Creating an immersive game world with evolutionary fuzzy cognitive maps

Yundong CAI

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

Zhiqi SHEN

Boyang LI

Creating an Immersive Game World with Evolutionary Fuzzy Cognitive Maps

Article *in* IEEE Computer Graphics and Applications · March 2010

DOI: 10.1109/MCG.2009.80 · Source: DBLP

5 authors, including:

Yundong Cai
Nanyang Technological University

Chunyan Miao
Nanyang Technological University

Ah-Hwee Tan
Nanyang Technological University

Digital Object Indentifier 10.1109/MCG.2009.80 0272-1716/$26.00 © 2009 IEEE

Creating an Immersive Game World with Evolutionary Fuzzy Cognitive Maps

Yundong Cai, Chunyan Miao, *Member, IEEE,* Ah-Hwee Tan, *Senior Member, IEEE,* Zhiqi Shen, *Member, IEEE* and Boyang Li

***Abstract*—An increasing number of serious games have been developed to enhance the user experiences in education and training. In order to bridge the gap of game experiences in the virtual environment and in the real life, it is crucial to generate believable characters and contexts in real-time. However, the variables to be simulated for a large-scale serious game are numerous. These variables are involved in complex causal relationships and their values change over time. In view that world modeling has not been well addressed with conventional models, this paper uses a computational model Evolutionary Fuzzy Cognitive Map (E-FCM) to model the variables of the characters and contexts with the causalities among them in serious games. As an extension of FCM, E-FCM models not only fuzzy causal relationships but also probabilistic causal relationships among the variables. It also allows asynchronous updates of the variables, so that they can evolve in a dynamic and stochastic manner. An interactive evolutionary algorithm has been proposed to determine the causal relationships in E-FCM. As a result, the players can have a more engaging and immersive experience from these games.**

***Index Terms*—Immersive game world, serious game, computational model.**

I. Introduction

SERIOUS games have gained a lot of interests from the research personnel and the industry for its advantages over traditional methods on education and training [1]. Recently, an increasing number of serious games have been deployed in 3D virtual environments to enhance the immersive experience of gameplay. According to Zyda [2], in order to achieve an immersive gameplay experience, cognitive game design approach enables developers to create theories and methods for modeling and simulating computer characters, stories, and human emotions though affective computing. As the causal relationships represent the way by which the players experience the game world through their own induction, there is a need to create a cognitive model to represent the causal relationships. A typical game world includes two key elements: characters (non-player characters and players) and the contexts. In order to model a character vividly, a large number of variables are required, e.g., physical parameters (strength, age etc.), emotions (happiness, sadness etc.), behaviors (moving, speaking etc.) and so on. There are several important issues for real-time character modeling, as outlined below.

Yundong Cai, Chunyan Miao, Ah-Hwee Tan and Boyang Li are with Emerging Research Lab, School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore (email: {caiy0004, ascymiao, asahtan, byli}@ntu.edu.sg).

Zhiqi Shen is with the Information Communication Institute of Singapore, School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore (email: zqshen@ntu.edu.sg).

Manuscript received July 30, 2008.

- How can a character's behaviors and emotions be updated according to the changes in contexts?
- How can a character's behaviors be consistent with his/her emotions?
- How does a character affect the environment/context through his/her behaviors?

The main concern of context modeling is how the values of the context variables are updated believably in real-time. For example, when the number of fish in the ocean increases, the number of sharks which feed on the fish will also grow. After some time, when the rate of fish being preyed by shark exceeds the rate of fish reproduction, the number of fish will drop. This is a typical network of causal relationships in an eco-system. Due to the complex causal relationships among contexts, the contexts will be updated in a dynamic and stochastic manner. It is challenging to simulate these updates in a virtual environment, especially for large-scale serious games.

In order to solve the problems mentioned above, an effective cognitive computational model is needed to model the causalities among the variables. So far, many cognitive computational models of dynamic causal relationships have been developed. Some of them are qualitative models for analyzing the trend of the events, e.g., rule-based expert system [3] and Markov-decision processes (MDP) [4] etc. EMA is a famous computational model by Gratch based on the cognitive appraisal theory [5], which combines previous works to be a sound model. It uses causal interpretation to describe the causal relationships among the events.

For serious games in the virtual world, there is a need to model the concepts with precise values rather than symbolic reasoning. Therefore, the quantitative models would be more useful. Cathexis is a distributed model by Velásquez [6] to model the emotions though activation functions, but it only models some basic emotions and reactions. Fuzzy Cognitive Map (FCM) [7] by Kosko is one efficient inference engine to model such complex causal relationships in both qualitative and quantitative way easily. Kosko and Dickerson [8] also modeled the hunting process of sharks and fishes by FCM in a virtual world. In [9], the author used FCMs to model the intensions/movements of the sheepdog and sheep in the virtual world successfully. As a generic model, FCM relies on quite a number of assumptions. For example, the activation values of the concepts are updated simultaneously at the same rate, and the causalities among the concepts are always in effect. However, as these assumptions may not always hold, FCM is not

Digital Object Indentifier 10.1109/MCG.2009.80 0272-1716/$26.00 © 2009 IEEE

powerful or robust enough to model a dynamic and evolving virtual world. In order to solve these shortcomings, numerous extensions of FCM have been proposed. In particular, Miao *et al.* [10] developed Dynamic Causal Network to model the concepts quantitatively with time variables. Moreover, the causality between two variables might be probabilistic rather than deterministic, beyond the fuzziness. Song *et al.* [11] proposed probabilistic events to model the uncertain concepts. In order to describe the general logical operation (AND/OR) of rules, rule-based FCM is also proposed [12]. In addition, Evolutionary Multilayered Fuzzy Cognitive Maps [13] was also used as an inference tool for a real-time system with evolutionary strategy. However, the models above are mostly used as inference engines rather than real-time modeling and simulation. On the other hand, Bayesian Network models the probabilistic causalities among the concepts, but does not describe the fuzzy causalities quantitatively. To our best knowledge, there is still a lack of good solutions that meet the requirements of virtual world modeling, in particular the modeling of the characters and the contexts.

Based on FCM, we propose a computational model, namely the Evolutionary Fuzzy Cognitive Map (E-FCM). E-FCM redefines concepts and their causal relationships, and re-designs the process of how the values of the concepts are updated [14]. In E-FCM, each concept has its respective evolving time schedule (as different concepts update asynchronously) and has a small self-mutation probability to update the value randomly. The causality between two concepts is not only represented with a single fuzzy value; a conditional probability is also used to quantify the probability of the causality. Therefore, E-FCM presents both fuzzy causalities and probabilistic causalities, and allows asynchronous concept updates.

In this paper, we use E-FCM to model the dynamic concepts and their causal relationships in the virtual world, which includes the modeling of the characters modeling and the contexts. The values of the concepts are updated asynchronously according to the changes in the values of the concepts affecting them, and the changes are subject to the causal probabilities. As a result, the virtual world becomes more dynamic with believable characters and contexts, consequently enhancing the player experiences in the interactions.

The rest of this paper is organized as follows. Section 2 outlines the architecture and game world representation in serious games. Section 3 presents the elements of E-FCM model and illustrates how E-FCM can be adopted in the game world modeling. A case study based on an exemplary scenario implemented in the Torque game engine is reported in Section 4, complete with experimental results and discussions. Lastly, we draw the conclusions and show our future work.

II. Game World in Serious Games

A. Serious Game Architecture

As shown in Figure 1, a serious game architecture in our view is composed of the following components:

1) *Game Scenario*: It denotes the designed text-based game scenario by the scriptwriter.

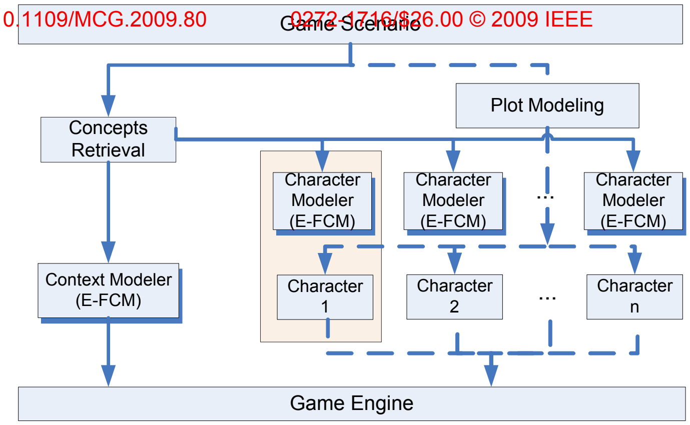


Fig. 1. A serious game architecture (The game scenario is presented in the game engine with two parts: game world modeling (components connected with solid lines) and game plot modeling (components connected with dashed lines)

2) *Concepts Retrieval*: The component retrieves the interested concepts from the game scenario. For the characters, the concepts include the characters' emotions (sorrow, happiness etc.), behaviors (walking, sleeping, resting etc.) and characteristics (age, gender, personality etc.). For the game contexts, the concepts include the background story, time, place, objects, and world construction and so on. It provides the basis for the context modeling and the character modeling.
3) *Context Modeler*: It models the real-time simulation of the context variables which are retrieved from the game story scenario. It aims to build an immersive ecosystem that the virtual agents inhabit in. According to the scenario, some context variables might affect the interactions of the player and the virtual character in real-time. For example, when the weather in the game changes from sunny to rainy, the virtual character might need to find an umbrella before going out, and some of the character's activities (e.g. going to picnic) cannot be executed.
4) *Plot Modeling*: It constructs the game story plot from the story scenario. Then the plot will be assigned to the characters to perform individually.
5) *Character Modeler*: Each character modeler models and simulates a virtual character in the real-time, which includes the emotion modeling and the behavior modeling. Both the emotions and the behaviors of characters can change in real-time in correspondence to the stimulus. Take one scene of the story *"Little Red Riding Hood"* as an example, Little Red Riding Hood goes to grandma's house. The closer she is getting to the house, the happier she becomes. However, when the wolf appears, she becomes scared. When the wolf is getting closer, Little Red Riding Hood runs away.
6) *Character n*: It represents a virtual character involved in the game, which can be a non-player character (NPC) or a player.
7) *Game Engine*: It represents the low-layer game engine to support the graphics, sound, and artificial intelligence,

Digital Object Indentifier 10.1109/MCG.2009.80 0272-1716/$26.00 © 2009 IEEE

which exhibits the contexts and characters in the virtual environment. Some game engines employed by us are Activeworlds[1] and Torque Game Engine[2].

B. Game World Representation

According to the functionalities of the components, there are two main tasks in developing a serious game: 1) To create a believable game world, i.e. to model the characters and contexts; 2) To plan the activities of the characters according to the game scenarios. The main consideration of this paper is the first point, which is shown as the components connected with solid lines in Figure 1.

A game world has two main categories: characters and contexts. Therefore, game world modeling can be seen as a combination of character modeling and context modeling.

Characters is one of the most important ingredients of a successful story. Dynamic and believable virtual characters are important for players to gain an engaging experience. Two most important aspects of character modeling are emotion modeling and behavior modeling. Because the characters interact with the virtual environment in real-time, the behaviors and emotions of the virtual characters should evolve over time in different time schedules.

Take the simulation game "The Sims[3]" as an example, the variables to define an avatar includes:

- Emotion (happiness, sadness, boredom etc.)
- Behavior (eating, celebrating, dancing, chatting, sleeping, listening music, etc.)
- Property (tiredness, hunger, cleanness, wealth, etc.)

Besides the character variables, some context variables which affect the gameplay include:

- Time
- Friends Around
- House Decoration
- Foods Available etc.

When more dynamic concepts are involved, the gameplay would become more engaging. However, this could be the barrier to the devices with limited computing power. Therefore, there is a tradeoff between the gameplay experience and the scenario complexity. We can assign different priorities to the concepts in order to suit different levels of requirements.

For serious games in the virtual environments, the modeling of the characters and contexts is becoming increasingly complex, with the following properties:

1) *Complex Causal Relationships*: Causal relationships are complex, which include the mutual causal relationships between the characters and environment as well as the causal relationships between the emotions and behaviors.
2) *Dynamic*: The emotions and behaviors of characters keep on changing as the gameplay goes on. The characters need to respond rationally to the real-time simulation, by showing the correct behaviors and emotions.
3) *Randomness*: The virtual characters should not perform in a deterministic way. There should be some randomness of the variables.

In order to solve the problems, after the variables of interest (e.g. emotions and behaviors, contexts) are retrieved with concept retrieval component, E-FCM will be used to model the causal relationships among them, and start to run in real-time. This process will be shown in details in sections 3 and 4.

III. Game World Modeling with Evolutionary Fuzzy Cognitive Map (E-FCM)

Based on conventional Fuzzy Cognitive Map (FCM), we have proposed Evolutionary Fuzzy Cognitive Map (E-FCM) to simulate real-time variable states, and used it to model the dynamic and complex causal-related context variables [14].

In E-FCM, the concept states evolve in real-time, based on their internal mental states, external assignment, as well as external causalities. Moreover, the concepts update their internal mental states asynchronous with a small mutation probability.

E-FCM models every temporal state, which is named as *Evolving State* in the running process, as a collection of concept values.

Figure 2 shows an overview of E-FCM structure. The bounding box indicates the system enclosure (E), which comprises all the concepts and causal-related information. *Clock* is the reference time for the concepts to update their values, which is not explicitly considered by conventional FCM. The graph model of E-FCM is constructed with two main components: *concepts* and *causal relationships*. The definitions of concepts and causal relationships, as well as how they are initiated, are illustrated as follows.

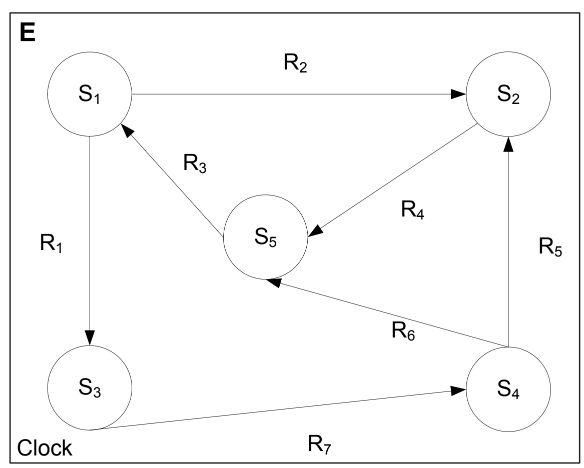


Fig. 2. Structure overview of E-FCM (E is the system enclosure, Clock is the reference clock, S_i is the concept and R_i is the causal relationship)

A. Concepts

Concepts represent the variables of interest in a real-time system. A concept is defined as a tuple of properties:

$$S = [S_V, T, P_s] \tag{1}$$

[1]Activeworlds: http://www.activeworlds.com
[2]Torque Game Engine: http://www.garagegames.com
[3]The Sims: http://thesims2.ea.com

Digital Object Indentifier 10.1109/MCG.2009.80 0272-1716/$26.00 © 2009 IEEE

where S_V is state value, T is state evolving time schedule, and P_s is state mutation probability.

1) State Value: S_V denotes the fuzzy value of the concept, which is the same as FCM. For simplicity, it uses a real value from [-1, 1] or [0, 1]. As different concepts have different unit scales, a concept value is represented on a relative scale over the concept standard value. For a system with N variables, the evolving state is represented as a vector $\mathbf{S_V}$:

$$\mathbf{S_V} = \begin{pmatrix} s_v^1 \\ s_v^2 \\ \vdots \\ s_v^i \\ \vdots \\ s_v^n \end{pmatrix} \tag{2}$$

where s_v^i is the value of concept i.

To model the characters in serious games, the emotions and behaviors are the concepts which need to be retrieved from the scenario first. In the story *"Little Red Riding Hood"*, the concepts might include: emotions ("happiness", "fear", "hesitation"), behaviors ("walking", "running", "calling for help", "collecting mushrooms"), and the related external contexts ("wolf nearby", "grandma nearby", "weather", "daytime").

The value of a concept is a qualitative description, which ranges from 0 to 1. Take the emotion "happiness" as an example, in the range of [0, 1], its value conveys that the character is " not happy at all", "somewhat happy" or "extremely happy".

2) State Evolving Time Schedule: In real-time, different variables might have different evolving time schedules, i.e. update time. With a system reference clock, all variables can update their states according to their own evolving time schedules. For a system with N concepts, it is represented as a vector $\mathbf{T}$:

$$\mathbf{T} = \begin{pmatrix} t^1 \\ t^2 \\ \vdots \\ t^i \\ \vdots \\ t^n \end{pmatrix} \tag{3}$$

where t^i is the evolving time schedule of concept i.

In the computer games, a "tick" is normally used as a time base to update the events. In our model, the evolving time schedule is calculated as number of "ticks". A commonly used time interval for one "tick" is 1 second. For example, we can assume that the refreshing rate for the emotions is 2 "ticks", i.e., they update every 2 seconds. The refreshing rate for the behavior "runaway" is 1 tick, as Little Red Riding Hood is very sensitive to the context changes and responds quite quickly (e.g., when the "wolf" is near).

3) State Mutation: Besides the causal effects from other variables, each concept will also alter its internal states randomly in real time. Different concepts might have different stabilities. The *stability* is modeled with a very small mutation probability, based on a uniform random number generator. If the probability is big, the system would become very unstable. For a system with N concepts, it is represented as a vector $\mathbf{P_s}$:

$$\mathbf{P_s} = \begin{pmatrix} p_s^1 \\ p_s^2 \\ \vdots \\ p_s^i \\ \vdots \\ p_s^n \end{pmatrix} \tag{4}$$

where p_s^i shows the self-mutation probability of the variable i.

In the simulation, the probability value used would be normally less than 0.1. The experiments show that the system becomes unstable when it is greater. To model the characters in the game, normally we put 0 for those variables that are stable in the scenario. For variables that might not be so stable, e.g. emotions, we can set a small value to simulate it.

B. Causal Relationship

Causal relationship R represents the strength and probability of the causal effect from one concept to the other concept. The uncertainty of the system variables can be twofold: *fuzziness* and *randomness* [15]. It is defined as the following tuple:

$$R = [W, S, P_m] \tag{5}$$

where W is weights of fuzzy causal relationships, S is signs of fuzzy causal relationships and P_m is probabilities of causal relationships.

1) Fuzzy Causal Relationships: The causal relationship between two variables, i.e., how much one variable will affect the other variable, is fuzzy. Some fuzzy terms are used, e.g. "much" and "a little". For a system with N variables, the fuzzy causal relationships of the system is represented as an $N \times N$ weight matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{pmatrix} w_{11} & \cdots & w_{1j} & \cdots & w_{1n} \\ w_{21} & \cdots & w_{2j} & \cdots & w_{2n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ w_{i1} & \cdots & w_{ij} & \cdots & w_{in} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ w_{n1} & \cdots & w_{nj} & \cdots & w_{nn} \end{pmatrix} \tag{6}$$

where *i* is the index of the causal concept, *j* is the index of the consequence concept, and w_{ij}, in the range of [0, 1], indicates the fuzzy weight of the casual relation under which the variable i affects the variable j. A higher value of w_{ij} implies a stronger causal relationship.

S in the definition of causal relationship shows the causal relationship is positive (+ve) or negative (-ve), which represents the increase of causal concept will lead to the increase or decrease of consequence concept respectively. It is usually combined with W in the computation.

Normally, the weight w_{ij} is determined by the expert. However, if there are available training datasets for the variables, the causal weights can be computed as the statistical correlation of the input data (changes in the causal variables)

Digital Object Indentifier 10.1109/MCG.2009.80 0272-1716/$26.00 © 2009 IEEE

and output data (changes in the consequence variable). More formally,

$$w_{ij} = \frac{Cov(i,j)}{\sqrt{var(i) \times var(j)}} \quad (7)$$

where $var(i)$ is the variance of the changes in variable i , $var(j)$ is the variance of the changes in variable j, and $Cov(i,j)$ is the co-variance of the changes in variable i and the changes in variable j.

For the purpose of modeling, it is important to determine the causal relationships between any pair of two variables. For example, the increase of "the wolf near" causes the decrease of "Little Red Riding Hood being happy" greatly. The appropriate value of the causality from "the wolf near" to "Little Red Riding Hood being happy" would be -0.8. By tuning the fuzzy weight, the impact might vary in the simulation accordingly.

2) Probabilistic Causal Relationships: The uncertainty of the causality is the conditional probability of one event over another event. Some terms are used to describe the causal probability, e.g. "always" and "often". For a system with N variables, it is represented as an $N \times N$ matrix $\mathbf{P_m}$ ($\mathbf{P_m}$ denotes the mutual causal probability):

$$\mathbf{P_m} = \begin{pmatrix} p_{1\to 1} & \cdots & p_{1\to j} & \cdots & p_{1\to n} \\ p_{2\to 1} & \cdots & p_{2\to j} & \cdots & p_{2\to n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ p_{i\to 1} & \cdots & p_{i\to j} & \cdots & p_{i\to n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ p_{n\to 1} & \cdots & p_{n\to j} & \cdots & p_{n\to n} \end{pmatrix} \quad (8)$$

where i is the index of the causal concept, j is the index of the consequence concept, $p_{i\to j}$ indicates the conditional probability of the relation under which the change in variable i causes the change in variable j. With prior knowledge, $p_{i\to j}$ can be calculated as P_{ij}/P_i statistically. Here, P_{ij} is the probability that both the causal concept i change and consequence concept j change happen and P_i is the probability that only the causal concept i change happens.

The causalities are not guaranteed to exist in the real world. For example, "The wolf near" might not decrease "Little Red Riding Hood being happy" all the time, but with a high likelihood. A causality probability 0.8 can be used to describe it. The higher the probability is, the more likely the causality will fire in the simulation.

C. Weight Adjustments with Interactive Evolutionary Computing

The causal weight and probability matrices are crucial in the modeling process of E-FCM. Expert knowledge is used to determine these matrices. However, due to the inevitable subjectivity in weight and probability assignment, these values may have small errors which accumulate over the iterations of inference. Therefore, the E-FCM may not always work as the expert expect. Worse still, because of the complexity of the inference process, it is not easy to identify and correct those small errors. The problem is further aggravated in our application because the intermediate inference results are important in order to create believable behaviors, which differs from traditional FCM inference where only the final result is needed. Hence, it is necessary to find a method to align the results of the E-FCM to our expectations by adjusting the weights and probabilities.

Towards this end, in this paper we employ the technique of Interactive Evolutionary Computing (IEC), originated from Dawkins' work [16]. The objective functions in traditional genetic algorithm or evolutionary strategies are replaced by subjective selection of human users. Believability, in our opinion, is a subjective measurement. It is difficult to determine numerically and objectively that which one is more believable among a number of scenarios. To cope with this problem, we resort to the subjective judgments of the expert once again. With IEC, the expert does not only design the matrices, but also examines the results produced and selects those resembling the real world more closely than others. The selected candidate solutions are then mutated and recombined to produce the next pool of candidates. The reproduction-selection loop repeats until the result is deemed by the expert as satisfactory, or a maximum number of iterations are completed.

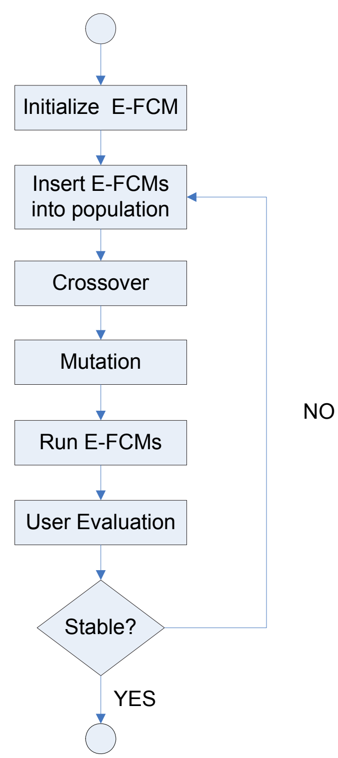


Fig. 3. Flow diagram of weight adjustments with interactive evolutionary computing

The flow diagram to adjust the weights with interactive evolutionary algorithm is shown as Figure 3. The process includes the following steps:

1) Initialize a single E-FCM by assigning the weights and probabilities according to expert knowledge.
2) Create a number of variations of the initial E-FCM by applying the mutation operator.
3) Use the crossover operator to create new E-FCMs. The

Digital Object Indentifier 10.1109/MCG.2009.80 0272-1716/$26.00 © 2009 IEEE

new weights are weighted sum of weights in E-FCMs at Step 2.

4) Mutate the weights with a small value.
5) Run each E-FCM for N times, summarize the results and present those results to the expert.
6) The expert selects several candidate E-FCMs as the parents of the next generation. If the running cycles are reasonable in the view of the expert, it is rated as "good", otherwise it is rated as "bad". Select the E-FCMs with top "good" rate.
7) If the "good" rate is higher than 90%, we consider the optimization process is stable, and the output E-FCM is the optimized version of E-FCM in Step 1. Otherwise, continue the optimization process from Step 2.

D. Pseudo-code of Running E-FCM in Real-Time

After the concepts and the causal relationships are confirmed, we are able to run the simulation with the E-FCM. Here is the pseudo-code for running the E-FCM in real-time.

Algorithm 1 E-FCM(n)

Require: Global clock t

Extract the relevant concepts into $concept\_list$
for all $Concept_i$ in $concept\_list$ **do**
  if $Concept_i$ at Update time, i.e $t\%T_i = 0$ **then**
    Update the concept value as
    $\Delta S_i^{t+T} = f(k_1 \cdot \sum_{j=0}^{n} \Delta S_j^t \cdot w_{i,j} + k_2 \cdot \Delta S_i^t)$ //addition subject to probability
    $S_i^{t+T} = S_i^t + \Delta S_i^{t+T}$ //self mutation with probability
  end if
  $t \Leftarrow t + 1$
end for

Some terms as used in the pseudo-code are defined as follows.

1) $f(\cdot)$: the activation function to regulate the state values, e.g. bipolar, tri-polar and logistics.
2) Variable State S_i^t: state value for the concept variable i at time t.
3) Variable State Change ΔS_i^t: state value change for concept variable i at time t.
4) Evolving Time Schedule T: time for concept i to update its value. Different concept may have different evolving time.
5) Time Slice t_0: an atomic time slice to update all the variables.
6) k_1 and k_2 are two weight constants.

Here, the summation of $\Delta S_j^t \cdot w_{i,j}$ is subjected to the conditional probability P_m^j, and the summation of ΔS_i^t is subjected to the self-mutation probability P_s. The computation complexity of the algorithm is $O(n^2)$, where n is the number of concepts.

E. Game World Modeling Procedure with E-FCM

The procedure to model the game world with E-FCM as shown in Figure 4, is presented in details below.

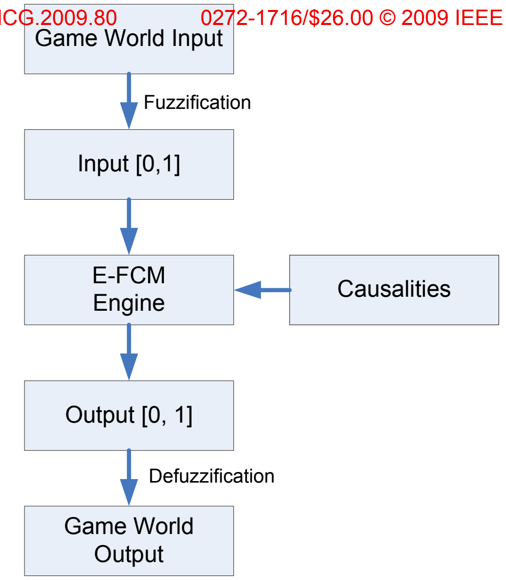


Fig. 4. Game World Modeling Procedure with E-FCM

1) Collect the relevant concepts (nouns or phases of descriptions) of game world, e.g., a) behaviors, emotions and attributes for characters and b) context variables. For example, a simple scenario is, "Little Red Riding Hood goes to the grandma's house. The sun rises, more mushrooms come out. Her happiness increases because she is able to collect a lot of mushrooms." Here, the concepts include "the altitude of the sun" and "mushroom growing", "Little Red Riding Hood being happy".
2) Pre-processing of the concept values: fuzzification or normalization, i.e. mapping the values into the range of [0, 1] with respect to their respective standard values. The concepts extracted from the game scenario use different units and different scales, e.g., "the altitude of the sun" (Concept A) can be expressed as the angle between the sun and the ground (with a value from 0 degree to 90 degrees), "mushrooms growing" (Concept B) is expressed as, "the number of mushrooms", which takes a value from 10 to 100. In order to represent the causal relationships between the two concepts, we need to map both them to a value between 0 and 1. It can be done by the fuzzification or normalization process. For concept A, 0 degree is mapped to 0; 90 degrees is mapped to 1 and t_0 degrees is mapped to $t_0/90$.
3) Calculate the evolving time schedule for each variable and self-mutation probability. Select a "tick" time of T for the serious game, e.g. 5 seconds, it is usually the minimum evolving time schedule of all the concepts in the game. Compute the evolving time schedule as t/T (t is the evolving time schedule in the game). If the expert observed the evolving time schedule of a concept in the game is 10 seconds, the evolving time schedule used in the E-FCM would be $10/5 = 2$.
4) Derive the necessary causal relationships (verbs) and connect the relevant variables with directed arcs. The

Digital Object Indentifier 10.1109/MCG.2009.80 0273-1716/$26.00 © 2009 IEEE

weight matrix normally is a sparse matrix. By reducing the unnecessary causal relationships, the calculations of the update process can be reduced.

5) Calculate the causal weights and probabilities (as shown in Section 3(C)).
6) Run the E-FCM to simulate the game world (as shown in Section 3(D).
7) Convert the concept values back to the real values in the game engine, i.e. defuzzification, which is a reverse operation of Step 2.

F. Comparisons to Other Computational Models

Rule-based models are by far the most commonly used methods to model the causal relationships. Nearly all the computational models use rules in the different ways. The advantage of Rule-based models is that, they are easy to construct and straightforward for simple systems. However, rule-based systems also have some limitations. If the system is complex, a large number of rules are required to be generated and the rules become complex. Moreover, the predefined rules might not be complete, thus the rule-based system does not work if the real-time state is new or the information required by the rules is incomplete.

Comparatively, E-FCM is an implementation of rule-based model, by embedding the rules into a graphical representation, which provides a straightforward way to represent the system with easy construction. An E-FCM model can be considered as a combination of FCM and Bayesian Network, as it models the causal relationships quantitatively with probabilities.

E-FCM is technically equivalent to FCM, when the probabilistic causal relationships and the evolving time schedules are not used. As FCM does not model the non-deterministic causal relationships, the same set of simulation outcomes are produced for each and every run of the gameplay. Virtual worlds modeled with FCM in [8] [9] are proven to be rational by the authors, but they are not adequate for serious games, as the players will be bored with the same world at each round. On the other hand, E-FCM is technically equivalent to Bayesian Network, when the fuzzy causal relationships are not considered. Bayesian network models the randomness of the causal relationships though belief, but does not do well to simulate the concepts quantitatively. As often in the games, players are more engaging with the numbers (e.g. health is 80 points) than the vague descriptions (e.g. health is low).

There are also some extensions to rule-based models to model the characters' emotions and behaviors. Among them, Fuzzy Logic Adaptive Model of Emotions (FLAME) is one well-established model by El-Nasr [17]. Similar to E-FCM, it involves fuzzy logic and probabilities in the computation. Fuzzy rules are used to compute the emotions and the behaviors in real-time, however, the generation of the fuzzy rules would be tedious at times as you need to consider the relationships among multiple causes and one consequence.

Comparatively, E-FCM is a generic model that is capable of modeling the causal relationships among the concepts in the real-time systems. It can be used to model not only the characters, but also the contexts in virtual world. Instead of a complex fuzzy rule which involves multiple causes and one consequence, it simply defines the causal relationship between any two concepts in the system with a clear graphical representation. As a result, it makes the construction easier and faster. Besides the enabling fuzzy and probabilistic causal relationship, the concepts are updated in an asynchronous way, which is required in the virtual world modeling.

IV. Case Study: Village of Mystery

In order to validate the E-FCM model, we illustrate this use with a case study based on a serious game on science learning, namely, the "Village of Mystery".

> "...The scientist John comes to a village for investigation of a mysterious disease. Villagers fall sick, but the disease has not been identified and no cure has been found. He needs to visit the village, talk to people around, do experiments and find some clues."

The game is designed for educational purpose, aiming to help the students learn different kinds of diseases from the exploration in the virtual world.

A. Experiment

In order to achieve a believable "John" with context awareness, the following causalities need to be modeled: how "John" is affected by the environment, e.g. water and mosquitoes, how the emotion of "John" changes as the story continues. To describe the elements in the gameplay, an E-FCM is constructed as shown in Figure 5. A total of nine concepts

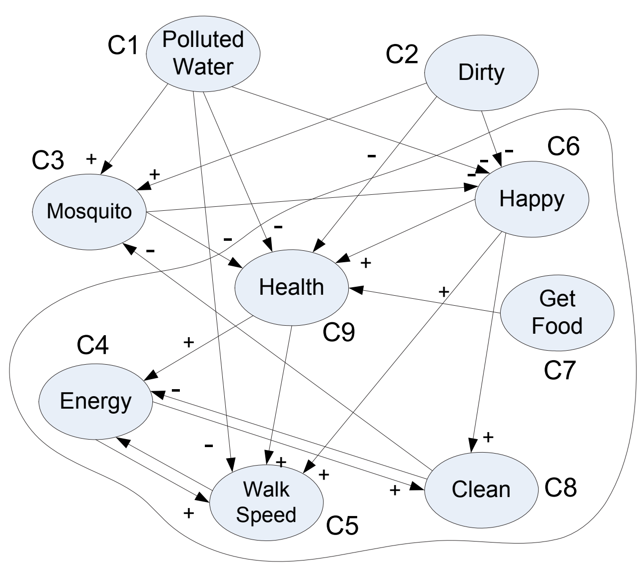


Fig. 5. The E-FCM model of the sample story scenario (The part inside the solid line shows the character model and the part outside shows the context model. '+' means "positive causal relationship" and '-' means "negative causal relationship".)

are extracted in the scenario, which are updated in real-time:

C_1: Polluted Water (0 - Not polluted water; 1 - Very polluted water)

C_2: Dirty (0 - Not dirty; 1 - Very dirty)

C_3: Mosquito (0 - No mosquitoes; 1 - Lots of mosquitoes)

Digital Object Indentifier 10.1109/MCG.2009.80 0272-1716/$26.00 © 2009 IEEE

W_{ij}		j								
		C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
i	C_1	0	0	0.6	0	-0.4	-0.4	0	0	-0.8
	C_2	0	0	0.4	0	0	-0.5	0	0	-0.8
	C_3	0	0	0	0	0	-0.8	0	0	-0.9
	C_4	0	0	0	0	0.8	0	0	0.8	0
	C_5	0	0	0	-0.7	0	0	0	0	0
	C_6	0	0	0	0	0.5	0	0	0.5	0.7
	C_7	0	0	0	0	0	0	0	0	1
	C_8	0	0	-0.9	-0.8	0	0	0	0	0
	C_9	0	0	0	0.5	0.8	0	0	0.6	0

TABLE I
WEIGHT MATRIX OF THE CAUSAL RELATIONSHIP

$P_{i \to j}$		j								
		C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
i	C_1	0	0	0.8	0	0.2	0.4	0	0	0.9
	C_2	0	0	0.6	0	0	0.5	0	0	0.8
	C_3	0	0	0	0	0	0.5	0	0	0.8
	C_4	0	0	0	0	0.8	0	0	0.7	0
	C_5	0	0	0	0.4	0	0	0	0	0
	C_6	0	0	0	0	0.6	0	0	0.5	0.6
	C_7	0	0	0	0	0	0	0	0	0.9
	C_8	0	0	0.8	0.7	0	0	0	0	0
	C_9	0	0	0	0.3	0.8	0	0	0.6	0

TABLE II
CONDITIONAL PROBABILITY MATRIX OF THE CAUSAL RELATIONSHIP

C_4: Energy (0 - No energy; 1 - Full of energy)
C_5: Walk Speed (0 - Stop walking; 1 - Walk fast)
C_6: Happy (0 - Not happy; 1 - Very happy)
C_7: Get Food (0 - Get no food; 1 - Get lots of food)
C_8: Clean (0 - Clean nowhere; 1 - Clean everywhere)
C_9: Health (0 - Totally sick; 1 - Very healthy)

Among the concepts, C_1 to C_3 are context variables that will affect the character's attributes and can be changed by the character. C_4 to C_9 are the character variables, including properties (energy and health), emotion (happy) and actions (walk, get food and clean).
The E-FCM can be constructed as two sub E-FCMs. The first E-FCM circled in Figure 5, represents the knowledge of the character; the second E-FCM, outside of the circle represents the knowledge of the contexts. The entire E-FCM model can be decomposed into four parts as follows:

- Internal of Character Model
- Internal of Contexts Model
- Characters' Actions over Contexts
- Contexts' Effects over Characters

Therefore, different domains of knowledge can be combined to suit more complex scenarios. This is an advantage inherited from FCM.
The weight matrix and the conditional probability matrix of the causal relationship are determined with experts' knowledge as shown in Table I and II respectively.

Take t_0 as a unit of evolving time, the evolving time for the nine variables are calculated as

$$\begin{pmatrix}1 & 3 & 1 & 1 & 2 & 3 & 2 & 3 & 1\end{pmatrix}$$

Here, we consider, the concepts "Dirty", "Happy" and "Clean" are updated in every three cycles; the concepts "Walking Speed" and "Get Food" update for every two cycles; and the rest of the concepts are updated in every cycle. Suppose the initial state vector representing the concepts $C_1, ..., C_9$ is encoded as

$$\begin{pmatrix}1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1\end{pmatrix}$$

In the virtual world, it shows as, 1) the water is polluted; 2) there are dirties around; 3) there are no mosquitoes around; 4) John starts with full of energy and health, fast walking speed and happy mood; and 5) John does not clean any place and does not get food.

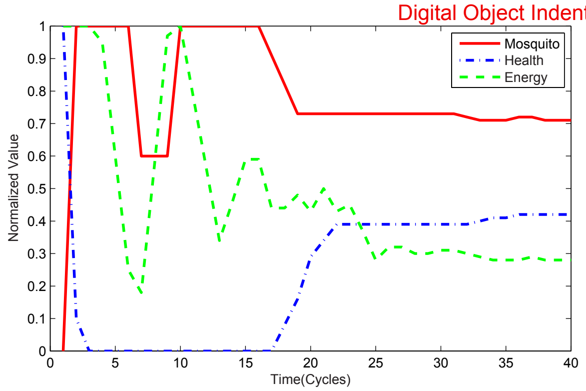
B. Results

1) Running results with E-FCM: Figure 6 (a)-(c) show the running cycles of the concepts "number of mosquitoes", "health" and "energy" in three rounds of experiments with E-FCM modeling respectively. The running results of the three concepts are represented with solid line, dot dashed line and dashed line respectively. There are no external assignments conducted over the concepts in the running process. As shown, the values of the concepts are updated asynchronously. Moreover, with the same initial vector, the running cycles are different in different rounds of experiments. This is due to the involvement of the probabilistic causalities as well as the small mutation probability of the concept values in the E-FCM model.
In addition, the causalities among the concepts are shown to meet our expectations. For example, when the "number of mosquitoes" increases, the "health" of the character drops; on the other side, when the "mosquito" decreases, the "health" of the character improves. The running activity patterns of "health" and "energy" also show the "causal increase" relationships between the two concepts. The lag time between the consequence concept and the causal concept depends on the evolving time schedules of the two concepts. As the evolving time schedules for the three concepts are all 1, the consequence concept will be updated one cycle after the causal concept, as observed clearly from the figure.

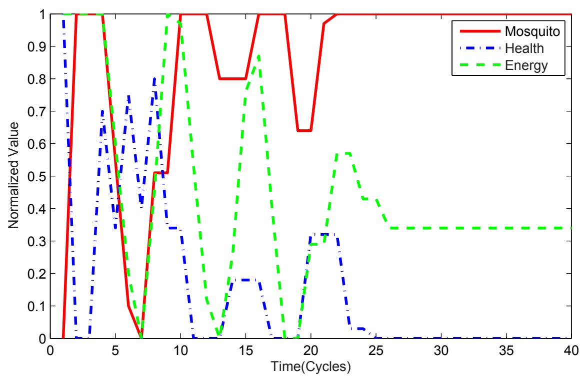
2) Effect of changing a concept value: As shown in Figure 6(b), the "health" of the character drops to a very low level. According to the E-FCM in Figure 5, one way to increase the "health" would be "cleaning the area" to reduce the "mosquitoes". Once the character in the virtual world assumes the behavior of "cleaning the area", the variable "clean" has a positive change to the value of 1.
The running cycles after the change of the concept "clean" are shown in Figure 7. We see that, the "health" of the character increases gradually. However, the "energy" of the character remains low. We thus conclude from the E-FCM that "cleaning the area" actually consumes the energy of the character greatly.

3) Effect of changing the evolving time schedule: As an important feature exhibited by E-FCM, the values of concepts are updated asynchronously. Figure 8 shows the running sequences when the evolving time schedule of the concept "health" changes from 1 to 5. It can be useful to model the scenarios, wherein different characters have different rates of health recovery.

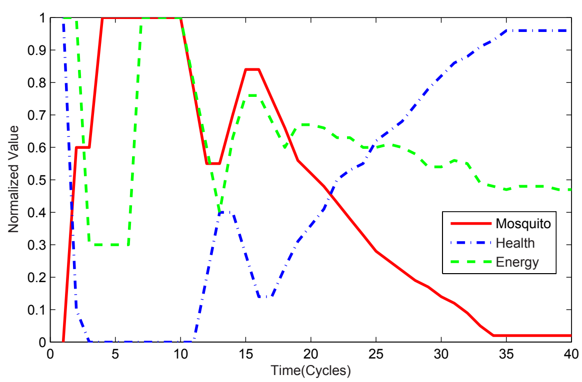
Digital Object Indentifier 10.1109/MCG.2009.80 0272-1716/$26.00 © 2009 IEEE



(a)



(b)



(c)

Fig. 6. Results of concepts “mosquito”, “health” and “energy”: (a) first round experiment of E-FCM; (b) second round experiment of E-FCM; (c) third round experiment of E-FCM.

4) *Running results with FCM:*

FCM, as a basis of E-FCM, provides a good foundation to model the causal relationships among a set of concepts quantitatively. To compare the simulation results of E-FCM with FCM, we model the variables with the same weight matrix and ignore the evolving time steps and the probabilistic causal relationship. The results of FCM model are depicted in Figure 9. Different from E-FCM model, all the states are updated in the same cycle. As shown in the figure, the variables reach the equilibrium loop quickly in around 5^{th} cycle. The running cycles are the same in every different rounds of the experiments when the initial states are

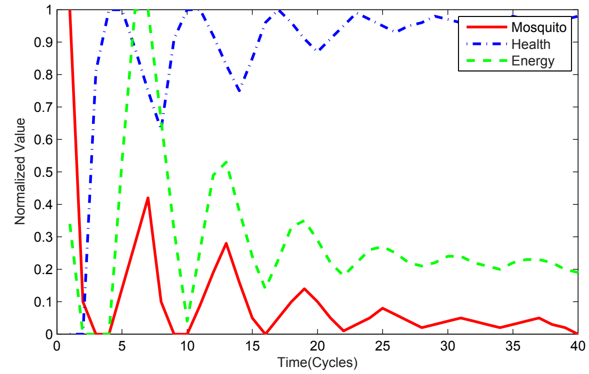


Fig. 7. Results after changing value of “clean” to 1 following simulation shown in Figure 6(b)

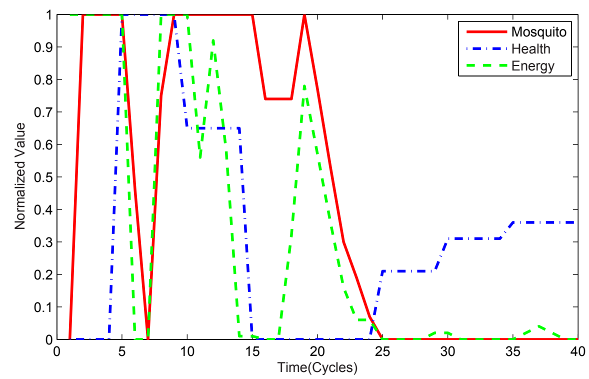


Fig. 8. Running results after changing time schedule of “health” from 1 to 5.

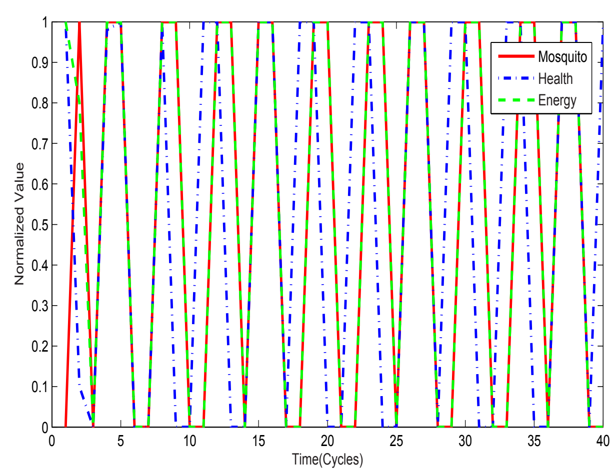


Fig. 9. Running results of the scenario with FCM

Digital Object Indentifier 10.1109/MCG.2009.80 0272-1716/$26.00 © 2009 IEEE

fixed. This shows a major limitation of FCM in modeling real-time world, as a player will experience the same set of events every time he/she plays the game.
In comparison, E-FCM allows the concept states to cover most combinations of the entire state vector space, i.e. the player can experience the dynamics of different virtual world.

5) Virtual World Implementation: In order to test the E-FCM model in the virtual world modeling, we implement the game world with the Torque game engine. Two screenshots are taken in the first round of the experiments as shown in Figure 10. The running sequence of the variables is shown in Figure 6(b). Figure 10(a) shows that, the character is in good health when he is far away from the dirty water initially; Figure 10(b) shows that the health level drops when the avatar approaches the dirty water and the mosquitoes. Different from traditional rule-based modeling, the variables evolve even when no rules are executed, which presents a more dynamic picture of the virtual world. As a result, the character is more believable.

(a)

(b)

Fig. 10. Screenshots at the first round experiment (a) Avatar with full health (b) Avatar with decreased health with dirty water and mosquitoes nearby

C. Evaluations

Evaluating the quality of user experience in a gameplay is a subjective matter. Based on Murray [18], user experience can be evaluated in terms of three key aspects, namely *immersion*, *agency*, and *transformation*.
Immersion is the feeling of involvement in the virtual environment, and the ability to interact with the environment. With the proposed model, the characters and the contexts are closely related once the causal relationships are defined. The affective computing increases the immersive experiences of the players.
Agency is the feeling that empowers the user to take actions in order to fulfill its intention. In our dynamic world, the player is able to carry out some actions over the contexts in order to achieve the goals. For example, as shown in last section, the player can increase its "health" by "cleaning the area".
Transformation refers to the variety of the world presentation. Different players may experience the game world differently as E-FCM simulates the virtual world and characters dynamically and stochastically. For example, once the "mosquito" increases, the "health" of the character drops; after the player "cleans the area", the "mosquito" decreases and the "health" of the character increases. The detailed evaluations based on the subjective metrics will be conducted in the future work.

D. Discussions

As an inference tool, FCM produces the desired inference results after the model stabilizes. More importantly, E-FCM shows the evolution of the states in real-time. This is essential as a simulation tool for modeling real-time characters and contexts.
Compared with FCM, E-FCM has the following improvements:

1) It allows a different update time schedule for each variable. For example, the value of "actions" can be changed faster than that of "emotions" in a gameplay.
2) It enables the self mutation of the context variables, which presents the dynamics of the world variables as evolving behavior.
3) It involves the probabilistic causality among the variables, which reflects realistic relationships among the concepts, and adds more dynamics to the character as a result, i.e., the character will not act in a deterministic way.

Though E-FCM is similar to other extensions for FCM with the concepts of evolving strategy and probabilistic events, E-FCM models the entire process of emotion and behavior evolution as a simulation engine. Therefore, the evolution of the state vectors are the main concerns rather than the equilibrium state vectors, as each evolving time state shows a state of the system in real-time. This is important for describing a believable virtual environment, and for the intelligent agents to make decisions in real-time.

V. Conclusions and Future Work

In this paper, Evolutionary Fuzzy Cognitive Map (E-FCM) is used for the first time to model the dynamic variables of virtual world in the serious games. Beyond the fuzzy causal relationships modeled in FCM, the probabilistic causal relationships among the variables are modeled, and the variables update their states with respect to individual time schedules. By modeling the causal relationships among the dynamic variables, characters and contexts are dynamic and believable, so as to ease the effort in presented, which provide the players a more engaging experience. In the future work, we shall explore the automatic methods for learning non-linear causal relationships, as represented by the weight matrix and the probability matrix, so as to ease the effort in model construction.

Digital Object Indentifier 10.1109/MCG.2009.80 0272-1716/$26.00 © 2009 IEEE

References

[1] D. R. Michael and S. L. Chen, *Serious Games: Games That Educate, Train, and Inform*. Muska & Lipman/Premier-Trade, 2005.
[2] M. Zyda, "From visual simulation to virtual reality to games," *Computer*, vol. 38, no. 9, pp. 25–32, 2005.
[3] C. D. Elliott, "The affective reasoner: a process model of emotions in a multi-agent system," Ph.D. dissertation, Evanston, IL, USA, 1992.
[4] M. S. El-Nasr, J. Yen, and T. R. Ioerger, "Flame—fuzzy logic adaptive model of emotions," *Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 3, pp. 219–257, 2000.
[5] J. Gratch and S. Marsella, "Tears and fears: modeling emotions and emotional behaviors in synthetic agents," in *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*. New York, NY, USA: ACM, 2001, pp. 278–285.
[6] J. D. Velásquez and P. Maes, "Cathexis: a computational model of emotions," in *AGENTS '97: Proceedings of the first international conference on Autonomous agents*. New York, NY, USA: ACM, 1997, pp. 518–519.
[7] B. Kosko, "Fuzzy cognitive maps," *International Journal of Man Machine Studies*, vol. 24, pp. 66–75, 1986.
[8] J. A. Dickerson and B. Kosko, "Virtual worlds as fuzzy cognitive maps," in *Fuzzy Engineering*, B. Kosko, Ed. New Jersey: Prentice Hall, Englewood Cliffs, 1994.
[9] J. T. Marc Parenthoen, Patrick Reignier, "Put fuzzy cognitive maps to work in virtual worlds," in *The 10th IEEE International Conference on Fuzzy Systems*, vol. 1, Australia, 2001, p. 38.
[10] Y. Miao, Z. Liu, and C. Miao, "Dynamical cognitive network, an extension of fuzzy cognitive maps," *IEEE Transactions on Fuzzy Systems (IEEE TFS)*, vol. 9, pp. 760–770, 2001.
[11] H. Song, Z. Shen, C. Miao, Z. Liu, and Y. Miao, "Probabilistic fuzzy cognitive map," in *IEEE International Conference on Fuzzy Systems*, 2006, pp. 1221– 1228.
[12] J. Carvalho and J. Tom, "Rule based fuzzy cognitive maps – qualitative systems dynamics," in *Proceedings of the 19th International Conference of the North American Fuzzy Information Processing Society*, Atlanta, US, 2000, pp. 407–411.
[13] N. Mateou, A. Andreou, and C. Stylianou, "Evolutionary multilayered fuzzy cognitive maps: A hybrid system design to handle large-scale, complex, real-world problems," *Information and Communication Technologies*, vol. 1, pp. 1663–1668, 2006.
[14] Y. Cai, C. Miao, A.-H. Tan, and Z. Shen, "Context modeling with evolutionary fuzzy cognitive map in interactive storytelling," in *IEEE International Conference on Fuzzy Systems, WCCI 2008*, Hongkong, China, 2008, pp. 2320–2325.
[15] B. Kosko, "Fuzziness vs. probability," *International Journal of General Systems*, vol. 17, no. 2, pp. 211–240, 1990.
[16] R. Dawkins, *The Blind Watchmaker*. New York, NY, USA: W. W. Norton & Company, Inc., September 1986.
[17] M. S. El-Nasr, J. Yen, and T. R. Ioerger, "Flame—fuzzy logic adaptive model of emotions," *Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 3, pp. 219–257, 2000.
[18] J. Murray., "Hamlet on the holodeck." *Cambridge, MA: MIT Press*, 1998.

Yundong Cai received a B.Eng. with second upper class honors from School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include artificial intelligence for planning and believable characters in interactive virtual storytelling and games.

Dr. Chunyan Miao is an Assistant Professor and Deputy Director of Emerging Research Lab, School of Computer Engineering, Nanayang Technological University Singapore. She received her PhD from the same university and did her postdoc research at Simon Fraser University Canada. Her research interests include artificial intelligence, software agent, multi-agent system and real world agent systems especially agent augmented digital media (web, game, virtual world and mobile communication) in different application domains.

Dr. Ah-Hwee Tan received the B.S. degree (First Class Honors) and the M.S. degree in Computer and Information Science from the National University of Singapore and the Ph.D. degree in Cognitive and Neural Systems from Boston University. He is currently an Associate Professor and the Head of Division of Information Systems at the School of Computer Engineering, Nanyang Technological University. Prior to joining NTU, he was a Research Manager at the A*STAR Institute for Infocomm Research (I2R), responsible for the Text Mining and Intelligent Agents programmes. His current research interests include biologically-inspired cognitive systems, information mining, media fusion, machine learning, and intelligent agents.

Dr. Zhiqi Shen , Visiting Assistant Professor, is with School of EEE, NTU. He has extensive experiences in transfer the research results into commercialized technologies. He successfully led a number of well known projects in Singapore and Canada and transferred the research outcome into commercial applications. His research interests include agent technology, computational intelligence, e-learning, mixed reality, semantic web/grid and interactive storytelling/games.

Boyang Li received a B.Eng. with first-class honors from School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include evolutionary optimization, organization of multi-agent systems, and interactive narration generation.