Visual and Computational Modelling of Minority Games

Robertas Damaševičius [1], Darius Ašeriškis [1]

[1] *Department of Software Engineering, Kaunas University of Technology, Studentų 50, Kaunas, Lithuania*

Abstract **–The paper analyses the Minority Game and focuses on analysis and computational modelling of several variants (variable payoff, coalition-based and ternary voting) of Minority Game using UAREI (User-Action-Rule-Entities-Interface) model. UAREI is a model for formal specification of software gamification, and the UAREI visual modelling language is a language used for graphical representation of game mechanics. The URAEI model also provides the embedded executable modelling framework to evaluate how the rules of the game will work for the players in practice. We demonstrate flexibility of UAREI model for modelling different variants of Minority Game rules for game design.**

Keywords: **executable modelling; multi-agent systems; gamification; minority game.**

1. Introduction

A complex system is a system which is made up of a large number of interrelated agents. In such systems the individual agents and the complex interactions between them often lead to behaviors which are not easily predicted from knowledge of individual agents. The concepts of complex systems such as self-organization, emergence and level hierarchies, and methodologies such as multi-agent modeling and simulation gaming, are applicable to a wide range of natural and social phenomena such as ecosystems, social interactions, the economy and financial markets, road traffic, cloud computing, the Internet, disease epidemic modeling, cybersecurity and even entire human societies.

Agent-based models are computational models, which simulate interactions among agents, in order to understand the emerging behavior of the overall system based on the microscopic behavior dynamics of each agent [1]. Agent-based modeling and simulation enables the researcher to create, analyse, and experiment with models composed of agents that interact within an artificial environment [2]. This approach combines elements of game theory, multi-agent systems and stochastic methods.

Game theory recently has become widely used in social sciences and economics [3]. A game can be described as any social situation involving two or more players. A game is a system in which players are drawn in an artificially made-up conflict, which is defined by rules and the outcome can be measured [4]. The goal of game theory is to find and describe the behaviors of players, which provide best response to other players' individual decision choices. The rules governing interaction between the two players are defined as a part of the description of the game. In a game, the rewards are defined by the rules of the game as points, badges, etc. The player is free to make a move or to do an action as defined by the rules of the game aiming to increase his outcome of the game (reward). A social game is a game defined over the elements of social state, social motivations, and social moves [5]. Social gaming is directly related to games with a purpose (GWAP). GWAP are games, in which some useful computation is performed by humans as an element of a game [6]. GWAP have been applied in areas of computer vision [7], content management [8], semantic search [9], and education [10]. Humans, however, require some incentive (reward or engagement) to become and remain part of a social game of GWAP, which is defined as reinforcement model.

Designing social games or GWAP requires gamification, i.e. turning human‘s everyday interactions or work into games that allow to enhance productivity and engagement of a user for business purposes or achieving other meaningful results. Gamification [11] involves the use of game

DOI: 10.18421/TEM61-16
https://dx.doi.org/10.18421/TEM61-16

Corresponding author: Robertas Damaševičius,
Department of Software Engineering, Kaunas University of Technology, Kaunas, Lithuania
Email: robertas.damasevicius@ktu.lt

mechanics to non-game activities in order to influence people‘s behavior, to engage audiences and solve problems. Gamification and serious games are related because their common aim is to achieve some value beyond plain entertainment. Serious games offer an enjoyable way to solve real-world problems [12]. However, the design of engaging games that can keep their players interested in continuing playing the games for a long time is still a major problem in the area of gamification research. To understand gamification and its effects the use of effective game modeling and simulation methods and tools is required. Recently the game mechanics of GWAPs have begun to be modeled formally [13], aiming to standardize the design of GWAP.

We focus on Minority Game (MG), which is a kind of social game with both coordination and competition mechanisms active [14]. MG has become a paradigm to study social phenomena with a large number of competing agents. In case of a single available resource, agents taking minority strategy are the guaranteed winners. MG has been extensively studied in the domain of statistical physics [15], but it also has become common for modeling various social and economic systems with shared resources [16]. To analyse games, mathematical models are developed to predict and understand player in a game as well as for understanding and selecting strategies that will lead them to a better payoff in the future [17].

The aim of the paper is to describe the use of the UAREI (User-Action-Rule-Entities-Interface) visual modeling language for analyzing, modeling and simulating different variants of the MG.

2. Minority game

Minority Game (MG) has been studied previously as a model of market behaviour [18]. MG is based on the idea that the decision of the majority is always wrong. Minority-like games occur frequently in everyday life, when an action taken by more people becomes less attractive. This occurs, e.g., in the selection of candidates in the university admission system, or when selecting a route in urban traffic systems. MG is also related to congestion games, which can model diverse phenomena such as processor scheduling, routing, and network design [19]. In these games each agent is allowed to choose a subset from a set of resources, and agents’ costs depend on the number of the other agents using the same resources. A congestion problem arises whenever there is a competition for a limited resource and the lack of coordination among users how to exploit it [20].

The classical MG is defined as follows [21]. The MG is played with an odd number of agents *N*. Each agent *i* can choose between two possible actions: to use the resource – represented by 1 – or not to use it – represented by 0. The payoff is +1 if the agent is in the minority and −1 if it is in the majority.

The principles of MG have been formulated in [21] as follows: (1) Competition for limited resources: not all agents can win at the same time. (2) Behaviour is good only with respect to other agents’ behaviour. (3) A good behaviour may become bad when other agents change their behaviour. (4) Agents try to predict next winning choice, which is defined only by their own choices.

We begin by first introducing the notation and the terminology used in this paper:

Agent: A player of the game that makes decisions based on its strategy. The number of agents that participate in the MG is $N = \{2k+1:\ k\ in\ Z\}$. Agent is indexed by an integer *I*, where $I \in \{1,\ 2,...,N\}$.

Choice: An action of an agent. Choice *C* has two possible values: $C \in \{0,\ 1\}$. The total number of choices are *N*. In the game, the choices can be viewed as a sequence of choices $C_1,\ C_2,...,C_N$ where C_n is the choice of ***n***-th agent.

Game: Every run of the MG is a “game”. The total number of games is specified as *G*.

Minority Choice: The winning outcome of the game in the MG. Formally, the minority choice in game is defined by:

$$o = \begin{cases} 1, & \sum C_i < \frac{n}{2} \\ 0, & otherwise \end{cases}$$

Strategy: A set of rules of a player, which take the previous minority choices as inputs, and governs the choice of future individual actions of a player [22]. A strategy $s \in S$ maps each possible combination of previous winning actions to the action a_i to be taken next by agent *i*.

Hereinafter, we focus on the variable payoff, coalition-based, and ternary variants of the MG. In multi-agent systems, coalitions allow to promote cooperation of agents aiming to improve their performance, or increase their benefits, with applications in e-business.

Variable payoff MG (VPMG) is important in studying emergent behaviour in complex systems in real-world social and biological systems, which depend upon resources which increase or decrease in various ways as the size of the minority group changes [23]. In general case, there may be various kinds of rewards and the payoff may depend on the size of the minority group.

In the coalition -based MG (CBMG), when a group of players agree to cooperate they gain an advantage over other players [24]. We consider the advantage obtained in CBMG by a coalition sharing their state. There can be two types of coalition: equal and unequal. In case of equal coalition, the prize is shared in equal parts by the members of the coalition. In case of unequal coalition, the prize is shared by unequal parts using the ration defined by the coalition agreement. The winning strategy is to enter into the most advantageous coalition agreement that guarantees the most generous pay-off. The game is transformed to the auction game, there players bid to each other for the most advantageous offer. The coalition game algorithm is defined in Figure 1.

In ternary voting MG (TVMG), a third option is added for the decision of each agent: abstention. TVMG is important in decision theory with applications in political science [25]. Choosing a third option prevents a player from winning, but also from losing the game. Ternary voting introduces more options for bargaining (therefore, cooperation) in search of common agreement over a set of feasible alternatives.

```
ALGORITHM: CoalitionGame
BEGIN
   IF player is in coalition
      Select best offer
      IF offer is better than current coalition
         Leave current coalition
         Join player with best offer
      ELSE if player is alone
         Bid other players for coalition
         Accept best offer and enter the coalition
      ENDIF
   ENDIF
   Play canonical minority game
   Share rewards
   Generate leaderboard
END
```

Figure 1. Algorithm of the coalition-based minority game

What is common to the analysed extensions of MG is the influence of cooperation factors on the results of the game. To succeed in the game, the players must cooperate with other players or at least to consider the behaviour of other players when taking the decisions. So the game moves to the meta-strategy level [26].

The reinforcement model in a MG is defined as follows. For each player, satisfaction points are awarded in each step for:

- Winning (the player has won in the previous round of the game);
- Leadership (the player has been listed in one of the top positions of the leaderboard);
- Advancement (the player overtook competitors in the previous move);
- Achievement (the player has achieved the best result in some record, e.g., was in the smallest minority group);
- Power (the player, whose decision more often has decided the outcome of the game round).

Hereinafter we consider winning as the most simple variant of the reinforcement.

3. UAREI model for description of game systems

For modelling of the game systems, we use the UAREI (User-Action-Rule-Entities-Interface) model. UAREI is a model for formal specification of software gamification [27, 28], and the UAREI visual modelling language we use for graphical representation of game mechanics. Below we present a brief description of the UAREI model as follows.

The gamified systems can be described as tuple $G = \{U, A, R, E, I\}$, here: U – users, which are interacting with the system; A – actions, which trigger system behaviour; R – rules, which encapsulate logic in the system; E – data entities; and I – interfaces which define data format.

The users are defined as a tuple $U = \{L_U, S_U\}$, here: L_U – a set of all outgoing links to other elements in the model; and S_U – a selection function which defines how a user is selected from a collection in a simulation mode.

Actions are a collection $A = \{A_1, A_2, \ldots, A_i, \ldots, A_n\}$, here A_i is a single action, n the total number of actions. A single action is defined as $A_i = \{L_A, S_A\}$, here: L_A – a set of all outgoing links to other elements in the model, and S_A – a selection function, which defines how an action entity is selected from a collection.

Rules are a collection $R = \{R_1, R_2, \ldots, R_i, \ldots, R_n\}$, here R_i is a single rule, n the total number of rules. A single rule is defined as $R_i = \{L_R, r_i(C, M)\}$, here: L_R – a set of all outgoing links to other elements in the model, and $r_i(C, M)$ is a rule function defined as:

$$r_i(C, M) = \begin{cases} NULL - if\ no\ value\ is\ computed \\ y - if\ value\ is\ computed\ by\ rule \end{cases}$$

here: C – context of current execution path; M – a system model; y is a result value, and the NULL value is assigned if a rule was not computed successfully.

Rules are used to control context flow in the system. If a rule execution evaluates to an empty result the current execution path is continued. We can define the "else" path by using inversion "$!R_i$". No data will be stored in storage and no other rules will execute if the previous rule failed or returned empty value, but system flow will continue giving feedback to the user node. Rules can update the context in any way needed for the application.

Entity collection is a collection of all data entities in the system $E = \{E_1, E_2, \ldots, E_i, \ldots, E_n\}$, here E_i is a single storage entity and n is the total number of storage entities. A single entity is defined as $E_i = \{D, O, L_E\}$, here: D – entity scheme definition, O – data objects, and L_E – a set of all outgoing links to other elements in the model.

Interface is a collection $I = \{I_1, I_2, \ldots, I_i, \ldots, I_n\}$, here I_i is a single interface and n is the total number of interfaces. A single interface is defined as $I_i = \{L_I, Q\}$, here: L_I –a set of all outgoing links to other elements in the model, Q – data query, on which the data for the interface is selected.

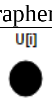
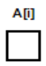
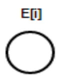
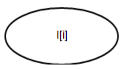
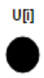
The UAREI model is visualized as a directed graph consisting of nodes (vertices) and links (edges) between nodes as follows:

$G = \{L, N\}$,

here: N is a set all nodes $N = \{N_1, N_2, \ldots, N_i, \ldots, N_m\} = U \cup A \cup R \cup E \cup I$; L is a set of links between nodes $L = L_U \cup L_A \cup L_R \cup L_E \cup L_I$, and L_U, L_A, L_R, L_E, L_I are collections of corresponding types of nodes $L_X = \{L_{X_1}, L_{X_2}, \ldots, L_{X_i}, \ldots, L_{X_{n_X}}\}$, L_i is the list of links, $L_i = (N_{out}; N_{in})$, here $N_{in}, N_{out} \in N$, L_{N_i} – are links which start N_i node.

In Table 1. we present the list of graphical symbols (graphemes) used in the UAREI model diagrams.

Table 2. Graphical notation of UAREI modelling language

Type	Grapheme	Description
User node		Visualizes system user group. Normally a single action is triggered from this node
Action node		Visualizes an action. Action triggers its outgoing connections. Normally actions are connected to rules and other actions
Rule node		Visualizes a rule node. Rule encloses all logic of a model. Rule triggers other rules, entities and interfaces
Entity node		Visualizes a data entity. On triggering the node stores the data received with the current context
Interface node		Visualizes user interfaces. Triggers user nodes finishing the feedback loop
Connection		Visualizes relationships in the model. The direction of arrow points from the outgoing node to the incoming node
User node		Visualizes system user group. Normally a single action is triggered from this node

4. Modeling minority games in UAREI

Formally we can define an agent in CBMG as follows:

$$\alpha_1 i = \{a_1 pick\ (model), a_1 key\ \ (model), \\ a_{feedback}(model), E_{agent\_state}\ \}$$

In general, user behaviour can be supplemented by agent behaviour. An agent first picks an action using a_{pick} function (in case of original MG definition as El Farol Bar problem [29], the agent picks “go to bar” or “stay at home” based on his strategy). To map the strategy to the current model state we define a a_{key} function, which generates a memory key to reference the current situation. After the cycle ends the agent receives a call-back to $a_{feedback}$ function to evaluate his choice. entity stores all data relevant to particular agent.

In case of classic MG we can specify such agent description as:

$$\alpha_1(i, minority\ game) = \{a_1 pick\ (model), a_1 key\ (model),$$

$$a_{feedback}(model), E_{agent\_state}\ \},$$

here a_{pick}:

- On first call
  - Generate S random strategies for all possible keys.
  - Initialize strategy quality so one would be better.
- On all calls
  - Generate key using $a_{key}(model)$ function for current model state.
  - Return best quality strategy and take from it action for generated key.

$a_{key}(model)$: return M records from game win history entity;

$a_{feedback}(model)$: if the action of previously chosen strategy has won, then increase strategy quality by one. The function is executed in every round of game.

E_{agent_state} is a collection of strategies and an vector of strategy quality.

The user (player) behaviour is defined as follows:

$U = \{L_{user}, S_U, \alpha\}$

MG agent model has a problem because it is bound by $N_{user} \times N_{actions}\ S^M$ which in reality causes performance issues then modelling large numbers of agent in a model with large number actions where agents use large number of strategies and can remember large history. To avoid this we offer an alternative MG agent:

$$a_1(i, alternative\ minority\ game) = \{a_1 pick\ (model), a_1 key\ (model), a_1 feedback\ (model), E_{agent\_state}\}$$

here a_{pick} generates a key using $a_{key}(model)$ function for current model state, which returns a random action, if it is called for the first time, and the best action, if called subsequently.

Formally, MG is defined as $G_{minority\ game} = \{U, A, R, E, I\}$,

here $U = \{L_{users}, S_{order}, \alpha_{minority\ game}\}$;

$L_{users} = \{A_{Go\ to\ bar}, A_{Stay\ at\ home}\}$; S_{order} - pick user from order for each round; $\alpha_{minority\ game}$ minority game agent list of N players with S strategies and M memory size; $A = \{A_{Go\ to\ bar}, A_{Stay\ at\ home}\}$;

$A_{Go\ to\ bar} = \{S_{random}, L_{Go\ to\ bar}\}$; $A_{Stay\ at\ home} = \{S_{random}, L_{Stay\ at\ home}\}$;

S_{random} – randomly generated action data; $L_{Go\ to\ bar} = \{R_{Record\ Option}\}$;

$L_{Stay\ at\ home} = \{R_{Record\ Option}\}$.

$$R_{Record\ Option} = \{r_{Record\ Option}, L_{Record\ Option}\}$$

$$r_{Record\ Option} \begin{cases} \text{"Go to bar"}, & if\ chosen\ action\ was\ A_{Go\ to\ bar} \\ \text{"Stay at home"}, & if\ chosen\ action\ was\ A_{Stay\ at\ home} \end{cases}$$

; $L_{Record\ Option} = \{E_{Choices}\}$;

$E_{Choices} = \{L_{Choises}, D_{Choises}\}$ is the entity collecting all user choices, here $L_{Choises} = \{I_A, R_{Win}\}$; $D_{Choises}$ has three fields: user ID, chosen action, and game round;

I_A – defines a view which groups users by choices; $I_A = \{L_A, Q_A\} = \{\{U\}, Q_A\}$; Q_A- groups data from $E_{Choises}$ by round and chosen action and counts all users in group;

$R_{Win} = \{L_{Win}, r_{win}\} = \{\{E_{History}, U\}, r_{win}\}$, here r_{win}- defines the winner for each round. The rule is executed once at the end of every round and returns the action which was chosen by the minority group $E_{History} = \{L_{History}, D_{History}\} = \{\{U, I_{Leaderboard}\}, D_{History}\}$;

$D_{History}$ – has two fields: round number and winning action; $I_{Leaderboard} = \{L_{Leaderboard}, Q_{Leaderboard}\} = \{\{U\}, Q_{Leaderboard}\}$;

here $Q_{Leaderboard}$ – computes user success rate.

In Figure 2., we can see the classic MG model represented visually in UAREI.

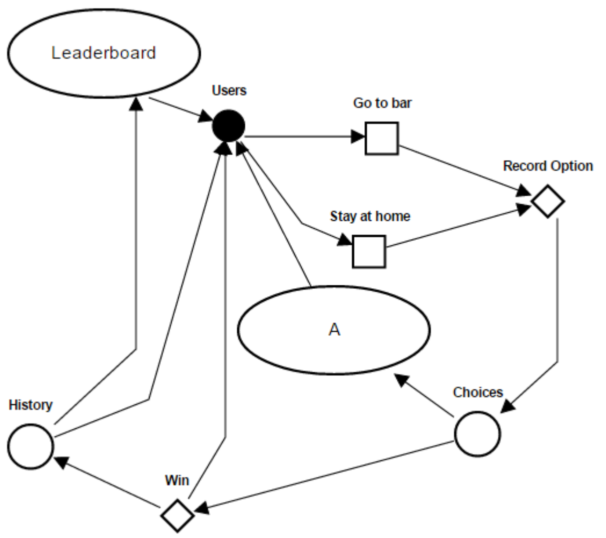


Figure 2.Minority game model in UAREI

Using UAREI we also can analyse different variants of MG model. For variable payoff MG, the visual representation of the model is the same as given in Figure 2. The model for coalition-based MG is presented in Figure 3. A new entity "Bank" has been introduced with a specialized interface to visualize the monetary situation of agents in the model. Figure 4. represents the ternary voting MG model. This model has all the attributes from cooperation-based MG model with the addition of action "Sustain".

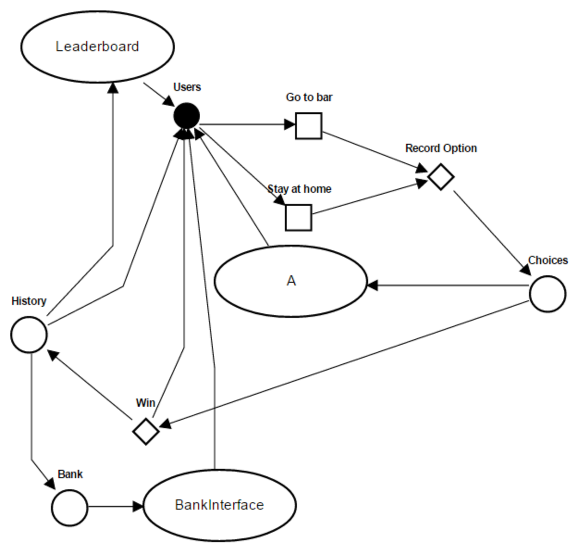


Figure 3. Model of coalition-based MG in UAREI

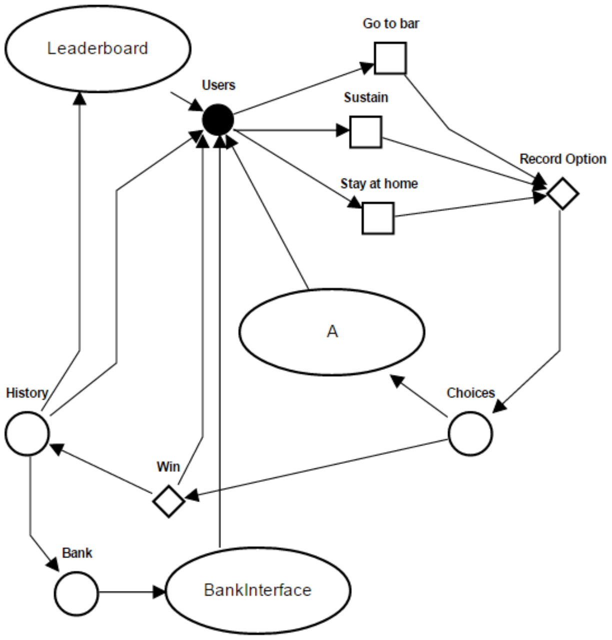


Figure 4. Model of ternary voting MG in UAREI

The modifications of classic MG model are summarized in Table 3.

Table 3. Modeling variants of minority game in UAREI

Variant of MG	Change in classic MG model
Variable payoff	In this model $a_{feedback}(model)$ gives N-k if the user wins and k-N to the strategy.
Coalition-based	$\alpha = \{\alpha_{minority\ game}, \alpha_{cartel}\}$ $N_{minority\ game}$ and N_{cartel}
Ternary voting	$A = \{A_{Go\ to\ bar}, A_{Stay\ at\ home}, A_{Sustain}\}$

5. Simulation and results

We have run simulations with different variants of minority models defined in Section 4, including the classic MG, and observed their behavior expressed as the winning function, defined here as the ratio of wins to the number of played games in percents. In Figure 5., we can see the histogram of winning function after 100 game rounds in different simulations of the classic MG. We can see that the number of winning agents follows the Gaussian probability distribution (see the values of mean, std and Kolmogorov-Smirnov (KS) normality test in Table 4.).

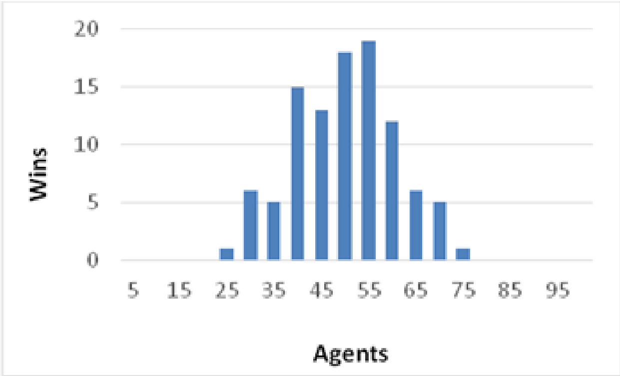


Figure 5. Histogram of wins in simulated classic MG

Distribution of wins for the variable payoff MG is presented in Figure 6. The size of reward is in proportion to the size of minority group, which favors the formation of small minority groups as well as allows for more rapid changes in the leaderboard of players during the game.

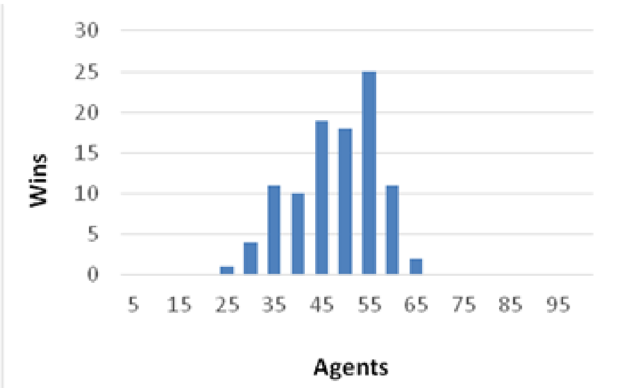


Figure 6. Histogram of wins in simulated variable payoff MG

In coalition-based MG, simulation introduces a 20-member coalition into a system. Members of the coalitions are divided into two equal groups which bid on different actions and split the reward between the members of the coalition. Each agent bids 1 point per round. Rewards are distributed equally to all players. The histogram of wins (Figure 7.) shows a small shift over the random change success rate (see Table 4.). Therefore, one can conclude that the introduction of coalition as a meta-game strategy into a classic MG model allows improving the results of the game for some players.

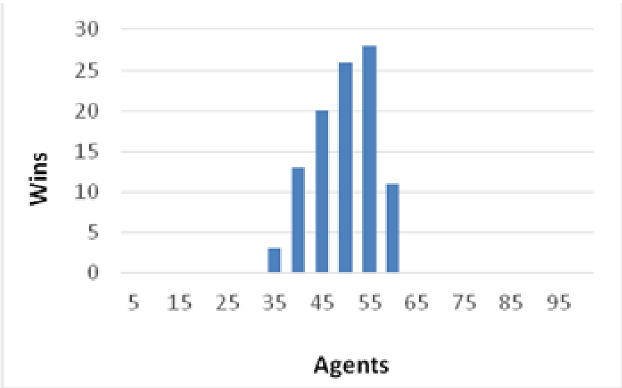


Figure 7. Histogram of wins in simulated coalition-based MG

In the simulation of the ternary voting MG (TVMG), the model introduces a third option for payers to sustain from playing in their strategies. All players who choose to sustain do not participate in the current round of the game. All agents initially have 10 points each. In every round, each participating agent must bid 1 point. A player who lost all his points has to leave the game. In the histogram of wins in TVMG (Figure 8.), we can see two peaks, which correspond to low performing agents and high performing agents, which is also confirmed by the results of the KS normality test (see Table 4.).

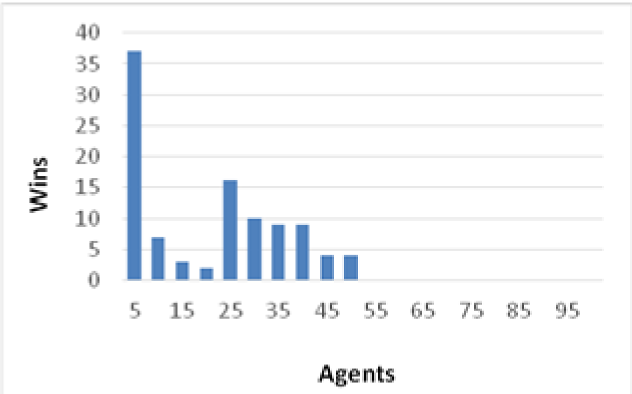


Figure 8. Histogram of wins in ternary voting MG

To evaluate the interestingness of each variant of MG, we use the negentropy value of win function. In information theory and statistics, negentropy is used as a measure of distance to normality. The results of coin tossing game, the most simple and the least interesting game without any strategy of playing would have Gaussian distribution. On the other hand, the games with uniform or constant probability of win are equally uninteresting. The game, which differs more in terms of negentropy from the Gaussian distribution with the same mean and variance, can be considered more interesting. Such entropy-based measures have already been used for defining the concepts of interestingness and surprise of data, including that of algorithmic zero sum games [30]. The results of statistical analysis of win results in the analyzed variants of MG are presented in Table 4.

Table 4. Statistical evaluation of variants of minority game

Variant of MG	Mean	Std.	Skew ness	KS test	Entro py	Negen tropy
Classic	49.95	10.78	-0.04	1	0.932	0.02
Variable payoff	48.02	8.87	-0.41	1	0.838	0.03
Coalition-based	49.75	6.50	-0.29	1	0.710	0.03
Ternary voting	20.74	14.82	0.32	0	0.844	0.16

The win values for classic, variable payoff and coalition-based variants of MG are normally distributed and with acceptable asymmetry (skewness between -2 and 2). The ternary voting model departs from the normality due to the rules of the game, which throw out players with poor performance out of the game. The value of negentropy, which is used to evaluate the interestingness of the game, shows that the classic MG, which has the simplest set of game rules, as the least interesting, whereas the ternary voting variant of MG, which allows the users to abstain as well as to go bankrupt and leave the game, is the most interesting.

6. Conclusion

We have developed the visual modeling language and simulation framework UAREI, which is intended for visualizing and modeling game rules and game mechanics in the gamified systems. In this paper we have focused on four variants of Minority Game (MG) as a case study aiming to analyze, and computationally evaluate the gamification models described in UAREI and based on these games. The results of agents in each game are analyzed and compared using a simple win function, which registers the number of wins for each agent. The results of classic MG model are similar to random coin toss game, meaning that the game most likely would not be interesting for its players for a long time. The extensions of the classic MG introduce a layer of meta-game to the game thus introducing new opportunities for the players to cooperate or compete between themselves. The variable payoff MG provides an opportunity for an agent to earn more points in one game round and makes the game more interesting. The ternary voting MG model introduces a third decision option as well as bankruptcy of the player as one of the outcomes of the game. Such model allows analysis of player behavior, which is more similar to real-world games. The coalition-based MG model enriches the rules of the game by an opportunity of bargaining between players thus introducing a market-like behavior in the meta-game scenario.

The analysis and computation modeling of different MG models provides new insights into behavior of players and allows comparing models based on their interestingness (evaluated in terms of negentropy of probability distribution of the win function). Such evaluation can help to produce the sustainable game mechanics, which can keep game players motivated in continuing playing the games with a purpose.

In future work we will explore more thoroughly the use of various information-theoretic measures to define the interestingness of the game rule-sets as well as relate them with empirical data of player engagement in real games.

References

[1]. G. A. Marsan. (2009). New paradigms towards the modelling of complex systems in behavioral economics. *Mathematical and Computer Modelling*, 50, 584-597.

[2]. G. N. Gilbert. (2008). *Agent-based models.* Sage, London, UK.

[3]. A.E. Roth. (2002). The Economist as Engineer: Game Theory, Experimentation, and Computation as Tools for Design Economics. *Econometrica*, 70(4), 1341–1378.

[4]. K. Salen, and E. Zimmerman. (2003). *Rules of Play: Game Design Fundamentals.* The MIT Press.

[5]. D. G. Shapiro, K. Tanenbaum, J. McCoy, L. LeBron, C. Reynolds, A. Stern, M. Mateas, B. Ferguson, D. Diller, K. Moffitt, W. Coon, and B. Roberts. (2015). *Composing Social Interactions via Social Games. Proc. of the 2015 Int. Conference on Autonomous Agents and Multiagent Systems AAMAS 2015*, 573-580.

[6]. L. von Ahn, and L. Dabbish. (2008). Designing Games With A Purpose. *Communications of the ACM* 51, 57.

[7]. L. Galli, P. Fraternali, D. Martinenghi, M. Tagliasacchi, and J. Novak. (2012). A Draw-and-Guess Game to Segment Images. *Proc. of 2012 Int. Conference on Privacy, Security, Risk and Trust (PASSAT) and 2012 Int. Confernece on Social Computing (SocialCom)*, 914-917.

[8]. E. Giouvanakis, C. Kotropoulos, A. Theodoridis, and I. Pitas. (2013). A game with a purpose for annotating Greek folk music in a web content management system. *Proc. of 18th Int. Conference on Digital Signal Processing (DSP)*, 1-6.

[9]. M. Lux, M. Guggenberger, and M. Riegler. PictureSort: gamification of image ranking. (2014). *Proc. of the First Int. Workshop on Gamification for Information Retrieval (GamifIR '14)*, 57-60.

[10]. M. Muratet, P. Torguet, J.-P. Jessel, and F. Viallet. (2009). Towards a serious game to help students learn computer programming. *Int. J. Comput. Games Technol.*, Article 3, 12 p.

[11]. S. Deterding, M. Sicart, L. Nacke, K. O'Hara, and D. Dixon, Gamification: Using Game Design Elements in Non-Gaming Contexts. (2011). *Proc. of the 2011 Annual Conference Extended Abstracts on Human Factors in Computing Systems, CHI EA'11,* 2425–2428.

[12]. G. Richter, D.R. Raban, and S. Rafaeli. (2014). Studying Gamification The Effect of Rewards and Incentives on Motivation. In T. Reiners and L. C. Wood (Eds.), *Gamification in Education and Business.*

[13]. K. T. Chan, I. King, and M.-C. Yuen. (2009). Mathematical modelling of social games. *Proc. of the 2009 Int. Conference on Computational Science and Engineering,* Vol. 04.

[14]. J. Linde, J. Sonnemans, and J. Tuinstra. (2014). Strategies and evolution in the minority game: A multi-round strategy experiment. *Games and Economic Behaviour* 86, 77-95.

[15]. D. Sherrington. (2006). The minority game: A statistical physics perspective. *Physica A: Statistical Mechanics and its Applications*, 370(1), 7–11.

[16]. K. Wawrzyniak. (2011). *On phenomenology, dynamics and some applications of the minority game.* Ph.D. thesis, Institute of Computer Science, Polish Academy of Sciences, 2011.

[17]. J. E Mazur. (2006). Mathematical Models and the Experimental Analysis of Behaviour. *Journal of the Experimental Analysis of Behaviour*, 85(2), 275–291.

[18]. Y. Ma, G. Li, Y. Dong, and Z. Qin. (2010). Minority game data mining for stock market predictions. *Proc. of the 6th international conference on Agents and data mining interaction (ADMI'10),* 178-189.

[19]. E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown (2004). Run the GAMUT: A Comprehensive Approach to Evaluating Game-Theoretic Algorithms. *Proc. of the 3rd Int. Joint Conference on Autonomous Agents and Multiagent Systems AAMAS '04,* Vol. 2, 880-887.

[20]. Bottazzi, G. Devetag, and G. Dosi. (2002). Adaptive learning and emergent coordination in minority games. *Simulation Modelling Practice and Theory*, 10(5-7), 321-347.

[21]. D. Challet, and Y.-C. Zhang (1997). Emergence of cooperation and organization in an evolutionary game. *Physica A*, 246(3-4):407-418.

[22]. W. E. Walsh, R. Das, G. Tesauro, and J. O. Kephart (2002). Analyzing complex strategic interactions in multiagent systems. *Proc. of Workshop on Game Theoretic and Decision Theoretic agents.*

[23]. Y. Li, A. VanDeemen, R. Savit. (2000). The minority game with variable payoffs. *Physica A,* 284, 461-477.

[24]. M. Sysi-Aho, J. Saramäki, and K. Kaski (2005). Invisible hand effect in an evolutionary minority game model. *Physica A* 347, 639-652.

[25]. D.S. Felsenthal, and M. Machover. (1997). Ternary voting games. *Int. J. Game Theory* 26(3), 335-351.

[26]. C. Kiekintveld and M. P. Wellman. (2008). Selecting strategies using empirical game models: an experimental analysis of meta-strategies. *Proc. of the 7th international joint conference on Autonomous agents and multiagent systems AAMAS '08*, Vol. 2, 1095-1101.

[27]. D. Ašeriškis, and R. Damaševicius. (2014). Gamification Patterns for Gamification Applications. *Procedia Computer Science,* 39, 83-90.

[28]. D. Ašeriškis, and R. Damaševicius. (2014). Gamification of a Project Management System. *Proc. of the 2014 Int. Conference on Advances in Computer-Human Interactions, ACHI2014*, 200-207.

[29]. A. W. Brian. (1994). Inductive Reasoning and Bounded Rationality. A*merican Economic Review* (Papers and Proc.) 84: 406–411.

[30]. J. Schmidhuber. (2009). Driven by Compression Progress: A Simple Principle Explains Essential Aspects of Subjective Beauty, Novelty, Surprise, Interestingness, Attention, Curiosity, Creativity, Art, Science, Music, Jokes. In, G. Pezzulo et al. (Eds.), *Anticipatory Behavior in Adaptive Learning Systems.* Lecture Notes in Artificial Intelligence, 5499, 48-76.