

# Automatic Linguistic Feedback in Computer Games

Clemente Rubio-Manzano<sup>1</sup> Gracian Trivino<sup>2</sup>

<sup>1</sup>Department of Information Systems, University of the Bío-Bío, Chile

<sup>2</sup>European Centre for Soft Computing, Spain

## Abstract

This paper presents a new technology for automatically generating linguistic reports and immediate feedback from actions performed by players during play sessions. These reports allow us to provide players with a more complete and personalized feedback about their behaviors, abilities, attitudes, skills and movements. In order to show and explore the possibilities of this new technology, we incorporate and implement this model into the core of a single computer game.

**Keywords:** Linguistic Description of Complex Phenomena, Computer Games, Player Experience

## 1. Introduction

Current computer games provide players with some feedback about their play sessions. This feedback, from classical scoreboards to sophisticated head-up displays, is used for improving the motivations, training, learning and immersion of players [10]. Additionally, current computer games provide researchers and designers with feedback about players behavior by using data analysis techniques [5, 7]. This feedback is used for improving the design and implementation of computer games since provides them with expert knowledge about play sessions and with a better understanding of player behavior at different levels.

Most of feedback provided by current computer games is not enough when the objective is to deliver actually effective feedback about player behavior. In fact, feedback provided by data analysis tools is employed by designers or researchers at testing and post-production phases but usually is not delivered to the players during play sessions.

A more complete feedback about player behavior should provide players with more immediate and complete knowledge about their way of playing, main errors, attitudes or skills. Also, a feedback of this kind should provide designers and researchers with more knowledge about all the relevant events occurred during play sessions.

In this paper we propose a novel technology, specialized for computer games, which is based on monitoring and analyzing the actions of actors (players and agents) in game worlds. This allows us to perform linguistic descriptions from events occu-

red between different actors (players and agents) by remarking the most relevant features and hiding the irrelevant ones happened during play sessions (see Figure 1). This technology provides players, designers and researchers with automatic and more complete feedback about player's behavior by using linguistic reports. These reports can be easily included as a narrative component into the user interface.

The structure of the paper is as follows. Section 2 introduces the general concepts regarding computer games. Then, in Section 3 we propose a basic architecture for generating automatic linguistic feedback in computer games. Section 4 details the linguistic modeling of the architecture previously presented. Section 5 describes the report templates for customizing a final player report. Afterwards, Section 6 explains the experimentation carried out. Finally, Section 7 provides some concluding remarks.

## 2. Preliminary concepts

This section deals with introducing the reader some background information.

- **Computer Game.** A Computer Game is an interactive system formed by three components [1]: Input-Output devices (controllers and interface); Game (objectives and rules of the game, relationship between different elements) and Program (how the game is implemented at the code level). The player plays the game using the controllers and interface. The game provides player with immersive and pleasure experiences.
- **Game World.** A game world is the environment in which the game takes place. It is formed by either static or active elements.
- **Actor.** An actor is an entity that exists within the game world and that can interact with the elements of the game world, and with other actors. The mechanism of interaction is fixed by means of the rules of the game.
- **Player.** A player is an actor whose movements are performed by human players.
- **Agent.** An agent is an actor whose movements are performed by a virtual player, i.e., a specialized software.
- **Player Experience (PX).** PX is the overall impression and feeling of the game that a player

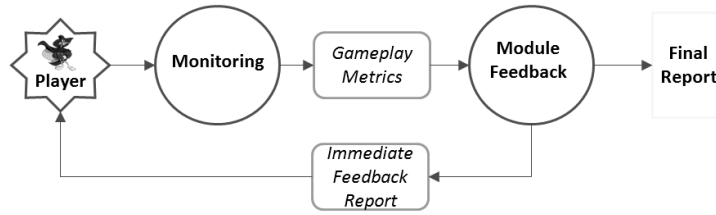


Figure 1: General diagram for automatically generated linguistic reports

has when he/she interacts with others actors in the game world. According to [4] the elements of PX are: motivation (interest and reward), meaningful choices (strategy and tactics), balance, usability (feedback) and aesthetics. PX must be measured in order to ensure that players have the expected experiences playing the game. Several methods have been proposed for measuring and evaluating playability and usability: Survey-based or interviews; Audio-visual recording of the game sessions; Biofeedback measures [6]. Recently, Gameplay metrics have been proposed as an analysis tool for usability and playability testing that offers insights into how people are actually playing the games under evaluation [2].

- **Gameplay metrics.** Gameplay Metrics are related to User-Initiated Events (UIEs) which refer to events that occur when a user interacts with the computer game. UIEs is a user-centered methodology of Human Computer Interaction that offers insight into how to create usable interfaces [9]. Gameplay metrics also include events taken from the computer game, for example the behavior of computer-controlled agents. By combining this gameplay metrics with traditional user experience measures (biofeedback, surveys and usability methods) it is possible to directly link player experience with design elements [12].
- **YADY Computer Game:** YADY (“Your actions define you”) is a single 2D game computer prototype written in Java. The main input-output devices of YADY are mainly the keyboard and the screen, respectively. The game consists in a player that must capture a set of pills distributed by a 2D scenario (the game world) by avoiding both obstacles and opponent. The game successfully finishes when the player captures all the pills. By contrast, the game over is produced when the player is hunted by the opponent.

### 3. Basic Architecture for generating Automatic Linguistic Feedback in Computer Games

Automatic Linguistic Description of Complex Phenomena (LDCP) aims to extract and represent knowledge by using natural language sentences (re-

port) as if they were produced by a human expert, describing the most relevant aspects of a phenomenon for certain users in specific contexts.

LDCP is based on the Computational Theory of Perceptions [14] and it has been used in previous works to generate, for example, linguistic description of surface of Mars [11] and linguistic reports assessing the results of simulation sessions in a driver trainer [3].

The basic architecture is based on the concept of Computational Perception (CP). A CP takes as values the elements of a set of linguistic expressions (words or sentences in natural language) that represents all the possible values of the perception (e.g.: “The current situation is { dangerous, safe, easy, risky }”).

We apply this methodology to generate linguistic feedback for players during play sessions. The result is a computational system able to collect and interpret events performed by actors (players and agents) in the game world during play sessions, yielding reports in natural language about their behaviors, abilities, attitudes, skills or movements. With that aim, the generated reports are customized to include information that is relevant for players and even for designers and researchers.

The basic architecture for automatic linguistic feedback is described as follows.

#### 3.1. Data structures

The main data structures of this architecture are, namely, the Gameplay metrics, the Granular Linguistic Model of Complex Phenomena (GLMP), and the Report Template (see Figure 2):

- **Gameplay Metrics.** The designer must define the gameplay metrics by analyzing which describe the actions of players during play sessions. The goal is to find out the best set of gameplay metrics in order to describe the most relevant events occurred during play sessions.
- **GLMP.** A Granular Linguistic Model of a Phenomena for automatic linguistic feedback (explained in detail on Section 4) is a computational structure which organizes all related perceptions in a similar way as either designers or players usually organize their experience in computer games by means of natural language. In particular, the designer must de-

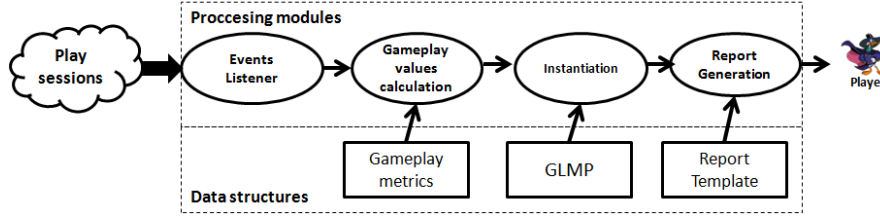


Figure 2: Main components of the proposed computational system for automatic linguistic feedback during play sessions

fine his/her subjective perceptions as computational perceptions. He/she uses the resources of the computer, e.g., action listeners, to acquire data about events and interactions produced by actors in the game world (gameplay metrics). Next, he/she uses his/her own experience to interpret these data and to create a network of computational perceptions (see Figure 3). Each computational perception covers a specific aspect of the events occurred with certain granularity degree. Defining this network is an iterative process that starts with the definition of the top-order computational perceptions. Here, it is a matter of answering general questions about how was a particular session and which was the behavior of the players. Here we have to take into consideration several features: (i) the kind of situations produced between the player and the opponent, (ii) the kind of attitude showed by the player in a particular situation, (iii) kind of movements performed by the player and (iv) needed time in capturing the pills distributed on the scenario. From these features we could answer one important question: which is the ability and the skill level for the players during play sessions.

- **Report Template.** The designer analyzes the particular meaning of each linguistic expression in specific situation types to build the Report Templates. A Report Template is defined considering the application requirements to deliver feedback to players about their actions. In particular, the designer must create in collaboration with human factor experts a document which summarizes general observations about movements, attitudes, skills, speed, abilities and situations. These observations will be written down in natural language and will be incorporated into the User Interface as visual components (see Section 5).

### 3.2. Processing modules

The main processing modules of this system are, namely, the Events Listener module, the Gameplay Values Calculation module, the Instantiation module and the Report Generation module (see Figure 2):

- **Events Listener Module.** The system captures the positions and their variations for each actor (player or agent) in the game world by using events listeners.
- **Gameplay Values Calculation Module.** From these positions the system computes the values for each gameplay metric defined in the Gameplay metrics data structure.
- **Instantiation Module.** This module uses the metrics to generate an instance of the GLMP. (see Section 4).
- **Report Generation Module.** The Report is obtained from input data as result of instantiation processes of the Report Template, i.e., the computational system selects (instantiates), among the available possibilities, the most suitable linguistic expressions in the GLMP to describe the input data.

## 4. Granular linguistic Model

This section describes the granular linguistic model for YADY players behavior which is shown in Figure 3.

### 4.1. First-order Computational Perceptions

A special type of computational perception is used to process the input data, we call them first order computational perception (1CP). In our case the inputs are numerical data obtained from events captured and logged during play sessions. We implement our GLMP model from the values associated to these gameplay metrics.

#### 4.1.1. CP of Distance

The distance is the number of cells among the different actors (see Figure 3-right). This is a gameplay metric whose values are represented by a real number. Given two points  $A(x, y)$  and  $B(x', y')$ , the distance between both points is calculated by means of the following equation:  $\sqrt{(x - x')^2 + (y - y')^2}$ .

We define three different CPs of Distance: Distance between the player and the opponent ( $D_{po}$ ); Distance between the player and the closest pill with respect to him ( $D_{pc}$ ) and Distance between the opponent and the closest pill with respect to the player ( $D_{oc}$ ). Since these parameters are real

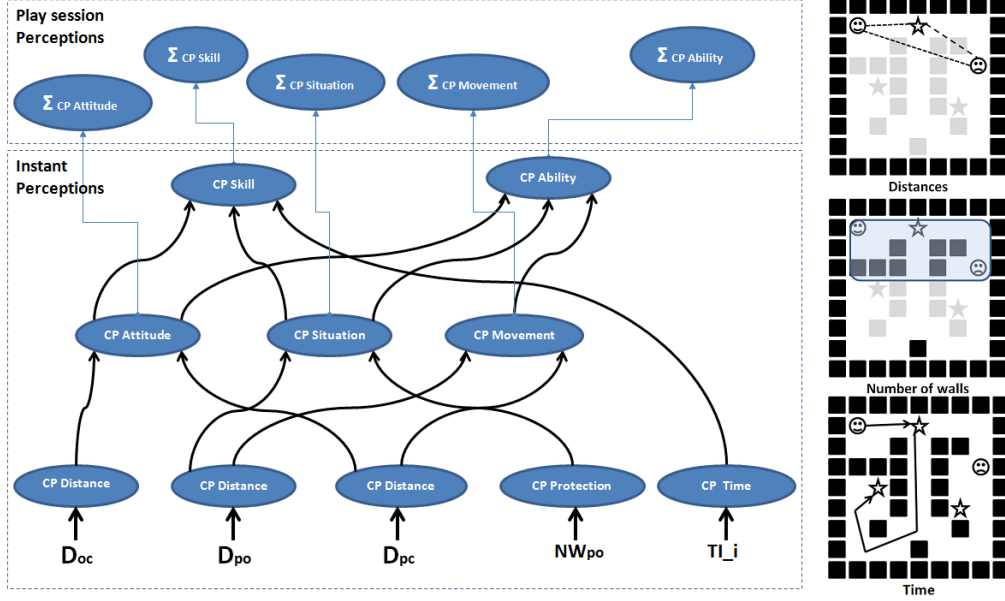


Figure 3: GLMP (left) and Gameplay metrics (right).

numbers, we define a first-order CP as follows: CP-Distance=  $((close, far), (\mu_{close}, \mu_{far}))$ , where  $\mu_{close}, \mu_{far}$  are fuzzy subsets defined on  $U = [0, N]$ , being  $N$  the size of the game world.

#### 4.1.2. CP of Protection

The protection degree is the number of walls between the player and the opponent (see Figure 3-right). This is a gameplay metric whose values are represented by a real number. Given two points  $A(x, y)$  and  $B(x', y')$ , being the positions of the player and the opponent, respectively. The number of obstacles between both points can be calculated by using the following steps: i) Calculating the rectangle formed by both points; ii) Counting the number of obstacles in such a rectangle. Since this parameter is a real number, we define this first-order CP as follows: CP-Protection=  $((low, high), (\mu_{low}, \mu_{high}))$ , where  $(\mu_{low}, \mu_{high})$  are fuzzy subsets defined on  $U = [0, N]$ , being  $N$  the total number of walls in the game world.

#### 4.1.3. CP of Time

This CP measures the spent time to capture the pills distributed in the game world (see Figure 3-right). Given two points  $A(x, y)$  and  $B(x', y')$ , the time is calculated by measuring the needed time to go from  $A$  to  $B$ . For example, suppose that the player is at the position  $(0, 0)$  and the pills are at the positions  $(3, 4)$  and  $(8, 8)$ . CP-Time measures the time required to go from  $(0, 0)$  to  $(3, 4)$  and then from  $(3, 4)$  to  $(8, 8)$ . Since the parameter is a real number, we define a first-order CP as follows: CP-Time=  $((large, short), (\mu_{large}, \mu_{short}))$ , where  $\mu_{large}, \mu_{short}$  are fuzzy subsets defined on

$U = [0, N]$ , being  $N$  a maximum fixed in seconds for capturing a coin.

## 4.2. Second-order Computational Perceptions

We call second order computational perception (2CP) to those explained by previous CPs. Figure 3 shows a Granular Linguistic Model for generating automatic linguistic feedback in which 1CPs are obtained from Gameplay Metrics and 2CPs are calculated based on subordinate CPs, providing information about play sessions at different levels of abstraction.

### 4.2.1. CP of Situation

The CP-Situation is obtained by aggregating CP-Protection and CP-Distance (CP- $D_{po}$ ). This CP is defined as follows: CP-Situation=  $((Risky, Safe, Dangerous, Easy), f(CP-Protection, CP-D_{po}))$ ; where  $f$  is an aggregation function implemented by a set of fuzzy rules. Here, CP-Situations are calculated by using the Mamdani-type rules [8] expressed in the Table 1. Each row must be read as follows: “IF CP-Protection is Low and CP- $D_{po}$  is Close THEN CP-Situation is Dangerous”.

Table 1: Rules for CP of Situation

CP-Situation	CP-Protection	CP - $D_{po}$
Risky	High	Close
Dangerous	Low	Close
Safe	High	Far
Easy	Low	Far

### 4.2.2. CP of Attitude

The CP-Attitude is obtained by aggregating CP-Distance between the player and the closest coin

(CP- $D_{pc}$ ) and CP-Distance between the opponent and the closest coin (CP- $D_{oc}$ ). This CP is defined as follows: CP-Attitude=((Wise, Brave, Cautious, Passive),  $f(\text{CP-}D_{pc}, \text{CP-}D_{oc})$ ). Here, CP-Attitudes are calculated by using the rules expressed in the Table 2:

Table 2: Rules for CP of Attitude

CP-Attitude	$CP - D_{pc}$	$CP - D_{oc}$
Wise	Close	Far
Brave	Close	Close
Cautious	Far	Close
Passive	Far	Far

#### 4.2.3. CP of Movement

The CP-Movement is obtained by aggregating 1CP-Distance between the player and the closest coin (1CP- $D_{pc}$ ) and 1CP-Distance between the player and the opponent (1CP- $D_{po}$ ). This 2CP is defined as follows: CP-Movement=((Good, Bad, Scare, Kamikaze),  $f(1\text{CP-}D_{pc}, 1\text{CP-}D_{po})$ ). Here, 2CP-Movements are calculated by using the rules expressed in the Table 3.

Table 3: Rules for 2CP of Movements

2CP-Movement	$1CP - D_{pc}$	$1CP - D_{po}$
Good	Close	Far
Scare	Far	Far
Kamikaze	Close	Close
Bad	Far	Close

#### 4.2.4. CP of Skill

CP-Skill is obtained by aggregating CP-Attitude, CP-Movement and CP-Time. This CP is defined as follows: CP-Skill=((Expert, Intermediate, Basic, Beginner),  $f(\text{CP-Attitude}, \text{CP-Movement}, \text{CP-Time})$ ). Here, CP-Skills are calculated by using the rules expressed in the Table 4 (for simplicity only a subset of them are shown).

Table 4: Subset of rules for CP of Skill

CP-Skill	CP-Attitude	CP-Movement	CP-Time
Expert	Wise	Good	Short
Intermediate	Brave	Good	Short
Basic	Passive	Bad	Large
Beginner	Passive	Scare	Large

#### 4.2.5. CP of Ability

The CP-Ability is obtained by aggregating CP-Attitude, CP-Movement and CP-Situation. This CP is defined as follows: CP-Ability=((Skillful, Somewhat-skillful, Somewhat-clumsy, Clumsy),  $f(\text{CP-Attitude}, \text{CP-Movement}, \text{CP-Situation})$ ). Here, CP-Abilities are calculated by using the rules expressed in the Table 5 (for simplicity only a subset of them are shown).

Table 5: Subset of rules for CP of Ability

CP-Ability	CP-Attitude	CP-Movement	CP-Situation
Skillful	Wise	Good	Easy
Somewhat-skillful	Cautious	Good	Safe
Somewhat-clumsy	Brave	Bad	Dangerous
Clumsy	Passive	Bad	Risky

### 4.3. Linguistic summaries

In order to provide information about the whole play session, we calculate the summaries of each instant 2CP. The process consists in adding the values of a particular instant 2CP during the play session by using the concept of fuzzy cardinality. We call these CPs, Play session CPs ( $\Sigma CP$ ). We use these  $\Sigma CP$  to generate the report.

Note that we could improve this sentences by using ,e.g, linguistic summaries (in the sense of Yager [13]) to resume the information provided for the play session CPs. Giving a 2CP, each sentence in the linguistic summary has the form “ $q$   $s$  were  $v$ ”, where  $q$  is a linguistic label (nearly-none, few, several, many, nearly-all) of a quantifier  $Q$  on  $[0, 1]$ ;  $s$  represents the name (plural) of such 2CP and  $v$  is a qualifier value of such 2CP. For example, the following summaries can be obtained from  $\Sigma CP$ -Situation: “Many situations were safe; Few situations were easy;”.

### 5. Report Template

We have created a report template in order to provide players with the most relevant details about their YADY’s play sessions. Figure 4 shows a possible report template for the YADY computer game. Each  $v_{S_i}$  is an element of an array of strings  $S_i$  listed below.

$$S_1 := \{ \text{not}, \text{“void”} \}^1$$

$$S_2 := \{ \text{wise}, \text{brave}, \text{cautious}, \text{passive} \}.$$

$$S_3 := \{ \text{skillful}, \text{somewhat-skillful}, \text{somewhat-clumsy}, \text{clumsy} \}$$

$$S_4 := \{ \text{passed for risky}, \text{enjoyed of safe}, \text{passed for dangerous}, \text{enjoyed of easy} \}$$

$$S_5 := \{ \text{beginner}, \text{basic}, \text{intermediate}, \text{advanced} \}$$

Note that,  $v_{S_1} := \text{“not”}$  when the play session finished in game over and  $v_{S_1} := \text{“void”}$  when the play session successfully finished;  $v_{S_2}^1$  is the highest value obtained from  $\Sigma 2CP$ -Attitude and  $v_{S_2}^2$  is the second highest value obtained from  $\Sigma 2CP$ -Attitude;  $v_{S_3}^1$  is the highest value obtained from  $\Sigma 2CP$ -Ability;  $v_{S_3}^2$  is the highest value obtained from  $\Sigma 2CP$ -Situation and  $v_{S_4}^1$  is the second highest value obtained from  $\Sigma 2CP$ -Situation;  $v_{S_4}^2$  is the highest value obtained from  $\Sigma 2CP$ -Skill.

<sup>1</sup>“void” represents an empty string

Table 6: Beginner Player versus Advanced Player

The play session has been successfully completed. You have showed a <u>cautious</u> attitude during the most part of the play session, although sometimes you were a <u>passive</u> player. Most of movements evidence <u>clumsy</u> abilities. You passed for dangerous situations during the main body of the play session, although also you <u>passed for risky</u> situations. Your skill level is <u>beginner</u> .	The play session has been successfully completed. You have showed a <u>brave</u> attitude during the most part of the play session, although sometimes you were a <u>wise</u> player. Most of movements evidence <u>skillful</u> abilities. You <u>enjoy of safe</u> situations during the play session, although also you <u>passed for</u> risky situations. Your skill level is <u>advanced</u> .
---	--

The play session has  $v_{S_1}$  been successfully completed. You have showed a  $v_{S_2}^1$  attitude during the most part of the play session, although sometimes you were a  $v_{S_2}^2$  player. Most of movements evidence  $v_{S_3}^1$  abilities. You  $v_{S_4}^1$  situations during the main body of the play session, although also you  $v_{S_4}^2$  situations. Your skill level is  $v_{S_5}^1$ .

Figure 4: The Report Template for the YADY Computer Game

## 6. Experimentation

In order to obtain a first intuitive validation of our system for automatic linguistic feedback, we have checked it on the YADY computer game. Two different players played the game: a beginner player and an advanced player. Table 6 shows the final reports generated for each type of player. These reports show that the expert players enjoyed of a greater number of safe and easy situations, while the non-expert players often passed for risky and dangerous situations. Additionally, brave and wise attitudes are related with expert players, while passive and cautious attitudes are related with non-expert players. These general facts are identified by using the final descriptions generated about both skill levels and abilities.

## 7. Conclusions and future work

In this paper, we have shown a first prototype of a promising technology that could be used to improve the current feedback provided for computer games. This work is a first approximation towards a broader approach and much work remains to be done in this direction. The preliminary results show that the incorporation of this kind of feedback into the core of computer games could improve the overall gaming experiences.

## Acknowledgments

This work has been done in collaboration with the research group SOMOS (Software-MOdeling-Science) funded by the Research Agency and the Graduate School of Management of the Bío-Bío University under grant 130415 GI/EF during a research stay performed at the research unit “Computing with Perceptions” from the European Centre for Soft Compu-

ting. Finally, we want to express our gratitude to D. Sanchez-Valdes for his kind and useful comments.

## References

- [1] E.H. Calvillo-Gamez, P. Cairns, A.L. Cox. Assessing the core elements of the gaming experience. In *Evaluating user experience in games* (pp. 47-71). Springer London (2010).
- [2] A. Drachen and A. Canossa. Analyzing Spatial User Behavior in Computer Games using Geographic Information Systems. In *Proc. of the 13th International MindTrek Conference*, pp. 182-189, ACM (2009).
- [3] L. Eciolaza et al. Automatic linguistic reporting in driving simulation environments. *Applied Soft Computing* 13.9 (2013): 3956-3967.
- [4] J. Ferrara. The Elements of Player Experience. *UX Magazine*. Article No. 651. April 7 (2001).
- [5] J.H. Kim et al. Tracking Real-Time User Experience (TRUE): a comprehensive instrumentation solution for complex systems. In *Proc. of the SIGCHI conference on Human Factors in Computing Systems*, ACM, pp. 443-452 (2008).
- [6] N. Lazzaro. Why we play games: four keys to more emotion without story. *Player Experience Research and Design for Market Interactive Entertainment* (2004).
- [7] C. Lewis and N. Wardrip-Fruin. Mining game statistics from web services: a World of Warcraft armory case study. In *Proc. of Foundations of Digital Games* (2010).
- [8] E.H. Mamdami, S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Human Computer Studies* 51(2), pp. 169-196 (1999).
- [9] R.J. Pagulayan et al. *User-Centered Design in Games. The HCI handbook: fundamentals, evolving technologies and emerging applications* (2003).
- [10] M. Prensky. *Digital Game-based Learning*. McGraw-Hill, 2001.
- [11] D. Sanchez-Valdes et al. Linguistic description about circular structures of the Mars’ surface. *Applied Soft Computing* 13.12 (2013): 4738-4749.
- [12] A. Tychsen. Crafting User Experience via Game Metrics Analysis. *Proc. of the Workshop Research Goals and Strategies for Studying User Experience and Emotion at the 5th Nordic Conference on HCI* (2008).
- [13] R.R. Yager. Fuzzy Summaries in Database Mining. In *Artificial Intelligence for Applications*, IEEE Computer Society (1995).
- [14] Zadeh, L. A. (2001). From computing with numbers to computing with words. *Annals of the New York Academy of Sciences*, 929(1), 221-252.