Adaptive Behavior Control Model of Non Player Character

Petrenko Tetyana
Department of Computer Technologies
Donetsk National University
Donetsk, Ukraine
quay_149@yahoo.com

Tymchuk Oleg
Department of Computer Technologies
Donetsk National University
Donetsk, Ukraine
o.s.timchuk@gmail.com

***Abstract*—This paper presents the adaptive behavior control model of non player character. The model is based on the theory of control systems and the theory of discrete interval type-2 fuzzy sets and systems. Dynamic changes of the structure of fuzzy rule base with taking into account cognitive parameters of player are executed in the model in on-line mode. It allows to individualize gameplay and to provide interactive interaction of player and NPC. Cognitive parameters of player (gestures, facial expressions, emotions) are determined on basis on the Kinect output data. The quality of dynamic changes of structure of the model (self-organization) is controlled by criteria of videogame enjoyment.**

Keywords-computational intelligence; adaptive system; control system; videogame; non player character; discrete interval type-2 fuzzy set; discrete interval type-2 fuzzy system

I. INTRODUCTION

The development of the videogame industry actualizes the creation of new information technologies for the interactive realization of videogames [1]. Videogame developers more and more frequently integrate natural user interfaces (GUIs) into loop control of videogame. For example, Microsoft Kinect is typical sample of NUIs [2]. It provides a natural interface for managing gameplay as well as increasing degree of player freedom. It also changes nature of player interaction with computer characters (NPCs) during gameplay using NUIs tools. NPCs are treated as intelligent videogame objects, which are not controlled by player. Traditionally, behavior of NPCs is based on behavior trees. However, these methods of NPCs behavior modeling should be complemented with the possibility of analysis and consideration of cognitive parameters of player that are analysis of gestures, facial expressions and emotions.

The use of new devices and forms of interaction requires new software for creation of interactive gameplay.

NPCs are the primary videogames, so the videogame enjoyment depends on form and content of its interaction with players. The development of methods of controlling behavior of NPCs with analyzes of individual characteristics of player is actual during gameplay.

The paper presents the behavior control model of NPC in keeping with NUIs possibilities. The purpose of the behavior control model of NPC is to maximize the videogame enjoyment. These requirements can be realized only in view of cognitive parameters of player and learning of NPC in on-line mode.

II. ADAPTIVE BEHAVIOR CONTROL MODEL OF NPC

The behavior control system of NPC is a complex dynamic system. The main elements of such system are states in which NPC can be (for example, "Attack", "Escape"). At any time, NPC can be in only one of its states. Generally, each state can be complex and consist of sub-states (for example, the state of "Attack" may consist of the sub-states "hunger" and "aggressive"). Also, NPC can be found in two special states - the initial (a state in which NPC is just after its creation) and the final (a state in which NPC is right before destruction). Parameters which have influence on the sub-state of NPC will be called factors. Internal factors correspond to needs and nature of NPC (for example, the food needs, energy, etc.) and external factors correspond to nature of player (the cognitive parameters). NPC can change one state to another on the basis of the transition. An event, in this case, refers to reaction of external factors NPC at time. This reaction causes NPC transition from one state to another. The transition can also be reflexive. At any transition NPC performs an action, that is, it implements a continuous behavior as part of the transition. Action determines what happening with NPC, when NPC enters or leaves any state.

The cognitive parameters of player are uncertain. Traditional methods of computational intelligence, as well as methods of the theory of type-1 fuzzy sets and systems do not let us to take into account the uncertainty of this kind. Therefore, we propose to use the methods of theory of discrete interval type-2 fuzzy sets (DIT2FSs) and systems (DIT2FLSs) [3] to determine the cognitive parameters of player in this paper.

The paper proposes to use the principles of evolving intelligence [4] for creation of the behavior control system of NPC. Such system should have such basic features as cognition and adaptability.

The cognition is provided by analysis and consideration of cognitive parameters of player.

The process of constructing an adaptive fuzzy inference model consists of two phases: parametric adaptation and structural adaptation.

The parametric adaptation configures the forms and parameters of membership functions of linguistic variables that describe states of NPC, internal and external factors,

978-0-7695-4994-1/13 $26.00 © 2013 IEEE
DOI 10.1109/UKSim.2013.68

and NPC actions. Nowadays, there are numerous tools to solve this question. The gradient methods of optimization and optimization methods, which are based on random search or on evolutionary computation theory, are standard and well-studied. The parametric adaptation has a high computational complexity, so videogame developers decided this problem in off-line mode. The parametric adaptation doesn't hold in the paper.

The base of fuzzy rules describing the NPC behavior is forming with the structural adaptation. The structural adaptation (change / add / remove the fuzzy rules) can be viewed as a process of learning of NPC, and the fuzzy rules base as memory of NPC. In order to ensure interactive interaction of player and NPC, learning is carried out in on-line mode (operational learning) taking into account not only internal, but also external factors.

At each discrete moment it is not rational to carry out an assessment of videogame enjoyment and respectively the learning of NPS in on-line mode. It is connected with the limited resource of the PC hardware and it can break a continuous behavior control of NPC. According to the dynamic properties of NPC, we choose the maximum allowable sampling rate θ . Define two parameters associated with θ :

- $t = p\theta, p = 0,1,\ldots$ – period of gameplay update (discretization);
- $T = qt, q = 1, 2, \ldots$ – period between two events.

The structure of rules of the behavior control model of NPC is unchanged within the period T and the change is only possible at time-steps ($0, T, 2T, \ldots, kT, \ldots, L\theta = \Theta$), where $[0, \Theta]$ – period of the passed level of videogame by player.

Control systems' theory is used to describe the dynamic properties of NPC control system. According to control systems theory and theory of DIT2FSs and DIT2FLSs let's present a hybrid behavior control model of NPC

$$
\begin{aligned}
&\langle \widetilde{Y}(t) \rangle = F^{*}\left(\langle \widetilde{Y}(t\quad h) \rangle, IN(t\quad 1), F^{**}\left(\langle R(kT) \rangle\right)\right),\\
&x(t) = F\left(F^{*}\left(\langle \widetilde{Y}(t) \rangle, \langle Rstate(t) \rangle\right)\right),\\
&h = \overline{1,H}, IN(t) = \langle in_n \rangle, n = \overline{1,N},
\end{aligned} \tag{1}
$$

where $\langle \widetilde{Y}(t) \rangle$ – set of DIT2FSs, describing sub-states of NPC at time-step t,

IN(t) – input crisp points (factors),

N – the total number of factors which have influence on sub-states of NPC,

H – logical depth of history of sub-states of NPC (memory of NPC),

$\langle R(kT) \rangle$ – set of fuzzy rules which form sub-states of NPC,

F^{**} – function realizes dynamic changes of structure of fuzzy rule base,

F^{*} – Mamdani fuzzy inference system,

x(t) – action of NPC at time-step t,

$\langle Rstate(t) \rangle$ – set of fuzzy rules which form states of NPC at time-step t,

F – function realizes an action which is performed by NPC during transition (e.g. movement on game field on one position to left).

Assessment of learning is carried out by criteria of videogame enjoyment [5], which should not be below a given ε. Define criteria of videogame enjoyment as

$$
\begin{aligned}
&B(t) = F^{*}\left(IN_{GI}(t), \langle GI \rangle, LO_{GI}, \langle R_{GI} \rangle\right),\\
&IN_{GI}(t) = \langle e_j \rangle, j = \overline{1,J},
\end{aligned}
$$

where $IN_{GI}(t)$ – input crisp points,

J – the number of criteria of player satisfaction,

$\langle GI \rangle$ – set of linguistic variables describing the criteria of player satisfaction,

LO_{GI} – output linguistic variable describing the level of videogame enjoyment,

$\langle R_{GI} \rangle$ – set of fuzzy rules,

F^{*} – Mamdani fuzzy inference system.

Operational learning of NPC is executed when $B(t)<\varepsilon$.

$$
\langle GI \rangle = \langle GES, GIS, GLC \rangle,
$$

where GES – criteria describing emotional state of player,

GIS – criteria describing rate of change the player satisfaction,

GLC – criteria describing development level of player.

GES = {"funny", "still", "agitated", "surprised", "irritated"}

GIS = {"low", "middle", "high"}

GLS = {"local", "middle", "high"}

LO_{GI} = {"low", "middle", "high"}

Vector $IN_{GI}(t)$ activates the terms of linguistic variables GES, GIS and GLC. It is determined with the help of analyze of the cognitive parameters of player (e.g. cognitive player style, the active memory type of player, the dominant memory type of player, emotional state of player etc.). The cognitive parameters of player are determined on basis on data, which are obtained by the NUI (Kinect is used in this paper).

Lets consider the example of fuzzy rule of the set $\langle R_{GI} \rangle$

$$
\begin{aligned}
r:\ &\text{IF } e_1 \in \text{"player is irritated" AND}\\
&\quad e_2 \in \text{"high rate of change of player satisfaction" AND}\\
&\quad e_3 \in \text{"high level of player development"}\\
&\text{THEN "low videogame enjoyment"}
\end{aligned}
$$

Lets have a closer look at the process of structure's changing of fuzzy rule base of the model. The structure of fuzzy rule base $\langle R(kT)\rangle$ of the behavior control model of NPC at time-step kT is defined as

$$\langle R(kT)\rangle = \left\langle \left\langle R^{*}(kT)\right\rangle, \langle R(kT \quad h)\rangle \right\rangle, \qquad (2)$$

where $\langle R(kT-h)\rangle$ – set of fuzzy rules which were formed at time-steps $kT-h,\ h=\overline{1,H}$. $\langle R(kT-h)\rangle$ describe the experience of NPC,

$\left\langle R^{*}(kT)\right\rangle, R^{*}(kT) \notin \langle R(kT-h)\rangle$ – new set of fuzzy rules which were formed during operational learning of NPC.

The following types of fuzzy rules is distinguished in the theory of fuzzy sets and systems [6]: standard IF THEN rules, incomplete IF rules, mixed rules, fuzzy statement rules, comparative rules, unless rules, quantifier rules.

We can define the following types of fuzzy rules in (2):

- standard IF THEN rules

$$\begin{aligned} R_m^{(i)}: \mathrm{IF}\ in_1\ \text{ is }\ \widetilde{F}_1^{(i)}\ \text{ AND} \ldots \text{AND} \\ in_n\ \text{ is }\ \widetilde{F}_n^{(i)}\ \text{ AND} \ldots \text{AND} \\ in_N\ \text{ is }\ \widetilde{F}_N^{(i)} \\ \mathrm{THEN}\ \widetilde{\Psi}_m^{(i)}(v_m),\ m=\overline{1,M},\ i=\overline{1,I} \end{aligned},$$

- incomplete IF rules

$$\begin{aligned} R_m^{(i)}: \mathrm{IF}\ in_1\ \text{ is }\ \widetilde{F}_1^{(i)}\ \text{ AND} \ldots \text{AND} \\ in_n\ \text{ is }\ \widetilde{F}_n^{(i)}\ \text{ AND} \ldots \text{AND} \\ in_p\ \text{ is }\ \widetilde{F}_p^{(i)} \\ \mathrm{THEN}\ \widetilde{\Psi}_m^{(i)}(v_m),\ m=\overline{1,M},\ i=\overline{1,I} \end{aligned},$$

$$F_n^{(i)} \in LF_n,$$

$$in_n \in IN, IN = \left\langle in_1, \ldots, in_n, \ldots, in_p, in_{p+1}, \ldots, in_N \right\rangle,$$

where LF_n – linguistic variable defines factors; it is defined as the primary variable $X_n, X_n \in [0,1]$,

$\widetilde{F}_n^{(i)}$ – membership function (DIT2FS) of term of LF_n.

M – the total number of rules which were formed during operational learning of NPC at all time-steps,

I – the total number of states of NPS,

$\widetilde{\Psi}_m^{(i)}$ – DIT2FS defines i-th state of NPC; it is defined as the primary variable $Y_i, Y_i \in [0,1]$,

$v_m, v_m \subseteq [0,1]$ –weight of m-th rule $R_m^{(i)}$.

The left side of rules $\left\langle R^{*}(kT)\right\rangle$ (antecedents) is formed of all sets of activated terms of linguistic variables LF_n. The right side of each rule $R_m^{(i)}$ (consequent) of the set $\left\langle R^{*}(kT)\right\rangle$ is defined as

$$\widetilde{\Psi}_m^{(i)} = \bigcup_{n=1}^{N} \bigcup_{d=1}^{D} \alpha \widetilde{S}_d,\ \widetilde{S}_d \in LS_i,$$

$$\alpha = \mu_{F_n^{(i)}}(in_n) \cdot w_n,$$

$$\mu_{F_n^{(i)}}(in_n) = \left[\ \underline{\mu}_{F_n^{(i)}}(in_n),\ \overline{\mu}_{F_n^{(i)}}(in_n)\ \right],$$

$$\langle w_n \rangle^{(i)},\ w_n \subseteq [0,1],$$

where N – the total number of variables in m-th rule,

LS_i – linguistic variable defines the types of i-th sub-states of NPC; it is defined as the primary variable Y_i,

$\widetilde{S}_d$ – activated term of linguistic variable LS_i which is characterized by DIT2FS membership function,

D – the total number of activated terms for n-th antecedents of rule. Set of functions $\langle f_n : X_n \to Y_i \rangle$ is used to determine the activated terms of linguistic variable LS_i,

$\langle w_n \rangle^{(i)}$ – input crisp points define degree of influence of n-th factor on i-th sub-state of NPC.

The fuzzy rule base (2) (memory of NPC) is limited because of the limited available memory in computer. We assume that in the memory of NPC can be store not more than M rules. At each time-step kT is activated a deterministic number of rules. Depending on whether the rule was activated or not, the weight of rule changes as

$$v_m = \begin{cases} v_m + 1/M, & \text{rule is activated AND } v_m < 1 \\ v_m - 1/M, & \text{rule doesn't activated AND } v_m > 0 \end{cases}.$$

In the situation when memory of NPC will be depleted, then from $\langle R(kT)\rangle$ removes redundant rules, which have the lowest weight (effect of forgetting information).

Automatic construction of the fuzzy rule base can give problems that arise in an operation of model:

- problem of incomplete rules: undefined behavior of model, if no rule from fuzzy rule base doesn't work;
- problem of conflicting rules: there are rules with the same left parts in fuzzy rule base.

The solution of problem of incomplete rules consists in generation of new rule in the case of absence of fuzzy rule with a given set of antecedents in fuzzy rule base.

Algorithm is used to solve problem of conflicts with set of rules:

- the union of linguistically similar rules. We choose rules with the greater weights if they have difference only in it;
- the selection of an appropriate rule. Among the rules remaining after the first step we choose that rule which rate of equality of a consequent and one of the term of the output linguistic variable has the greater value.

A system that implements the proposed behavior control model of NPC must solve the following additional problems to provide interactive interaction of player and NPC:

- continuous monitoring of gameplay (identification of factors);
- identification of the state of player (identification, analysis and taking into account of cognitive parameters);
- operational modification of the behavior control model of NPC.

III. The Software Supporting the Behavior Control Model of NPC

Software implementation of the behavior control model of NPC (DynamicNPC) contains 6 dynamic link libraries with the unified structure of the input and output data:

- the library of interpretation of player commands (PCommands.dll);
- the library of determination of player cognitive parameters (PCognitiveParams.dll);
- the library of estimation of videogame enjoyment (PSatisfaction.dll);
- the library of operational learning of NPC (NPCLearning.dll);
- the library of control of NPC (NPCManage.dll);
- package library supporting DIT2FLSs (DIT2FLS Package Library [7-8]).

Libraries PCognitiveParams.dll and PCommands.dll are based on data from Microsoft Kinect SDK. XML-format is used for the representation of input and output data of libraries (see example in [8]).

DynamicNPC has a transparent, scalable architecture with the ability to connect to game engines. Users must follow formats of input and output parameters, software requirements and libraries' interface in integration of the DynamicNPC.

DynamicNPC requires the following input parameters:

- linguistic variables describing the influencing factors;
- linguistic variables describing types of states of NPC;
- numeric vector defines the degree of influence of n-th factor on i-th sub-state of NPC;
- set of functions $\langle f_n : X_n \rightarrow Y_i \rangle$ describing the relation between Y_i and X_n;
- fuzzy rule base describing the initial behavior of NPC.

The Microsoft Visual Studio 2010, .NET Technology, and programming language C# were used by the designer of DynamicNPC software.

Software Requirements:

- Operating System: from Windows XP Service Pack 3 (SP3) (32-bit only);
- Microsoft .NET Framework 4.0.

IV. Conclusions

We propose the behavior control model of NPC, which realizes a nontrivial scheme of the interactive interaction of player and NPC. Interact is provided by operational learning of NPC. Operational learning in the model is realized by changing the structure of the fuzzy rule base. The quality of learning of NPC is controlled by the developed criteria of videogame enjoyment. They are based on the analysis of cognitive parameters of player. Analysis of data from NUIs allow us to determine extend required of the cognitive parameters of players.

This paper presents the software supporting of the behavior control model of NPC. The software is presented as a set of dynamic link libraries which are extended and complete, with a clear structure, contains unified data structure. These qualities allow users to integrate the software dynamic libraries to game engines.

Appendix A: Experiment

Lets have a closer look at one example of game, whose rules are based on the famous "Pac-Man". There are two characters in the game: Pac-Man and the enemy. Player can control Pac-Man with the help of sensor controller Microsoft Kinect. The aim of player is to eat all pac-dots. The aim of the enemy is to catch Pac-Man. The enemy can be in three different states: "escape", "neutrality", "aggression". Such factors as satiety of the enemy, its activity and emotional state of player have big influence on the enemy. According to the situation on the field, the enemy goes into one of the possible states, which affects its behavior with Pac-Man. If an enemy touches Pac-Man, a life of Pac-Man is lost.

Lets construct a system to control behavior of the enemy (NPC), according to (1).

The system is described by three input linguistic variables LF (factors) and two output - LS (NPC sub-states).

Linguistic variable LF_1 defines the satiety state of NPC. LF_1 is defined on the primary variable $X_1 = [0, 1]$ and consists of three terms

$$LF_1 = \{\text{"hungry"}, \text{"neutral"}, \text{"full"}\}$$

The meaning of membership function term "hungry" is defined with the help of such formula

$$\underline{\mu}_{hungry}(x) = \begin{cases} 0, & x < 0 \\ 1, & 0 \le x \le 0.2 \\ \dfrac{0.4 - x}{0.2}, & 0.2 < x \le 0.4 \\ 0, & x > 0.4 \end{cases}$$

$$\overline{\mu}_{hungry}(x) = \begin{cases} 0, & x < 0 \\ 1, & 0 \le x \le 0.2 \\ \dfrac{0.45 - x}{0.25}, & 0.2 < x \le 0.45 \\ 0, & x > 0.45 \end{cases} \quad (A.1)$$

The membership functions of the terms "neutral", "full" are defined similarly.

Linguistic variable LF_2 defines the activity of NPC. LF_2 is defined on primary variable $X_2 = [0, 1]$ and consists of three terms

LF_2 = {“passive”, “neutral”, “active”}

Membership functions of terms of LF_2 are defined similarly as in (A.1).

Linguistic variable LF_3 defines the emotional state of player. Player interacts with the NPC with the help of player character (Avatar). LF_3 is defined on primary variable $X_3 = [0, 1]$ and consists of four terms

LF_3 = {“funny”, “still”, “agitated”, “stressed”}

Membership functions of terms of LF_3 are defined similarly as in (A.1).

Linguistic variable LS_1 defines the emotional state of NPC. LS_1 is defined on primary variable $Y_1 = [0, 1]$ and consists of five terms

LS_1 = {“funny”, “still”, “agitated”, “frightened”, “stressed”}

Membership functions of terms of LS_1 are defined similarly as in (A.1).

Linguistic variable LS_2 defines the food needs of NPC. LS_2 is defined on primary variable $Y_2 = [0, 1]$ and consists of three terms

LS_2 = {“low”, “middle”, “high”}

Membership functions of terms of LS_2 are defined similarly as in (A.1).

Each factor LF_j affects i-th sub-state of the NPC. The degree of factors LF which have influence on the sub-states of NPC is presented in the table 1.

TABLE I. DEGREES OF INFLUENCE OF INPUT LINGUISTIC VARIABLE ON OUTPUT

Factors	Sub-states	
	LS₁	***LS₂***
LF_1	0.7	1.0
LF_2	0.9	0.8
LF_3	0.5	0

Dependence of Y_i on X_j is shown in the fallowing set of functions:

$$f_1^{(1)}: Y_1 = 1 \quad X_1,\ f_2^{(1)}: Y_1 = 1 - X_2,\ f_3^{(1)}: Y_1 = X_3$$
$$f_1^{(2)}: Y_2 = 1 - X_1,\ f_2^{(2)}: Y_2 = 1 - X_2,\ f_3^{(2)}: Y_2 = X_3$$

Output linguistic variable LO defines the states of NPC. LO is defined on primary variable U = [0, 1] and consists of three terms

LO = {“escape”, “neutrality”, “aggression”}

LO terms are interpreted as follows:

- "escape" - NPC movement in the opposite direction from Avatar;
- "neutrality" - standing state NPC on the board;
- "aggression" - the movement of NPC towards Avatar, choosing the shortest way.

Membership functions of terms of LO are defined similarly as in (A.1).

The fuzzy rule base, which forms the tactics of interaction between NPC and Avatar, is defined by the expert according to the videogame script. Fuzzy rule base contains 15 rules (see Tabl. 2).

TABLE II. FUZZY RULE BASE

Terms LS_2	**Terms LS_1**		
	low	***middle***	***high***
funny	neutrality	attack	aggression
still	neutrality	neutrality	aggression
agitated	escape	neutrality	escape
frightened	neutrality	neutrality	aggression
stressed	escape	escape	aggression

Fig. 1 illustrates the various states of the NPC movement.

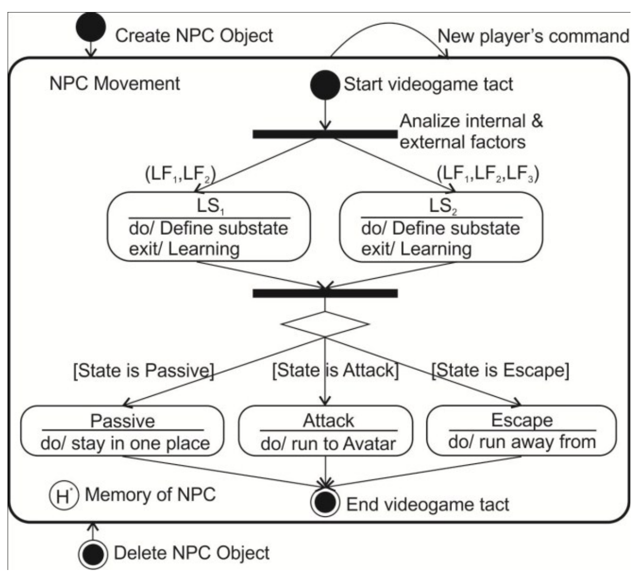

Figure 1. Statechart diagram of NPC

Fig. 2 depicts one of the results of the work, which were developed by behavior control system of NPC.

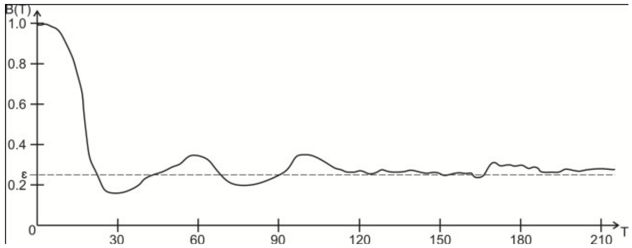

Figure 2. Results of work of behavior control system of NPC

We can see the graph of changing the enjoyment of player in videogame. A minimum level of enjoyment was installed as ε = 0.25. In case, when it will be low, the interest of player should not fall. In this example, NPC learns taking into account all cognitive parameters of player, when the interest of player falls below the ε.

REFERENCES

[1] G. De Prato, C. Feijóo, D. Nepelski, M. Bogdanowicz and J.P. Simon, "Born Digital / Grown Digital: Assessing the Future Competitiveness of the EU Video Games Software Industry," Publications Office of the European Union, 2010, p. 185.

[2] D. Lucio and L. Velho, "Kinect and RGBD Images: Challenges and Applications" Proc. of the 25th SIBGRAPI Conference, Aug. 2012, pp. 36-49.

[3] J.M. Mendel and R.I. John, "Type-2 fuzzy sets made simple," IEEE Transactions on Fuzzy Systems, vol. 10, no. 2, Apr. 2002, pp. 117-127.

[4] N. Kasabov, "Evolving Intelligence in Humans and Machines: Integrative Evolving Connectionist Systems Approach," IEEE Transactions on Computational Intelligence Magazine, vol. 3, no. 3, Aug. 2008, pp. 23-37.

[5] C. Browne and F. Maire, "Evolutionary Game Design," IEEE Transactions on Computational Intelligence and AI in Games, vol. 2, no. 1, March 2010, pp. 1-16.

[6] J.M. Mendel, "Fuzzy Logic Systems for Engineering: a Tutorial," IEEE Proc., vol. 83, March 1995, pp. 345-377.

[7] Cognitive&Uncertainty Research Group. DIT2FLS Library Package. [http://dit2fls.com/projects/dit2fls-library-package].

[8] T. Petrenko and O. Tymchuk, "Package Library and Toolbox for Discrete Interval Type-2 Fuzzy Logic Systems," Proc. of the 18th International Conference on Soft Computing (MENDEL), June 2012, pp. 233-238.