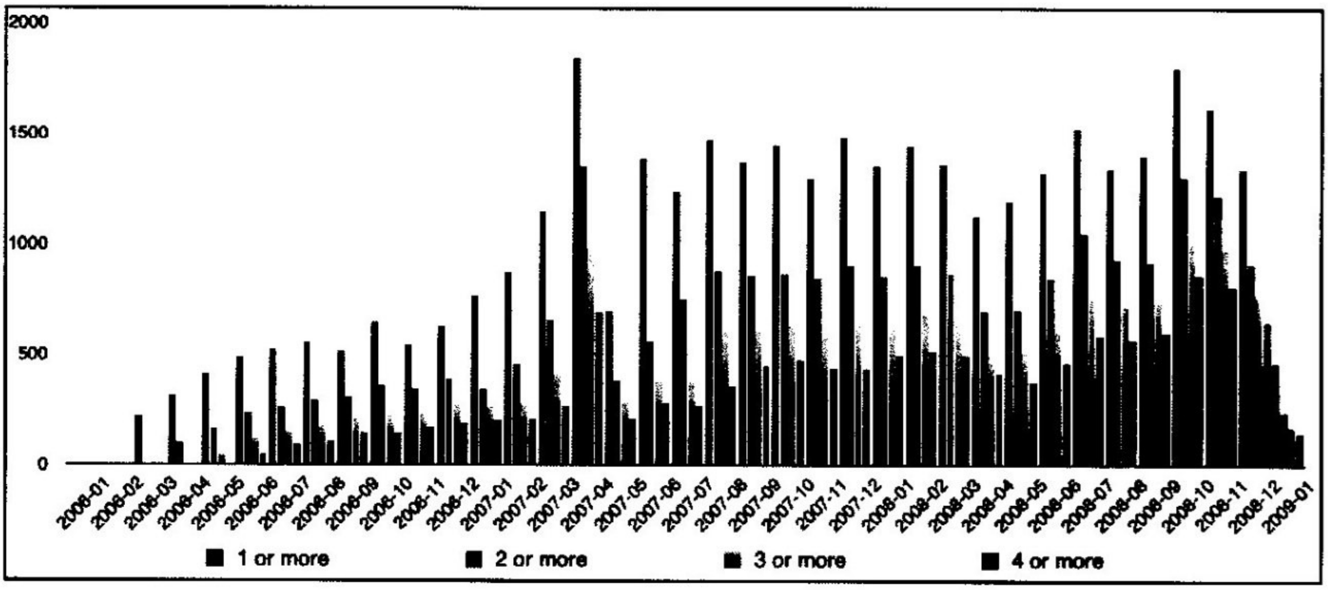

Figure 24 - Return rate over time.

6.4. Commitment Metric Computation

Having the preprocessed data, Data Mining techniques were applied to identify the commitment groups. We used the Weka tool (version 3.7.13). For the clustering step was chosen the K-means through the implementation SimpleKMeans. The algorithm has a limit of 500 iterations (default configuration), for the WOWAH data set, the algorithm usually stops after 10 iterations.

After identifying the distinct players' profile, each player on the database was labeled (examples in Table 4). Initially a clustering month by month was done and the players labeled according to it. With the labeled data, it allows the induction of classifiers based on the group created (classes). Classifiers are predictors, because they have the capacity of, given a training data, predict future instances based on the historical behavior learned. The player profile (low, average or high) corresponds to its class.

Table 4 - Instances labeled according to the cluster profile.

avatar_id	year_month	qty_days_played	min_month_level	max_month_level	qty_levels	commitment
0	2006-01	3	5	12	7	low
1	2006-01	2	9	10	1	low
2	2006-01	14	13	17	4	low
2	2006-02	9	17	18	1	average
2	2008-12	1	18	18	0	low
3	2006-01	4	14	17	3	low
4	2006-01	4	14	17	3	low
4	2006-03	1	17	17	0	low
5	2006-01	3	16	25	9	average
6	2006-01	5	18	20	2	average
6	2006-02	3	20	20	0	average

For each month it was created a decision tree based on the clustering data of the same month. The decision trees were generated according to the following experimental protocol: J48 implementation (algorithm C4.5), default configuration, no pruning and test executed with a 10-fold cross-validation. A comparison was done with the performances of others classifiers: MLP (Neural Network) and SVM. Figure 25 shows the accuracy percentage of all the classifiers for all WOWAH months.

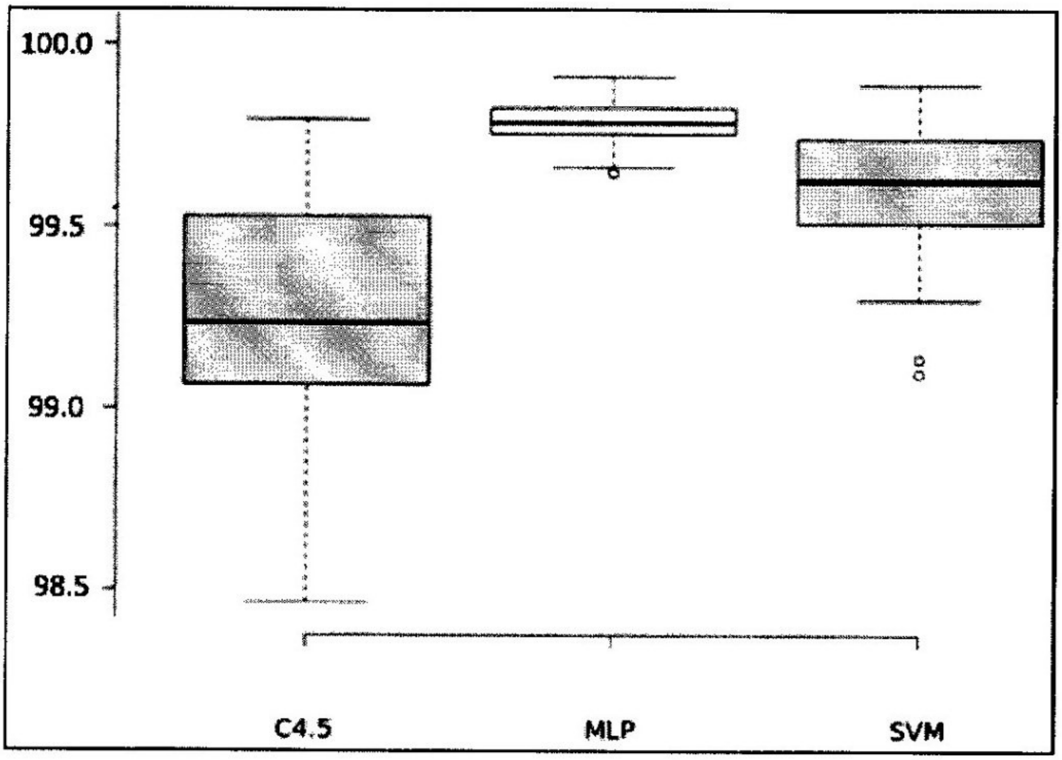

Figure 25 - Percentage of accuracy in class prediction.

A parametric t test was applied with $p < 0.05$. No differences were found between MLP and SVM, however, they both were statically better than C4.5. Despite the decision tree having the worst accuracy, it is not a bad one, because its lowest value was 98.485, what is very close to 100%.

C4.5			
Low	Average	High	
2541	18	0	Low
23	935	15	Average
0	18	1271	High

MLP			
Low	Average	High	
2552	7	0	Low
1	971	1	Average
0	3	1286	High

SVM			
Low	Average	High	
2554	5	0	Low
3	968	2	Average
0	12	1277	High

Figure 26 - Confusion Matrix for C4.5, MLP and SVM regarding to 07/2006.

For the of C4.5 worst accuracy month (07/2006), the confusion matrix was extracted, as shown in Figure 26. It is possible to verify that in none of the three classifiers a low committed player was labeled as high, the opposite situation did not occur too. All the errors were related to the "neighbor" commitment degree (e.g., low classified as average and average classified as high).

Classifiers aim to predict the future based on the past, one important aspect is that the data behavior changes over time, therefore an accuracy of almost 100% today cannot be the same in the future. To deal with this situation, it is common to use pruning in the Decision Tree, making it more generic (reducing the accuracy) and adapting it to changes in the data behavior.

In this research, it was chosen to maintain the Decision Trees accuracy close to 100% (as show in Figure 25), because we want to induce models loyal to the player behavior detected in a given month.

Ensemble refers to an inductor model which uses the opinion of many classifiers to label instances. The collect period allows the creation of 37 classifiers (one for each month), where each one has its own opinion about the class of a given player. The ensemble works through the majority vote, in other words, for each instance being evaluated, the class more returned as a result for all classifiers is the class labeled to the register. The quantity of classifiers in the ensemble represents the quantity of months present in the usage data (observation). It is expected that exists differences between the class discovered in the clustering and the class discovered in the ensemble, because the observation window enabled by the use of the ensemble allows a view of the players' behavior over time, contemplating changes in the player behavior. One example of when the behavior changes occur is in the availability of new game content (upgrades). For the WOWAH data set some upgrades were identified, some of them aimed at disposing new challenges and others in growing the max level. These dates were in: 11/2006, 04/2007, 08/2007 and 10/2008.

After creating the ensemble with the observation window of three years, each register was labeled according to the historical behavior. With this new data a comparison between the clustered data with the ensemble data can be made. The comparative graphs of this analysis are illustrated in Figures 27, 28 and 29:

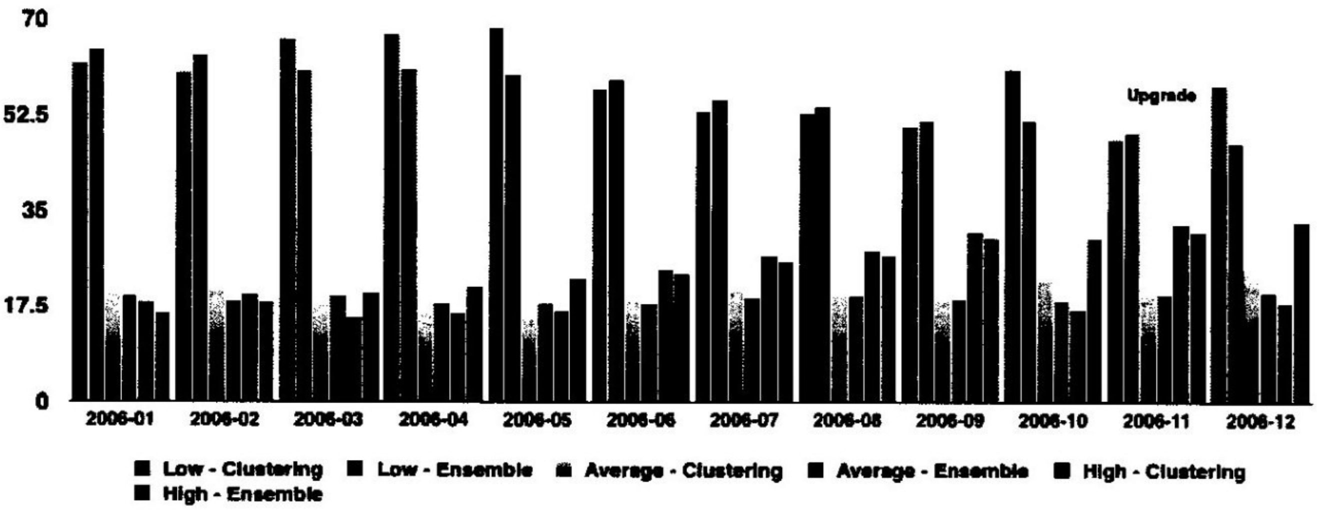

Figure 27 - Comparison between the clustered data and the ensemble data for 2006.

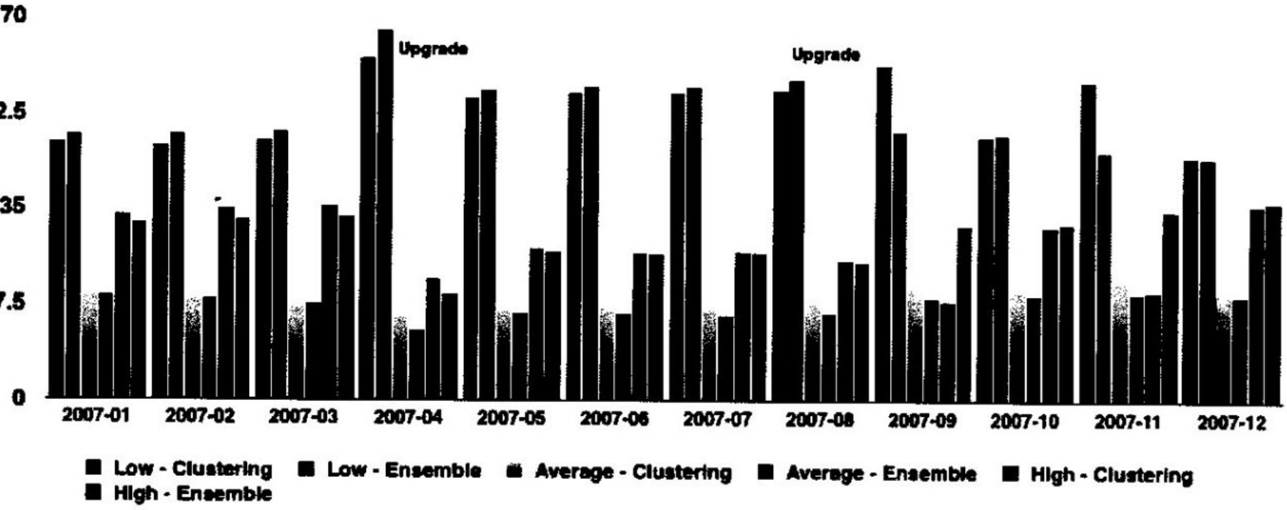

Figure 28 - Comparison between the clustered data and the ensemble data for 2007.

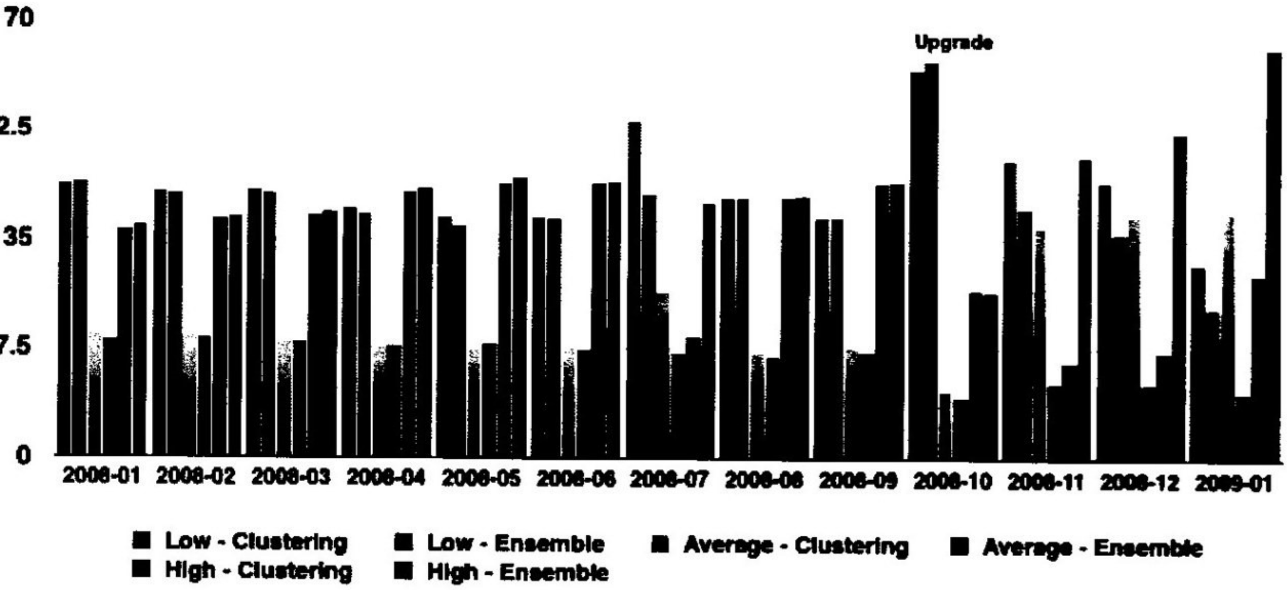

Figure 29 - Comparison between the clustered data and the ensemble data for 2008 + 01/2009.

After analyzing the result in these three figures it was possible to identify differences between the clustered data and the ensemble data. For low and average players the difference is small, normally it is not over 10%. However, for the high commitment class occurred a larger difference, in some cases it was over 40% (e.g., 2008-11, 2008-12 and 2009-01). For the upgrade months, differences usually occur in the previous and in the posterior months (e.g., 2006-11). Months with school holiday and many holidays can affect the player behavior too.

With these commitment data is possible to predict the Niche stage (Cook [2007] and Kummer et al., [2016]). Remembering, a Niche stage occurs when the majority of the players are high committed ones ("in love"). Linking this idea to the commitment prediction, a month can be labeled as a Niche month, when the quantity of high committed players is greater than the number of low committed ones. In the WOWAH data set, in some months the number of high committed players was greater than the number of average and low ones summed up. In Figure 30 the MAU is compared to the commitment, it is possible to understand the commitment metric as a discretization of MAU, in other words, commitment identifies inside the MAU the motivational usage in three degrees.

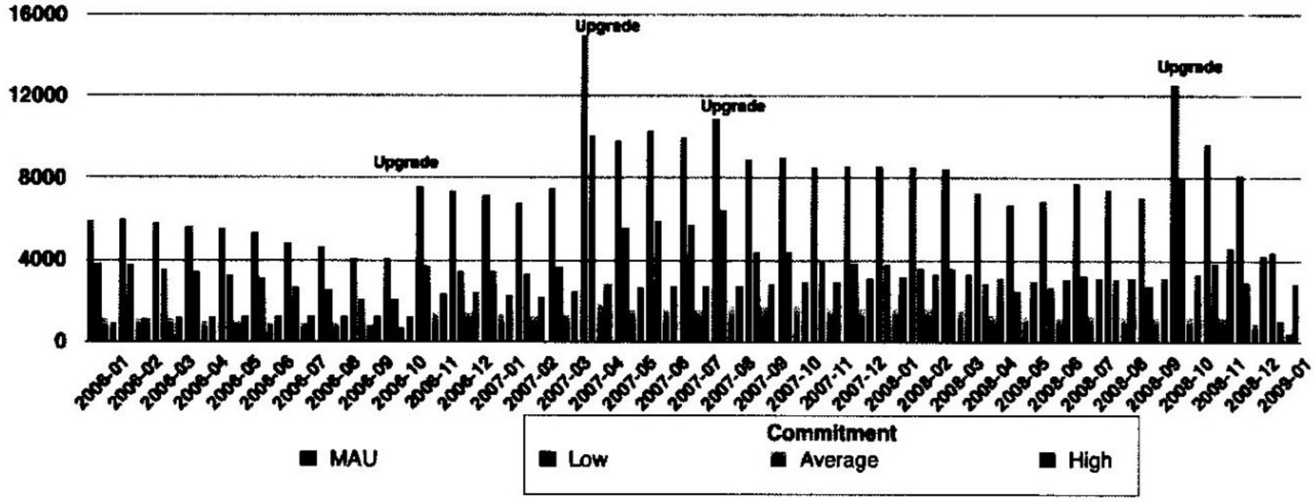

Figure 30 - MAU x Commitment in WOWAH data set.

When an upgrade exists, the MAU grows. In the commitment context, the number of low commitment players follows that grow, it means that the upgrades captivated new players. The number of average and high committed players stays stable, with an upgrade or not.

An assumption relative to game usage lifecycle is that the player behavior tends to be a Niche behavior over time. It can be illustrated through the percentage of each degree of commitment and its changes. Figure 31 shows that variation.

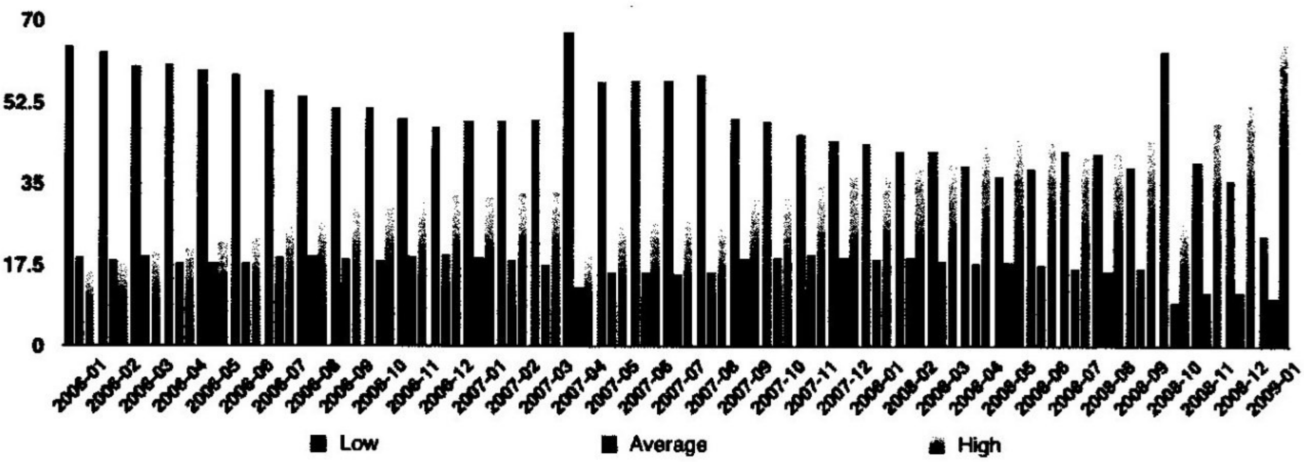

Figure 31 - Percentage of each commitment degree.

It is possible to identify a behavior that as the time advances, the number of low committed players drops and the number of high committed ones grows. When an upgrade is done, the difference between low and high increase. In the 28th month (2008-04), for the first time the percentage of high committed players is greater than the low committed ones. Even the occurrence of three previous game upgrades (11/2006, 04/2007 and 08/2007). This behavior illustrates the assumption that the player behavior tends to be a Niche behavior. Following the Niche description, we computed for each month, if the Niche exists or not (described in Table 5).

Table 5 - Niche computation result.

Month Count	Month	Stage
27	2008-03	Other
28	2008-04	Niche
29	2008-05	Niche
30	2008-06	Niche
31	2008-07	Other
32	2008-08	Niche
33	2008-09	Niche
34	2008-10	Other
35	2008-11	Niche
36	2008-12	Niche
37	2009-01	Niche

The Niche had been detected for the first time in 2008-04, but in some further months it does not occur again. In 2008-10 an upgrade was done, it changed the commitment in a composition that the Niche stage was not detected, but in the next month, the Niche occurred again. It means that the upgrade was not good enough to keep new players playing. As a future work, we intend to research ways to predict the other stages of the game usage lifecycle. Until now we can just predict the existence or not of the Niche stage, however, it enables an analysis of a new information that the MAU cannot provide. Taking 2008-09 as an example (Figure 30), in the MAU point of view, the upgrade of 2008-10 was a success, because it increased the number of players in 2008-10 and in 2008-11, but unfortunately, in 2008-12 it drops to a value lower than the value before the upgrade.

In Niche perspective, 2008-11 was a risk month (Niche), different from MAU, which considered it as a good one.

Another analysis was focused on the size of the observation window related to the Niche stage detection. For an observation window of one month (clustered data for example), the quantity of players on each degree of commitment (low, average and high) can represent an atypical month related to the others, then this "opinion" about the commitment can be very different from the others classifiers' opinions. Table 6 contains a comparison of Niche detection between the specific monthly classifier and the ensemble. It is possible to identify that all the Niche months identified by the specific monthly classifier are also identified by the ensemble. However, there are some months that the ensemble identifies as Niche and the specific classifier does not (2008-11, 2008-12 and 2009-01). This experiment shows that the use of an ensemble helps in better identifying when the Niche occurs.

Table 6 - Ensemble and Monthly Classifier Niche Detection Comparison.

Month	Classifier Approach	Low	Average	High	Niche
2008-04	Ensemble	2851	1315	3145	X
	Monthly Classifier	2928	1288	3095	X
2008-05	Ensemble	2469	1225	2989	X
	Monthly Classifier	2574	1164	2945	X
2008-06	Ensemble	2654	1210	3061	X
	Monthly Classifier	2662	1227	3036	X
2008-07	Ensemble	3268	1304	3157	
	Monthly Classifier	4171	2059	1499	
2008-08	Ensemble	3087	1221	3125	X
	Monthly Classifier	3090	1250	3093	X
2008-09	Ensemble	2727	1205	3129	X
	Monthly Classifier	2711	1250	3100	X
2008-10	Ensemble	8013	1234	3357	
	Monthly Classifier	7827	1397	3380	
2008-11	Ensemble	3851	1166	4648	X
	Monthly Classifier	4586	3598	1481	

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2009-01	Ensemble	1054	474	2887	X
	Monthly Classifier	1369	1743	1303	

Another perspective of the comparative analysis between the ensemble and the specific monthly classifier is the accuracy point of view. Figures 32 and 33 show that comparison. In this experiment, for each month, the correspondent C4.5 classifier and the ensemble were applied and the accuracy collected and compared. That comparison was based on the clustering result of each month.

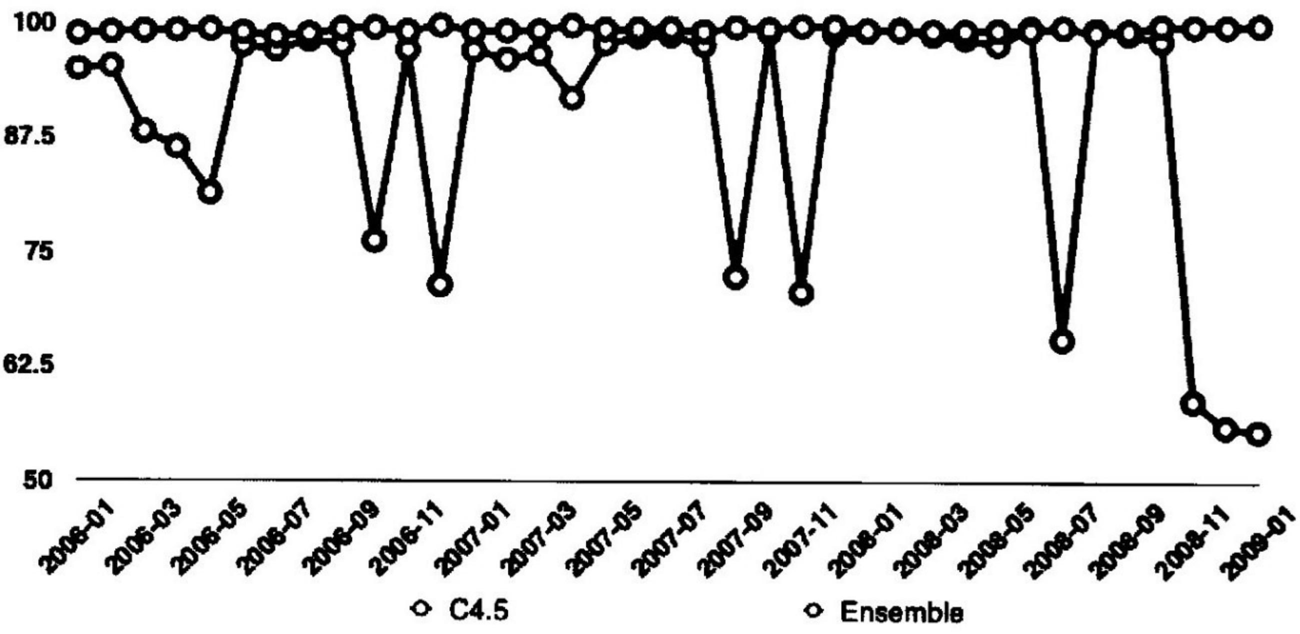

Figure 32 - Accuracy comparison between clustering and ensemble.

To better understand what is described in Figure 32 lets take the fifth month as an example (2006-05). What defines the 100% of accuracy is the label provided by the clustering algorithm, if a classifier predicts all instances as the same label of the clustering its accuracy is 100%. Having the C4.5 overfitted, its accuracy is close to 100%, different from ensemble, where it is close to 80%. This difference symbolizes how the current month (2006-05) behavior differs from the most predominant behavior of the usage lifecycle. It is a comparison between the local point of view and the holistic point of view. Figure 33 illustrates from another perspective the differences between those points of view. It is interesting to highlight the last three months, where a great difference (almost 50%) exists between the opinions of the specific classifier of each month (2008-11, 2008-12 and 2009-01) to the ensemble. It is possible to conclude that these percentages are a kind of correction applied by the ensemble over the specific monthly classifier.

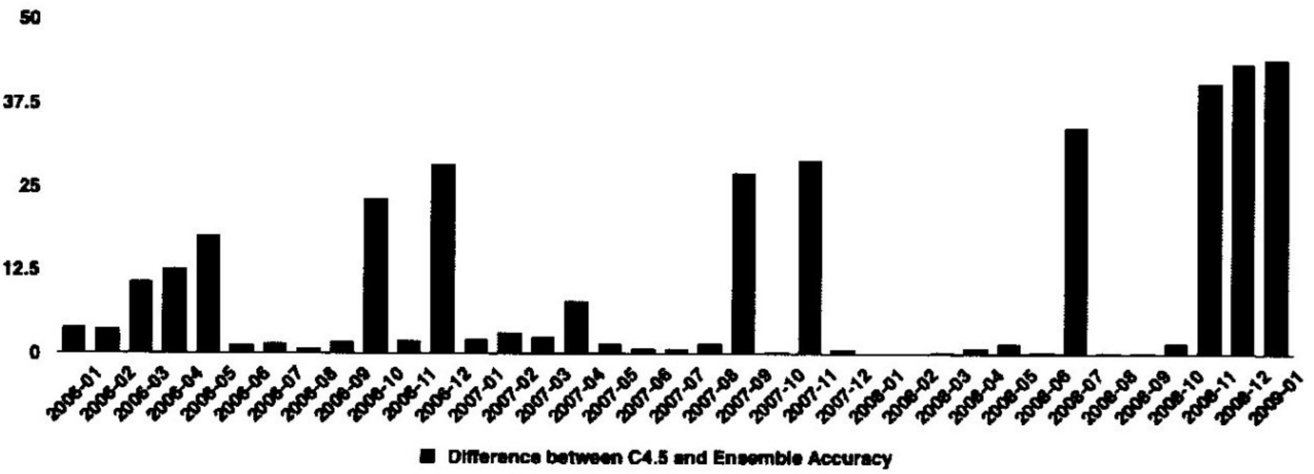

Figure 33 - Difference between clustering and ensemble accuracy.

Based on it and adding the aspect that a players' behavior tends to be a Niche behavior, we propose an assumption: a small observation window provides a small chance to identify the Niche stage. In order to evaluate that three experiments were done.

1. An observation window with only the six last months.
2. An observation window with only the 12 last months.
3. A progressive observation window without a limit size.

The months range is from 2006-01 to 2009-01. All experiments started in the first month. Each iteration consists in applying the method to the current month until the commitment prediction (without doing the Risk Computation).

For the first experiment (six months size), a total of 31 iterations were applied (from the range 2006-01/2006-06 to 2008-08/2009-01). For the second experiment (12 months size), a total of 25 iterations were applied (from the range 2006-01/2006-12 to 2008-02/2009-01). The third experiment ran 37 iterations for all months. We chose to define two degrees of Niche: Niche1 and Niche2. Niche1 occurs when the number of high committed players (HC) is greater than the number of average (AC) and low committed (LC) ones summed up, HC > (AC + LC). Niche2 occurs when HC > LC. We advocate that Niche1 represents a greater risk than the Niche2, because a majority of high committed players represents the end of the game usage lifecycle (Cook [2007] and Kummer et al. [2016]). Figures 34, 35 and 36 show the quantities of Niche months identified per iteration for each experiment:

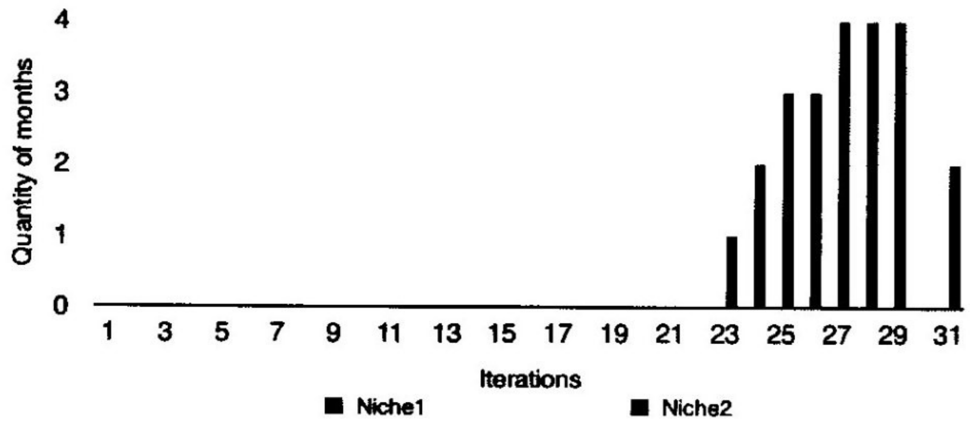

Figure 34 - The six months experiment result (quantity of Niches month identified by iteration).

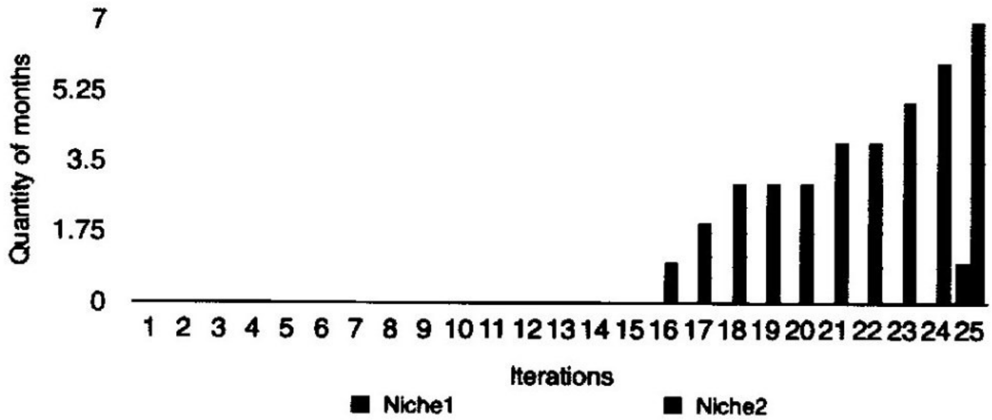

Figure 35 - The 12 months experiment result (quantity of Niches month identified by iteration).

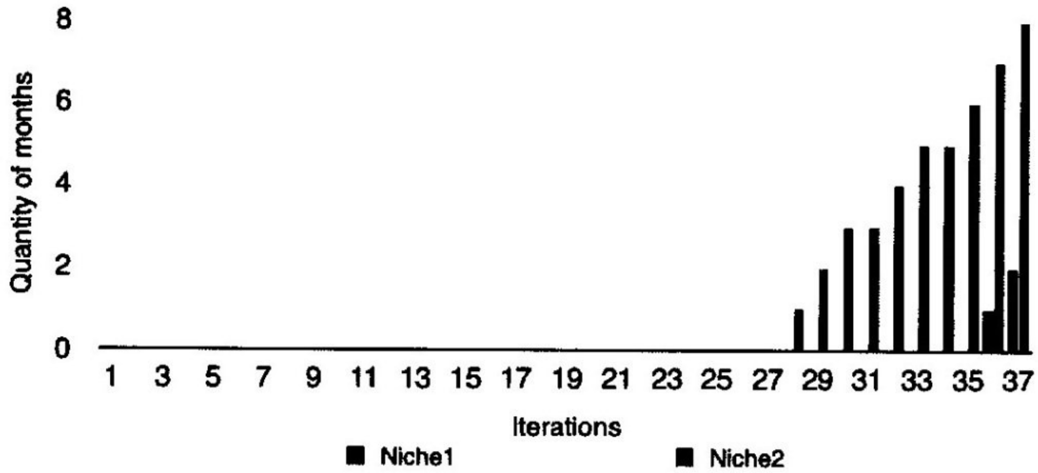

Figure 36 - The progressive experiment result (quantity of Niches month identified by iteration).

The first experiment identified less Niche months than the others, it did not identify any Niche1. The second experiment has a result similar to the third one. The Niche1 started to be identified. The third experiment identified more Niche months than the others. With these results we conclude that the longer is the observation window bigger are the chances to identify Niche, because it better contemplates the tendency of the players' behavior change to a Niche behavior. As a more detailed analysis, we extracted the last six months to compare them (illustrated in Table 7).

Table 7 - Observation window comparison.

Last six month	Progressive		Six months		12 months	
Month	Niche1	Niche2	Niche1	Niche2	Niche1	Niche2
2008-08		X				
2008-09		X				X
2008-10						
2008-11		X				X
2008-12	X	X		X		X
2009-01	X	X		X	X	X

As shown in Figures 34, 35 and 36, the 12 months experiment is very similar to the progressive one. The only disagreement was in the 2008-08, where it was a Niche2 for the progressive perspective and nothing for the 12 months one. But, some similarities also exist in all experiments: the last two months were identified as a Niche2, it means that even in a six month observation window, these months showed a very distinct behavior that enable Niche to be identified. Another situation was the month upgrade (2008-10), where the Niche was not identified in any perspective.

6.5. Risk Computation

The last step of the proposed method is the Risk Computation. At this moment, all data is labeled by the ensemble, it means that each player has a correspondent commitment degree associated to each time-span that he played. To compute the Risk Indicator (RI) Equation 5 is applied. After that, each time-span (month in this case) has a value between 0 and 1, considering 1 as a good commitment and 0 as a bad one. The 0 is labeled to the worst RI identified and 1 to the best. The RI measure the growth or drop of commitment from one month to the next, therefore, an

RI of a month represents the evolution or not of the motivational usage from the previous month to the actual one. In cases of game upgrade, where the MAU grows, the correspondent RI is shown next month, after computing the changes on commitment. The final result is shown in Figure 37.

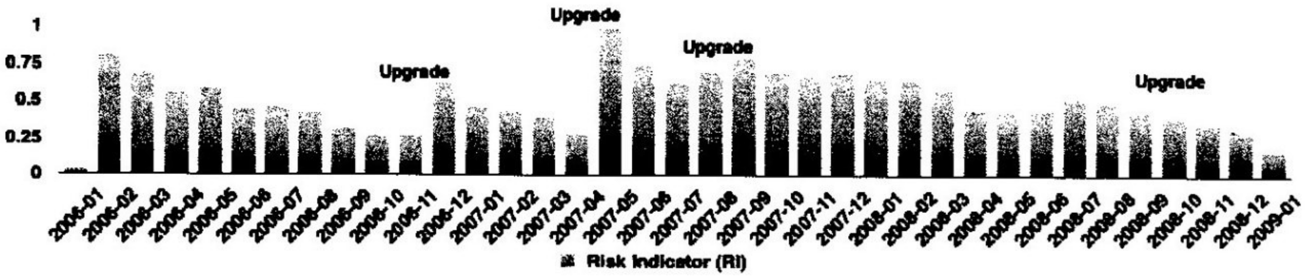

Figure 37 - Risk Indicator for WOWAH data set.

The worst month identified was the first one, with an RI of 0, and the best one was 2007-05. It is important to highlight that the RI represents an approach of the motivational usage, the greater the RI value, more motivated are the users. Comparing that RI with the classic MAU (Figure 38) some conclusions can be taken:

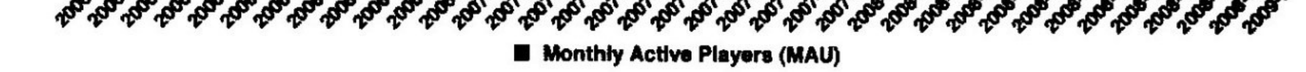

Figure 38 - MAU for WOWAH data set.

In the WOWAH data set, four main upgrades were applied in: 2006-11, 2007-04, 2007-08 and 2008-10. In the first three of them the RI follows the growth of MAU, but it does not happen in the last upgrade (2008-10), where the MAU grown and the RI continued to drop. In that case the RI illustrated that the game upgrade was not good enough to increase the motivational usage, because the commitment of its players continued to drop. As a similar analysis done to the Niche identification, using 2008-10 as an example, the MAU point of view identified a good situation in 2008-10, 2008-11 and 2008-12, because these MAU values were greater than the MAU value before the upgrade (2008-09), but in the RI view, 2008-10, 2008-11 and 2008-12 did not show any growth of commitment, showing that the upgrade did not affect the motivational usage.

The M5P regressor used to generate the Risk Indicator obtained as a result a correlation value of 0.9982 in a 10-fold cross-validation. The advantage of using an algorithm besides only the Equation 5 is the possibility to use its generated Equation to compute interval values. Equation 7 was generated by M5P algorithm.

$$RI_{M5P} = (0.0017 * LA) + (0.0015 * AH) + 0.0085 \qquad (7)$$

Analyzing this equation is possible to identify that the M5P algorithm considered as more relevant to predict the Risk Indicator the values that represent the growth of commitment: "low to average" (LA) and "average to high" (AH). If the decision-maker wants to know for a specific date the correspondent RI, he or she just needs to compute Equation 7 with the respective values. This Equation is not fixed, it can change as more data are included in the method.

Chapter 7.
Conclusions and Future Work

We proposed in this dissertation a new measure called Commitment, which represents how attached a player is to a game, given the idea of "how a player plays", differing from the classic MAU, where the idea consists in "how many players play". The new measure was obtained through a Data Mining approach, where the player behavior was clustered and then classifiers were created to represent each distinct time-span behavior. The observation window proportioned by the ensemble enabled labeling the players based on the historical behavior. The commitment measure allowed new analysis about the usage lifecycle, such as the quantities of players on each commitment degree and the Niche stage detection. New commitment metrics were computed based on the commitment measure, and based on the changes of these metrics the risk indicator was built.

The proposed method systematically generates a risk indicator. As a final result, it gives to game producers a new view about their game usage lifecycles. The procedure related in the previous chapter allowed us to validate the research hypothesis.

H 1)

That it is possible to extract commitment data based on game usage data.

Yes, it is. The clustering step can label commitment profiles of the players based on time spent and scores achieved.

H 1.1)

That it is possible to induce a model influenced by commitment data.

Yes, it is. Based on the cluster data classifiers can be induced to predict the commitment profiles.

H 1.1.1)

After the induction of the model, it can infer good and risk situations.

Yes, it is. Both, Risk Indicator (regressor) and the quantity of players on each commitment degree (ensemble) can identify when risk situations exist, for example when an unsuccessful upgrade occurs.

H 1.1.2)

A model with commitment metric presents a better assertiveness in identifying risk situations than MAU.

In the case of WOWAH dataset yes, it is. The Niche detection and the Risk Indicator identified in an earlier way than MAU when a risk situation is occurring. Taking the final upgrade on the WOWAH data set as an example, both Niche detection and Risk Indicator identified the unsuccessful upgrade two months before the MAU does.

One of the most important aspect about the original data is that they did not have the characteristics to predict the commitment, being necessary a previous preprocessing step. With the month granularity, new attributes were identified, allowing the commitment extraction, because a view of how the player plays (quantity of levels and max level) and time spent playing (quantity of days played) was created. This analysis allowed to label players in three degrees of commitment: low, average and high. These behaviors were associated according to the behaviors present in the literature, combining the conceptual with the experimental (Section 5.2).

It was identified the importance of using an observation window, because a view over the all historical data provides more information than just a part of it. An example of that is the Niche month detection, where the longer the observation window, more Niche months are detected.

The proposed Risk Indicator allowed a deeper view about the usage. The classic MAU idea provides information about "*How many players are playing my game?*" but it does not give the information of "*How they are playing in its motivational side?*". This motivational question can be answered by the Risk Indicator, which is based on the following concept: if a player likes the game, he or she will play for more time and will improve his or her score (max level in the case of WoW). Applying it to the model, it means that a player who likes the game will increase his or her commitment degree (e.g., low to average, average to high). In the same way, the Risk Indicator can detect drops of motivation too (e.g., average to low).

We understand that the proposed Risk Indicator must be used together with other metrics as MAU and profitability, because a motivational measure is just a part of a bigger context, that involves the quantity of players and profit. The main benefit of the Risk Indicator is to show to the decision-makers how attractive the game is to its active players, if the game content is not fully consumed, and if a new content pleased the players.

As future work, we intend to apply this proposed method to other games and compare the results. In order to do that, we will search and collect new usage data using new sources, as the

League of Legends API. Another point of interest consists in evaluating the changes of the main rules of Decision Trees over time (these rules describe changes in player behavior). This study can create rules which define low, average and high committed players over time.

The lifecycle stages are also an object of future work. In this dissertation we could predict only the Niche stage. A deeper study can provide new ways to predict the other stages. To do that, new metrics can be obtained from usage data and clustering techniques can be applied to identify some behavior tendencies that lead to other stages. New datasets of usage data can be built and then an automatic method can be developed to show to game producers the actual stage of their game based on the historical behavior of other games.

References

ALEXANDER, L., 2007. NPD: More Kids Playing Games, PC Play Dominates.
[Visited in: 2015; Available at:
http://www.gamasutra.com/view/news/106832/
NPD_More_Kids_Playing_Games_PC_Play_Dominates.php]

CASTRO, E. G. and TSUZUKI, M. S.G., 2015. Churn Prediction in Online Games Using Players' Login Records: A Frequency Analysis Approach, IEEE Transactions on Computational Intelligence and AI in Games, v. 7, n. 3, 255-265.

CHEN, K., PAO, H. K., CHANG, H., Game bot identification based on manifold learning. In: Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games. ACM, 2008. p. 21-26.

CHO, H. and KIM, K., Comparison of human and AI bots in StarCraft with replay data mining. In: Computational Intelligence in Games (CIG), 2013 IEEE Conference on. IEEE, 2013. p. 1-2.

COOK, D., 2007. The Circle of Life: An Analysis of the Game Product Lifecycle.
[Visited in: 2015; Available at:
http://www.gamasutra.com/view/feature/129880/the_circle_of_life_an_analysis_of_.php]

CORREA, B. D. and PASTOR, T. D., A evolução ao das técnicas de Inteligência Artificial. 2012.

DRACHEN, A., SIFA, R., BAUCKHAGE, C., and THURAU, C., 2012. Guns, swords and data: Clustering of player behavior in computer games in the wild. In 2012 IEEE conference on Computational Intelligence and Games (CIG) (pp. 163-170). IEEE.

DRACHEN, A., THURAU, C., SIFA, R. and BAUCKHAGE, C., 2014. A comparison of methods for player clustering via behavioral telemetry. arXiv preprint arXiv:1407.3950.

FERREIRA, E., Games e imersão: a realidade híbrida como meio de imanência virtual

[Visited in: 2016; Available at:
http://www.academia.edu/529960/Games_e_imersão_a_realidade_h%C3%ADbrida_como_meio_de_imanência_virtual]

GALWAY, L., CHARLES, D. and BLACK, M., 2008. Machine learning in digital games: a survey. Artificial Intelligence Review, Volume 29, Issue 2, pp 123-161.

GARDA, M. B., 2013. Neo-rogue and the essence of rogue likeness. University of Lodz.

GRAFT, K., 2009. NetEase Revenues Rise On Strong Online Game Performance.
[Visited in: 2015; Available at:
http://www.gamasutra.com/view/news/114647/NetEase_Revenues_Rise_On_Strong_Online_Game_Performance.php]

HONEYTRACKS. 2012. HoneyTracks game analytics.
[Visited in: 2015; Available at:
http://pt.slideshare.net/TomSente/casualconnect2012-honeytracks-game-lifecycle-kpis]

HOSTETLER, J., DERESZYNSKI, E. W., DIETTERICH, T. G., & FERN, A. Inferring strategies from limited reconnaissance in real-time strategy games. arXiv preprint arXiv:1210.4880, 2012.

HSIEH, J. and SUN, C., Building a player strategy model by analyzing replays of real-time strategy games. In: Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on. IEEE, 2008. p. 3106-3111.

KUMMER, L.B.M., BINDER F., JUNIOR, S.B., NIEVOLA, J.C. and PARAISO E.C., 2016. Predição do Estágio de Nicho em Jogos RPG Massivos de Multijogadores utilizando o Comprometimento. Encontro Nacional de Inteligência Artificial e Computacional. Page:289-300.

LEE, Y. and CHEN, K., Is server consolidation beneficial to MMORPG? A case study of World of Warcraft. In: Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on. IEEE, 2010. p. 435-442.

LEE, Y. T., CHEN, K. T., CHENG, Y. M., and LEI, C. L. . World of Warcraft avatar history dataset. In: Proceedings of the second annual ACM conference on Multimedia systems. ACM, 2011. p. 123-128.

LEECE, M. and JHALA, A., Sequential pattern mining in starcraft: brood war for short and long-term goals. In: Workshop on Adversarial Real-Time Strategy Games at AIIDE Conference, AAAI, AAAI (Raleigh, NC, 2014). 2014.

LEWIS, J. M., TRINH, P. and KIRSH, D., A Corpus Analysis of Strategy Video Game Play in Starcraft: Brood War. In: CogSci. 2011.

LUDGATE, S., 2011. MMO Chat: Scott Hartsman.
[Visited in: 2015; Available at:
http://www.gamasutra.com/blogs/SimonLudgate/20110707/89762/MMO_Chat_Scott_Hartsman.php]

MOORE, G. 1995. Crossing the Chasm. First edition. HarperCollins Publishers.

PINGEN, G. LJ., An implementation of a data-mining approach to strategy selection in Starcraft: Brood War. 2014. Doctorate Thesis. Radboud University Nijmegen.

ROBERTSON, G. and WATSON, I. D., An Improved Dataset and Extraction Process for Starcraft AI. In: FLAIRS Conference. 2014.

SALEN, K. and ZIMMERMAN E., 2003. Rules of Play. Cambridge/Massachusetts: The MIT Press.

SHEFFIELD, B. and ALEXANDER, L. 2008. Focus On Korea: T3's Kim Talks Hitmaking With Audition.
[Visited in: 2015; Available at:
http://www.gamasutra.com/view/news/107795/Focus_On_Korea_T3s_Kim_Talks_Hitmaking_With_Audition.php]

SILVER, D., HUANG, A., MADDISON, C.J., GUEZ, A., SIFRE, L., DRIESSCHE, G.V.D., SCHRITTWIESER, J., ANTONOGLOU, I., PANNEERSHELVAM, V., LANCTOT, M., DIELEMAN, S., GREWE, D., NHAM, J., KALCHBRENNER, N., SUTSKEVER, I., LILLICRAP, T., LEACH, M., KAVUKCUOGLU, K., GRAEPEL, T. and HASSABIS, D., 2016. Mastering the game of Go with deep neural networks and tree search. Nature 529, 484– 489.

SPELLER, T. H., 2012. The business and dynamics of free-to-play social-casual game apps. Master thesis, MIT.

STEWART, J., 2007. Different Approaches to Implementing Player Centered Game Design: A Comparison Study.
[Visited in: 2015; Available at:
http://www.researchgate.net/publication/249693949_Different_Approaches_to_Implementing_Player_Centered_Game_Design_A_Comparison_Study]

SYNNAEVE, G. and BESSIERE, P., A Dataset for StarCraft AI & an Example of Armies Clustering. arXiv preprint arXiv:1211.4552, 2012.

TAN, P., STEINBACH, M. and KUMAR, V., Association analysis: basic concepts and algorithms. Introduction to data mining, p. 327-414, 2005.

TARNG, P., CHEN, K. and HUANG, P., On prophesying online gamer departure. In: Proceedings of the 8th Annual Workshop on Network and Systems Support for Games. IEEE Press, 2009. p. 16.

UM, S., KIM, T. and CHOI, J., Dynamic difficulty controlling game system. IEEE Transactions on Consumer Electronics, v. 53, n. 2, 2007.

WANG, Q. and MAYER-SCHONBERGER, V., The monetary value of virtual goods: An exploratory study in MMORPGs. In: System Sciences (HICSS), 2010 43rd Hawaii International Conference on. IEEE, 2010. p. 1-11.

WEBER, B. G. and MATEAS, M., A data mining approach to strategy prediction. In: Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on. IEEE, 2009. p. 140-147.

WITTEN, I.H. and FRANK, E. 2005. Data Mining – Practical Machine Learning Tools and Techniques, 2nd Edition, Morgan Kaufmann Publishers.

ZHU, L., LI, Y. and ZHAO, G., Exploring the Online-Game Life Cycle Stages. In: E-Business and E-Government (ICEE), 2010 International Conference on. IEEE, 2010. p. 2436-2438.