

Guided Game-Based Learning Using Fuzzy Cognitive Maps

Xiangfeng Luo, Xiao Wei, and Jun Zhang

Abstract—Fuzzy Cognitive Maps (FCMs) can be used to design game-based learning systems for their excellent ability of concept representation and reasoning. However, they cannot 1) acquire new knowledge from data and 2) correct false prior knowledge, thus reducing the game-based learning ability. This paper utilizes *Hebbian Learning Rule* to solve the first problem and uses *Unbalance Degree* to solve the second problem. As a result, an improved FCM gains the ability of self-learning from both data and prior knowledge. The improved FCM, therefore, is intelligent enough to work as a teacher to guide the study process. Based on the improved FCM, a novel game-based learning model is proposed, including a teacher submodel, a learner submodel, and a set of game-based learning mechanisms. The teacher submodel has enough knowledge and intelligence to deduce the answers by the improved FCM. The learner submodel records students' study processes. The game-based learning mechanism realizes the guided game-based learning process with the support of the teacher submodel. A driving training prototype system is presented as a case study to present a way to realize a real system based on the proposed models. Extensive experimental results justify the model in terms of the controlling and guiding the study process of the student.

Index Terms—Game-based learning, Fuzzy Cognitive Maps, teacher model, learning guidance, interactive computing.

1 INTRODUCTION

THE first computer game was developed in the late 1960s and it was not long before computer games were also used and developed for educational purposes [1]. But the term game-based learning¹ was not coined until 2000, when Prensky first discussed it in detail [2]. After that, game-based learning systems started to be developed according to various theories and methods [3], [4], [5].

In terms of teaching and studying, good learning demands teachers and students to make joint efforts. For students, their interest plays such an important role in their studying [6] that they must keep interested to exert their full potential. For teachers, it is concluded that any attempt to improve students' achievements should be based on the development of effective teaching behaviors [7]. In other words, they should give students appropriate guidance.

As a new study mode, game-based learning processes do well in arousing students' interest as students can be attracted easily in the form of gaming. On the other hand, because learning guidance is a creative and flexible job, it is difficult for some teachers to guide the learning process effectively, let alone in a game-based learning system. Therefore, there is still significant work needed to be done

in order to improve the guidance ability of game-based learning. In this paper, we apply Fuzzy Cognitive Maps (FCMs) to address the aforementioned problem.

An FCM has excellent concept representation and reasoning ability, which makes it widely used in various fields, such as geographic information systems [8], tacit knowledge management [9], and virtual worlds [10].

But the original FCM lacks self-learning abilities due to two shortcomings. One is that it cannot acquire new knowledge from data and the other is that it cannot correct false prior knowledge. If an FCM can be improved to address the above two issues, it can be more suitable for game-based learning system design. The basic idea of this paper is to improve FCM by overcoming the two aforementioned shortcomings and equipping it with the ability of self-learning and knowledge acquisition from both data and prior knowledge. Specifically, we utilize the *Hebbian Learning Rule* to enable an FCM to acquire new knowledge and the *Unbalance Degree* to correct false prior knowledge in an FCM automatically.

Based on the improved FCM, a game-based learning model is proposed which focuses on how to improve the game-based learning system's guidance ability. A driving training prototype system is developed as a case study to present a way to realize an actual system using the proposed models. The system provides a game-based learning environment for driving training. Students can study driving skills and traffic laws in the system. The driving scenario is generated including roadway geometry, representations of interactive traffic and pedestrians, and roadside object representations, among others. The guidance actions are generated by the teacher submodel, and the driver's actions are collected and judged by our proposed model. With the system, the experimental results are collected and demonstrate that our system is effective and the proposed model is valid.

1. Learning falls into two categories in this paper. One is human learning and the other is machine learning. In order to avoid confusion with these two categories of learning, we use the word "studying" to represent human learning (except for game-based learning) and "learning" to represent machine learning.

• The authors are with the High Performance Computing Center, School of Computer Engineering and Science, Shanghai University, Room 303, Xingjian Building, No. 149 Yanchang Rd, Shanghai 200072, P.R. China. E-mail: {luoxf, xwei, zhangjun_haha}@shu.edu.cn.

Manuscript received 1 Dec. 2009; revised 16 Apr. 2010; accepted 30 July 2010; published online 17 Aug. 2010.

For information on obtaining reprints of this article, please send e-mail to: lt@computer.org, and reference IEEECS Log Number TLTSI-2009-12-0164. Digital Object Identifier no. 10.1109/TLT.2010.26.

The rest of this paper is organized as follows: In Section 2, related work is discussed. In Section 3, the FCM is improved and a game-based learning model is proposed based on the improved FCM. In Section 4, a case study of the proposed model is introduced. In Section 5, our game-based learning model is evaluated from objective and subjective aspects. Finally, the conclusion is reached in Section 6.

2 RELATED WORK

The related work falls into two main categories: how to improve the FCM's learning ability and how to design a game-based learning system.

On the first one, much work has been done on improving FCMs in various aspects. Elpiniki and Peter proposed a method to train an FCM, which is based on nonlinear Hebbian learning and the differential evolution algorithm [11]. Wojciech et al. proposed a novel parallel approach to the learning of FCMs in order to deal with large maps due to their high computational complexity [12]. Mateou et al. proposed two algorithms for a multilayer approach developed to expand the capabilities of FCMs, one is Multilayer FCM (ML-FCM) and the other is Enhanced Multilayer FCM (EML-FCM) [13]. Our research is different from the above work in that we utilize the *Hebbian Learning Rule* and the *Unbalance Degree* to improve an FCM, so as to make it able to learn by itself and acquire knowledge from both data and prior knowledge.

On the second one, there is also some work. To improve a distributed game-based system's performance, Ng et al. proposed a parallel architecture [14] and Chim et al. proposed a method of caching and prefetching [15]. Wu et al. focused on how to realize a special game-based learning system or a game-based framework [3], [4], [5]. Compared with these efforts, our research aims to propose a general game-based learning model instead of a special game-based learning system, so as to facilitate the design of other game-based learning systems.

3 GUIDED GAME-BASED LEARNING USING FCM

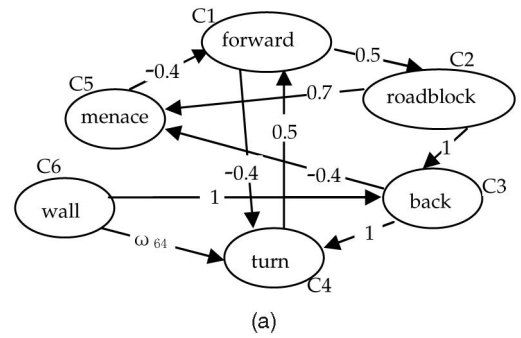
3.1 Fuzzy Cognitive Map

An FCM is a graphical model for causal knowledge representation and reasoning. It can represent not only causal relations between concepts but also knowledge of different granularity levels. An FCM comprises concepts (nodes) and the relations between concepts (edges). According to [8], the mathematic model of FCM is as follows:

$$V_{c_j}(t+1) = f\left(\sum_{\substack{i=1 \\ i \neq j}}^N V_{c_i}(t)w_{ij}\right), \quad (1)$$

where V_{c_i} and V_{c_j} are the state values of the cause concept C_i and the effect concept C_j , respectively, w_{ij} is the weight of the causal relation from concept C_i to concept C_j , and $f(x)$ is the threshold function of concept C_j .

Fig. 1a illustrates the FCM for robot high-level planning, where C_i is a concept with a state value. The state value can be a fuzzy value within $[0, 1]$ that represents the existent degree of a concept, or a bivalent logic in $\{0, 1\}$ that represents



$$E_1 = \begin{bmatrix} 0 & 0.5 & 0 & -0.4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0.7 & 0 \\ 0 & 0 & 0 & 1 & -0.4 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 \\ -0.6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \omega_{64} & 0 & 0 \end{bmatrix}$$

(b)

Fig. 1. (a) An FCM for robot high-level planning and (b) its matrix representation.

a concept's open/close state. The weight w_{ij} of an edge indicates the influence degree from the cause concept C_i to the effect concept C_j , which can be a fuzzy value within $[-1, 1]$ or a trivalent logic within $\{-1, 0, 1\}$. If the weight is positive, the increase/decrease of the state value of concept C_i leads to the increase/decrease of the state value of concept C_j . If the weight is negative, the increase/decrease of the state value of concept C_i leads to the decrease/increase of the state value of concept C_j . The adjacency matrix corresponding to the FCM is shown in Fig. 1b.

For a classic FCM, there are three major defects. First, an FCM is a closed system and lacks learning ability. Second, an FCM cannot acquire knowledge from the data automatically because it is overly dependent upon expert knowledge; thus, an FCM lacks the self-adaptation ability to a change of environment. Third, there is no global view of FCMs' behaviors because the behaviors are the outcomes of the interaction of concepts, and each concept has its own control subsystem.

The three defects are related. If an FCM is an open system, it should have the ability to acquire knowledge from data automatically. If an FCM has a global view, it should have the ability of adjusting the weights between concepts automatically in order to correct the false prior knowledge in an FCM.

In order to apply an FCM in game-based learning, the FCM should be improved to overcome the above three defects. So, the major work of this paper is:

1. how to make the FCM own the abilities of self-learning and knowledge acquisition,
2. how to build a guiding game-based learning model resting on an improved FCM,
3. how to realize an actual game-based learning system resting on the proposed model, and
4. how to verify the effectiveness of our proposed model.

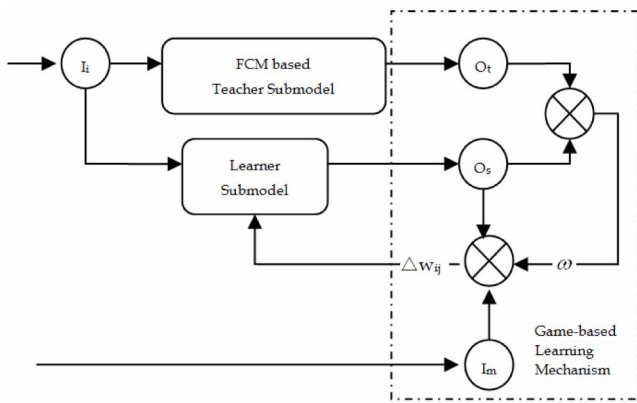


Fig. 2. Guided game-based learning model using fuzzy cognitive maps.

3.2 Guided Game-Based Learning Model

Game-based learning is different from computer games, as its objective is to improve a student's knowledge level. In particular, game-based learning needs a "teacher" to guide the student's study process. There are two issues to be discussed here. One is how the "teacher" is created, which we term as the teacher submodel. The other is how the "teacher" works, which is reflected by how the teacher submodel guides the student to study in a game-based learning environment.

To solve these two issues, we propose a game-based learning model using an FCM. As shown in Fig. 2, this model consists of three parts: a teacher submodel, a learner submodel, and a set of game-based learning mechanisms.

The teacher submodel based on the improved FCM has abundant knowledge and the ability of self-learning from both data and prior knowledge, which makes the FCM suitable for acting as a teacher. The submodel plays the role of monitor and guides the student's study process. Consequently, this game-based learning model has the teacher-guidance function. The construction and self-learning of the teacher submodel will be discussed in Section 3.3.

The study process of a student is recorded by the learner submodel. The learning mechanism consists of various parts to control the whole learning process. Both the learner submodel and the game-based learning mechanism will be discussed in Sections 3.4 and 3.5, respectively.

In Fig. 2, I_i is an input which represents the task or the scene of the game that the student will face, I_m is the real input of the student, O_t is the standard output of the teacher submodel, O_s is a real output of the learner submodel, ω is the difference between O_t and O_s , and Δw_{ij} is a variable to adjust the learner submodel.

3.3 Teacher Submodel

The core of the teacher submodel is a teacher FCM, the construction of which includes three steps as follows:

The first step is to build an initial FCM which stores the prior knowledge of the teacher and it is usually manually constructed by domain experts.

The second step is to study the relations between pairs of concepts in the FCM by the *drive-reinforcement Hebbian learning rule*. This step only focuses on certain parts of the concepts, so we call it the local learning of the teacher submodel, which will be discussed in Section 3.3.2.

The weight of a relation between a pair of concepts can be adjusted to an optimal FCM by local learning. But from the global viewpoint, the teacher submodel may not be optimal. So, the third step is the global learning of the teacher submodel, which aims to get a global optimal FCM.

Note that steps 2 and 3 are usually conducted by data training, which makes the FCM have the ability of automatic learning and knowledge acquisition from both data and prior knowledge. So, this method combines expert knowledge with data training's virtue.

3.3.1 Construction of the Teacher Submodel

The initial FCM of the teacher submodel is usually constructed by domain experts. Because domain experts have abundant prior knowledge of their domains, they can build the FCM effectively, which corresponds to the real world. This work involves finding the concepts in the model and the initial weight between concepts. These weights reflect the expert knowledge and are fine adjusted by using the local and global learning described in the next section, which makes them emulate the real world effectively.

3.3.2 Local Learning of the Teacher Submodel

Local learning of the teacher submodel is mostly to learn the weight value between the concepts of an FCM. We mainly use the classic *Hebbian Learning Rule* [16] and the *Drive-Reinforcement Hebbian Learning Rule* [17], [18] in the local learning process of the teacher submodel. The main steps are as follows:

1. Self-Learning Based on the Hebbian Learning Rule

Generally, for a common neural computing model $ncm^{(N)} = \langle C_G, A_G, IF, OF, WA, OA \rangle$, where C_G is the concepts set, A_G is the connection relation matrix, IF is the input set, OF is the output set, WA is the working algorithm, and OA is the organizing algorithm [19], we notate the output of random concept $C_i \in C_G (i = 1, 2, \dots, N)$ as V_{ci} and $V_{cj} \in \{0, 1\}$. According to Hebb supposition, the correction of connection weight r_{ij} between concept C_i and C_j is defined as

$$\begin{cases} \Delta r_{ij}(t) = \sigma V_{ci}(t)V_{cj}(t) & (\sigma > 0), \\ r_{ij}(t+1) = r_{ij}(t) + \Delta r_{ij}(t), \end{cases} \quad (2)$$

where t is the discrete time, that is, $t \in \{0, 1, 2, \dots\}$, σ is the learning factor, and $\Delta r_{ij}(t)$ is the correction value of r_{ij} at time t .

From (2), if presynaptic concept C_i and its postsynaptic concept C_j excite at the same time, that is, both the output $V_{ci}(t)$ of concept C_i and the output $V_{cj}(t)$ of concept C_j equal to 1, and the correction value (connection weight r_{ij}) $\Delta r_{ij}(t) > 0$, then the connection weight $r_{ij}(t+1)$ at time $t+1$ will be improved. At time t , if either the output $V_{ci}(t)$ of concept C_i or $V_{cj}(t)$ of concept C_j is zero, the correction value $\Delta r_{ij}(t) = 0$.

2. Self-Learning Based on the Drive-Reinforcement Hebbian Learning Rule

According to the drive-reinforcement algorithm and the *Hebbian Learning Rule* [17], [18], the correction of connection weight r_{ij} between concept C_i and C_j is defined as [18]

$$\begin{cases} \Delta r_{ij}(t) = \sigma \Delta V_{ci}(t) \Delta V_{cj}(t) & (\sigma > 0), \\ r_{ij}(t+1) = r_{ij}(t) + \Delta r_{ij}(t), \end{cases} \quad (3)$$

where $\Delta V_{ci}(t)$ is the change of the output $V_{ci}(t)$ of presynaptic concept C_i at time t .

For the concept of C_i , the change of its output $V_{ci}(t)$ at time t is

$$\Delta V_{ci}(t) = V_{ci}(t) - V_{ci}(t-1). \quad (4)$$

According to (3), if $\Delta V_{ci}(t)$ and $\Delta V_{cj}(t)$ have the same sign, that is, the changes from time $t-1$ to t of C_i and C_j are of the same sign, then the connection weight $r_{ij}(t+1)$ at the time of $t+1$ will be improved. On the contrary, if $\Delta V_{ci}(t)$ and $\Delta V_{cj}(t)$ have different signs, that is, the changes from time $t-1$ to t of C_i and C_j are of different signs, then the connection weight $r_{ij}(t+1)$ at the time of $t+1$ will be reduced.

3. Local Learning Algorithm

Every concept has some different influences on other concepts. Through the one-to-one learning between any two concepts, that is, the state value of a concept is adjusted singly on the condition that other state values are not changed, the influence on the other concept can be observed so as to learn the relation between the two concepts.

In the local learning cycles, the state value $V_{Ci}(t)$ of concept C_i is adjusted continuously. Meanwhile, the state value of the corresponding concept C_j also varies, and its changed value is denoted by $\Delta V_{Cj}(t)$. Also, the programming output of the concept C_j based on the expert knowledge also varies, and its changed value is denoted as $\Delta T_{Cj}(t)$. Thus, the error $\Delta \delta_j(t)$ is $\Delta T_{Cj}(t) - \Delta V_{Cj}(t)$. Besides, the changed value of concept C_i is denoted by $\Delta V_{Ci}(t)$, and the modifier Δw_{ij} between concepts C_i and C_j is then defined as

$$\Delta w_{ij}(t+1) = \sigma \Delta V_{Ci}(t) \Delta \delta_j(t) + \varepsilon \cdot \Delta \delta_j(t). \quad (5)$$

The algorithm of the local learning between the concepts is shown as follows:

Local Learning Algorithm:

Step 1: calculate $\Delta V_{Cj}(t)$, $\Delta T_{Cj}(t)$, $\Delta V_{Ci}(t)$;

Step 2: $\Delta \delta_j(t) = \Delta T_{Cj}(t) - \Delta V_{Cj}(t)$;

Step 3: $\Delta w_{ij}(t+1) = \sigma \Delta V_{Ci}(t) \Delta \delta_j(t) + \varepsilon \cdot \Delta \delta_j(t)$;

Step 4: if $round < m$ and $\Delta w_{ij}(t) > 10e-8$
 then $w_{ij}(t) = w_{ij}(t) + \Delta w_{ij}(t)$; go to Step 1;
 else go to Step 5;

Step 5: end.

In the algorithm, $round$ is the counter of learning cycles and m is the maximum of learning cycles. If $round$ reaches the maximum or $\Delta w_{ij}(t)$ is small enough, then the learning process will be completed.

In the local learning, from the local viewpoint of concepts, the teacher submodel can adjust the unreasonable weights of the relations in the FCM. However, because the learning is based on the local viewpoint, there may exist unreasonable points in the teacher submodel which need to be improved through the global learning.

3.3.3 Global Learning of the Teacher Submodel

Global learning of the teacher submodel refers to the global learning of the FCM itself. The construction of the FCM is from the top down and is used to represent prior knowledge, but the classic FCM has three major defects, as discussed in Section 3.1.

The dynamic behaviors of an FCM are realized through the interaction of its concepts, and the dynamic behaviors will then reach a fixed point, a limit cycle, or a chaos state. The positions of concepts in the FCM are equal and each has its own local viewpoint, but there is no global concept among all the concepts which deprive the global learning ability of the FCM. In this section, we discuss how to construct a global concept for the FCM so as to enable it to possess global learning ability.

In our approach, a virtual super concept is constructed, which does not belong to any of the other types of concepts. The state value of the virtual super concept may be regarded as the energy of the FCM. Before proceeding to subsequent discussions, the following three definitions are defined as follows:

Definition 1 (Error of concept's state values δ_j). The difference between the expectation output and the actual output of a concept's state value is denoted as δ_j , and the sum of δ_j represents the difference between the FCM and the real world. The definition of δ_j is given as

$$\delta_j(t) = E_{C_j}(t) - V_{C_j}(t), \quad (6)$$

where $V_{C_j}(t)$ is the actual output of C_i at time t and $E_{C_j}(t)$ is the expectation output of the concept C_i in the FCM at time t .

Note that the dynamic behaviors of the system are the result of interaction between a concepts set C_G and its state values set V_{CG} , which reflects the state value of each reasoning concept at time t . The interaction of concepts' state values and their weights will produce the simulation of the real world. So, the sum of δ_j is an import parameter that represents whether the FCM can simulate the real world.

By comparing the actual output with the expectation output of all the concepts, the differences between the FCM and the real world can be obtained. Specifically, let

$$\delta = \frac{\sum_{j=1}^n \delta_j^2}{n}, \quad (7)$$

where n is the number of concepts. If $\delta \approx 0$, the difference between the FCM and the real world is small and the FCM reflects the world factually.

Definition 2 (Concept weight adjusting variable Δw_{ij}). The adjusting variable of concept weight Δw_{ij} is to hold the adjustment value in order to keep the minimal change of weight in the learning process of the FCM. Δw_{ij} is defined as

$$\Delta w_{ij}(t+1) = \delta_j(t) \cdot w_{ij}(t) + \varepsilon \cdot \delta_j(t). \quad (8)$$

The weight adjustment of the FCM follows Linsker's maximum entropy principle [20], i.e., when the environment needs to change the weight of concepts, the change should be minimal, so it can mostly keep the original information.

In (8), ε is a random adjusting variable used for disturbing the learning process of the FCM. In order to prevent the FCM from being trapped in a local minimum point in the learning process, a random adjusting variable ε is set. If the weight of the FCM does not fit the expectation value after being trained several times, there may be an erroneous setting in the connection intensity of important

causality. Hence, ε should be increased in positive or negative scope.

Sometimes, the random adjusting variable ε is added, which might cause the FCM to be in an error mode. So, the random adjusting variable is modified to $\delta_j \cdot \varepsilon$ to remove this fault. This modification can correct the error mode of the FCM and hold its right mode.

Definition 3 (Unbalance degree h). *The unbalance degree h is a measurement of the difference between the FCM and the real world at time t , and is defined as*

$$h(t) = \sum_i \sum_j \Delta w_{ij}(t) (\delta_j^2(t) + \varepsilon \cdot \delta_j(t)), \quad (9)$$

where ε is a random adjusting variable.

Under the unbalance degree h , the closer the state value of the FCM is to the real world, the more effective the adjustment it produces, so it will be a better adjustment. The same applies to the opposite situation too. The adjustment can be completed when the system touches the minimal unbalance degree.

After Δw_{ij} is gained and w_{ij} is adjusted, the value of h should be minimal. When h reaches a specified value, the adjustment process can be terminated. A global concept of the FCM can control the learning process of the FCM.

After the combination of the concept states is given, the real output is compared from a global viewpoint with the expectant output of the teacher submodel, so as to learn all the relationships of the concepts as a whole.

Let V_{C_j} and E_{C_j} denote the real output and the expectant output of the teacher submodel, respectively, $\delta_j(t)$ denote the error at time t , and $\Delta w_{ij}(t)$ denote the modifier of the relationship weight between concepts C_i and C_j at time t . The global learning algorithm of teacher submodel is as follows:

Global Learning Algorithm

Step 1: calculate $E_{C_j}(t)$, $V_{C_j}(t)$;

Step 2: $\delta_j(t) = V_{C_j}(t) - E_{C_j}(t)$;

Step 3: $\Delta w_{ij}(t+1) = \delta_j(t) \cdot w_{ij}(t) + \varepsilon \cdot \delta_j(t)$;

Step 4: $h(t) = \sum_i \sum_j \Delta w_{ij}(t) (\delta_j^2(t) + \varepsilon \cdot \delta_j(t))$;

Step 5: if round $< m$ and $h > 10e-8$

then $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$; go to Step 1;

else go to Step 6;

Step 6: end

After the local and global learning process, the teacher submodel is revised and optimized, and the final one can be used to guide the study process of the student.

3.4 Learner Submodel

In Fig. 2, the learner submodel is depicted as an improved FCM too, which has the same structure the teacher submodel FCM. The learner submodel has all or part of the concepts of the teacher FCM, according to the concepts that the student plans to grasp.

Although the learner submodel, like the teacher submodel, is an FCM, they are quite different. First, the teacher submodel has expertise while the learner submodel does not. Second, during the study process, the teacher submodel is static while the learner submodel's knowledge is changing dynamically. Third, the change of the learner submodel is

under the guidance of the teacher submodel. The constructive processes of the learner submodel are as follows:

1. Storing Student's Knowledge

One important function of the learner submodel is storing the student's knowledge. Before studying, the learner submodel is a zero matrix, which means that the student does not have any knowledge or that his knowledge has not been recorded in the learner submodel.

In the study process of a student, the learner submodel's FCM approaches the teacher submodel, in whole or in part, which implies that the student's knowledge is being improved.

2. Output of the Learner Submodel

In Fig. 2, I_i is the input which represents the task or the scene of the game that students will meet. O_s is the output of the learner submodel.

$$O_s = I_i \times sE. \quad (10)$$

In (10), both I_i and O_s are the vectors of $1 \times n$ and sE is the adjacency matrix of the learner submodel FCM, which is an $n \times n$ dimension matrix. Equation (10) indicates that, with the current knowledge of the learner (sE), action (O_s) should be taken in the scene (I_i). Comparing the learner and teacher submodel outputs, the difference (ω in Fig. 2) can be estimated.

3. Adjustment of the Learner Submodel

The difference ω between the teacher and learner submodel outputs results from the difference of their FCMs, which further reflects the difference in knowledge between the teacher and student. Throughout the study process, the learner's real input I_m may be better or worse than the learner submodel output. According to I_m , ω , and O_t , the adjustment Δw_{ij} is calculated, which is used to modify the learner submodel by

$$sE_{ij} = sE_{ij} + \Delta w_{ij}. \quad (11)$$

After this adjustment, the learner submodel may be improved or worsened, which can be estimated by ω in the next round of the study process.

3.5 Game-Based Learning

Before the game-based learning starts, the teacher submodel has prior knowledge but the learner submodel has no prior knowledge. The goal of the learning is that the learner's knowledge should be as similar to the teacher's as possible after the end of the study process.

With respect to Fig. 2, before starting the learning process, the weight of each side in the learner FCM is zero. That is, the FCM's adjacency matrix of the learner submodel is a zero matrix, indicating that the student has no knowledge about it. It is expected that the weights of the FCM in the learner submodel will be adjusted to become similar to that of the teacher submodel afterward.

In Fig. 2, I_i is an input which represents the task or the scene of the game that the student will face. I_i will be sent to both the teacher and the learner submodel, O_t is an ideal output of the teacher submodel and is considered as the standard, and O_s is a real output of the learner submodel. The difference between O_t and O_s is notated as ω , which is the terminated condition of the game-based learning. If ω is

small enough, it means that the knowledge of the student is close enough to the teacher's that the study process can be terminated. Otherwise, it means that the student has not mastered the concept well and needs to continue studying. I_m is the real input of the student, which means the student's real action in the game process under I_i . I_m may be right or wrong. According to I_m , O_s , I_i , and O_t , the adjusting variable Δw_{ij} can be computed and used to adjust the learner submodel.

Combing the *unsupervised Hebbian learning with supervised δ learning rule*, the supervised Hebbian learning defined in [21] is as follows:

$$\Delta w_{ij}(t) = \eta \times (d_j(t) - O_j(t)) \times O_j(t) \times O_i(t), \quad (12)$$

where η is a learning factor, $O_i(t)$ is the input, $O_j(t)$ is the real output, $d_j(t)$ is the expectation output, and $d_j(t) - O_j(t)$ is called the "teacher signal."

Applying (12) to our game-based learning mechanism, Δw_{ij} can be calculated by

$$\Delta w_{ij}(t) = \eta \times (O_t(t) - O_s(t)) \times (I_m(t) - O_s(t)) \times I_i(t). \quad (13)$$

The difference between (12) and (13) lies in the fact that there are two teacher signals in (13), i.e., $O_t(t) - O_s(t)$ and $I_m(t) - O_s(t)$. The former is a teacher signal evaluating the effect of the study process, and the latter is another teacher signal representing the difference between the student's real action and learner submodel's real output. Both of the two signals can be used to guide the adjustment of the learner submodel. The following algorithm summarizes how the game-based learning works:

Game-based Learning Algorithm

Step1: Input $I_i(t)$

Step2: $O_t(t) = I_i(t) \times tE$

Step3: $O_s(t) = I_i(t) \times sE$

Step4: $\omega = O_t(t) - O_s(t)$

Step4: $\Delta w_{ij}(t) = \eta \times (O_t(t) - O_s(t)) \times (I_m(t) - O_s(t)) \times I_i(t)$

Step5: If $\omega < \varepsilon$ then goto step 6 else goto step 1.

Step6: end.

In the algorithm, tE is the adjacency matrix of the teacher submodel, sE is the adjacency matrix of the learner submodel, and ε is a threshold. When ω is less than ε , we regard that the student has mastered the knowledge and the study process can be finished.

3.6 Game Evaluation

Students can game in the system time after time. During the study process, guidance will be given to the students when their operations are not right. Corresponding to the matrix E of the learner submodel, matrix E' is set, which has the same elements as E and each element E_{ij} records the study effect of a concept. The average error of w_{ij} represents the study effect:

$$E_{ij} = \frac{1}{m} \sum_{t=1}^m (sw_{ij}(t) - tw_{ij}(t)), \quad (14)$$

where m is the study time and $sw_{ij}(t)$ and $tw_{ij}(t)$ are the weights of the learner submodel and the teacher submodel at time t , respectively.



Fig. 3. The system gives tasks according to the current environment.

3.7 Reward and Punishment

In Fig. 2, ω is the difference between the submodel outputs of the teacher and learner for the same input. That is, it is the difference between the student's real action and the anticipated action in the game scene. According to ω , we can design the reward and punishment mechanism for the game sensibly, the details of which are omitted here due to space limitations.

4 DRIVING TRAINING PROTOTYPE SYSTEM

In order to test this model's effectiveness, we developed a prototype system for studying driving. It provides a game environment for driving training using the proposed game-based learning model. In the game process, students can study approaches for following automobile and traffic laws while playing games.

4.1 Overview

The system was developed in Microsoft Visual Studio .Net 2005 with C# language. The system can be executed on Microsoft Windows family operating systems with .Net framework 2.0 or above versions' support.

All of the knowledge and rules that the system needs are derived from Chinese traffic laws. To acquire the knowledge, one of the students of our group attended classes in a driving school for two months. We also communicated with some driving school teachers and senior drivers many times. All of this knowledge is represented in the teacher submodel which will be discussed in Section 4.2.

It is difficult to develop a 3D system and it requires a lot of work, so instead, we developed a prototype system, as shown in Figs. 3, 4, and 5. It works based on scene changes. With this system, a car can be operated and controlled, but the scene shown to the student is generated by the system according to the student's actions. Although it is a flaw that the cartoon of the car moving is not shown, it is enough for the students to study driving in the real scenes and it is also enough for us to collect the student's actions to validate our model.

In the system, the driving student first selects the concept he/she will try to study. According to the selected concept, the system generates a scene (See Fig. 3) for the concept. Through the FCM's reasoning of the teacher submodel, the



Fig. 4. Teacher submodel gives guidance in study process.

system knows the right answers and actions he/she should take. In the game process, the student's real actions are gathered and sent to the learner submodel. During the study process, if the student needs guidance, he can press button F1 and the system will give guidance for this task to him (see Fig. 4). After being evaluated by the model, the evaluation result is shown to the student and advice is given to the student at the same time if possible (see Fig. 5).

4.2 Teacher Submodel Construction

The intelligence and the study guidance of the system are based on the teacher submodel to a great extent. For this system, the challenge is how to construct the teacher submodel. First, the teacher submodel should contain the basic driving knowledge which can be realized by building the initial teacher submodel. Second, the error or conflict driving knowledge should be overcome and the absent or implicit knowledge should be mined. All of these are realized by the self-learning of the teacher submodel.

4.2.1 Constructing the Initial Teacher Submodel

The initial teacher submodel for the system (Fig. 6) is built with the prior knowledge from the experts of the driving school and some senior drivers who have driven over tens of years. The initial teacher submodel, represented by the FCM, consists of concepts and relationships between them. Concepts are formed in terms of the car states, operations, and situations of the road lane when driving. The relationships among them are represented by edges. For instance, $C_i \xrightarrow{w_{ij}} C_j$ means that concept C_i has an influence on concept C_j . Generally, a state of a car can be affected by multiple operations or situations of the lane. For example, *upgrade*, *downgrade*, *throttle*, and *brake* all have an influence on the car *speed*. Herein, we use w_{ij} to express the degree of the effect from concepts C_i to C_j .

The following four steps are essential to construct the initial teacher submodel:

Step 1. Determine basic concepts in driving.

Through discussion with the experts of the driving school and some senior drivers, 48 basic concepts were selected, which are described as the nodes in Fig. 6. These nodes are as follows:



Fig. 5. After study is finished, student's actions are evaluated and the advices are given by the teacher submodel.

C_1 : Steering Wheel, C_2 : Left-Turn Indicator, C_3 : Horn, C_4 : Clutch, C_5 : Driving Brake, C_6 : Throttle, C_7 : Parking Brake, C_8 : Gears, C_9 : Speedometer, C_{10} : Fog Lamp, C_{11} : Clearance Lamp, C_{12} : Hazard Warning Lamps, C_{13} : Low Beam, C_{14} : High Beam, C_{15} : Urban Road, C_{16} : Slow Sign, C_{17} : Stop Sign, C_{18} : Tunnel, C_{19} : Upward Trail, C_{20} : Downhill, C_{21} : Right Turn Indicator, C_{22} : Fast Traffic Lane, C_{23} : Slow Traffic Lane, C_{24} : Crossroads, C_{25} : Heavy Road, C_{26} : Snowy and Icy Road, C_{27} : Rainy Road, C_{28} : Flooded Road, C_{29} : Speed Up Sign, C_{30} : Crosswind, C_{31} : Tire Puncture, C_{32} : Steering Failure, C_{33} : Braking Failure, C_{34} : Collision, C_{35} : Foggy Weather, C_{36} : Congestion, C_{37} : Left-Hand Bend, C_{38} : Right-Hand Bend, C_{39} : U Turn, C_{40} : Night Driving, C_{41} : Immovable Obstruction, C_{42} : Human, Livestock, Nonmotor Vehicle, C_{43} : Rear-End Collision Danger, C_{44} : Traffic Light (Yellow to Red), C_{45} : Traffic Light (Yellow to Green), C_{46} : Speed Limit Sign, C_{47} : Sidewalk, C_{48} : Getting Start.

Step 2. Find the relation between concepts.

By analyzing each pair of concepts, if one concept influences another, there exists a relationship between them. Correspondingly, there exists an edge between the pair of concepts in Fig. 6. Fig. 6 is a directed graph and the arrows of edges show the influence direction.

Step 3. Determine the weight of relation.

This step determines the degree of influence of a pair of concepts, namely, the weight of relation. The weights fall into three types as follows:

The first type is a constant weight value in the range of $[0, 1]$. This type of relation does not vary with the state values of other concepts.

The second type is a condition dependence whose weight is related to the state value of other concepts and the value is binary: 1 or 0. For example, the means of $P(1|V(C_2) \leq 0, V(C_4) \geq 0)$ is that if $V(C_2) \leq 0$ and $V(C_4) \geq 0$, then $P = 1$; otherwise, $P = 0$.

The third type is a function whose weight is also related to the state value of concepts. To different state values, relevant weights are various. For example, the means of $P(V(C_3), V(C_5))$ is that if $V(C_5) - V(C_3) > 0$, then $P = -0.2$; if $V(C_5) - V(C_3) < 0.2$, then $P = 0.2$; otherwise, $P = 0$.

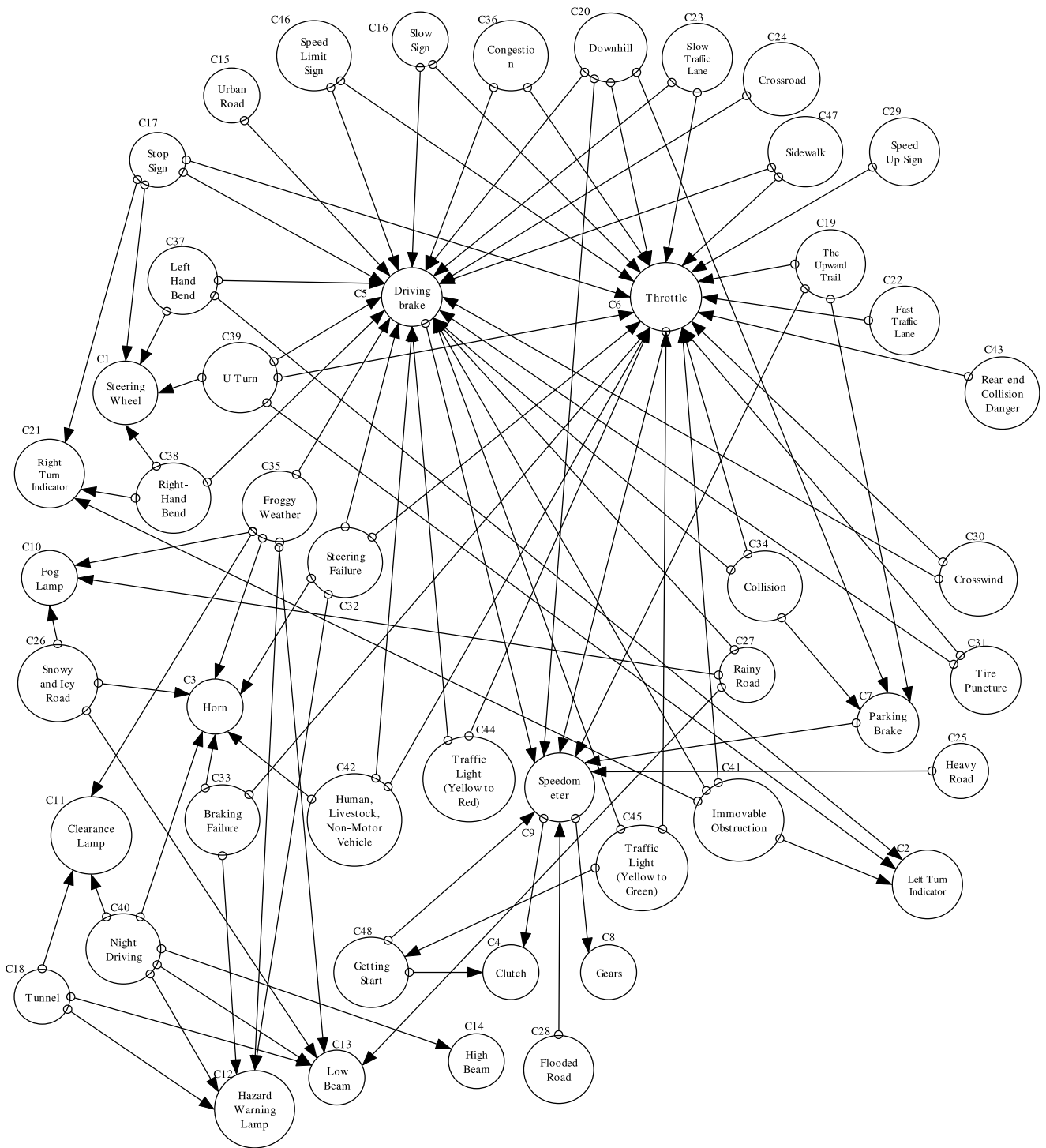


Fig. 6. The initial teacher submodel for the driving training prototype system. The submodel is an FCM with 48 concepts. It stores in a 48×48 matrix.

Step 4. Build the initial FCM.

According to Fig. 6, the initial FCM can be obtained and the part adjacency matrix of Fig. 6 is shown in Fig. 7. Because there are so many concepts and the adjacency matrix of FCM is too big to be fully shown here, matrixes in Fig. 7 are only parts of the full matrixes.

It is unavoidable that the initial teacher submodel will have some defects resulting from the structure of the FCM or unreasonable weights in the FCM. The defects, related to unreasonable weights, can be overcome by the self-learning of the teacher submodel.

4.2.2 Self-Learning of the Teacher Submodel

As mentioned in Section 3.3, after the initial teacher submodel is gained, the next step is the self-learning process of the teacher submodel to create the optimal FCM by using the local and global learning process. The learning process is based on the local learning algorithm and the global learning algorithm. In addition, based on the local and global learning, the teacher submodel can learn continuously according to real situations.

Fig. 8 shows the global learning process of the teacher submodel. For this case, it needs five learning steps to

	C5	C6	C7	C8	C9
C15	$p(0.2 c9 > 0.2, c33 = 0, c34 = 0, c5 = 0)$	0	0	0	0
C16	$p(0.2 C9 > 0.2, c33 = 0, c34 = 0, c5 = 0)$	$p(0 c6 > 0)$	0	0	0
C17	$p(0.8 c9 > 0.2, c33 = 0, c34 = 0, c5 = 0)$	$p(0 c6 > 0)$	0	0	0
C18	0	0	0	0	0
C19	0	$p(0.2 c6 = 0, c9 < 0.2, c16 = 0, c17 = 0, c20 = 0, c31 = 0, c30 = 0, c32 = 0, c33 = 0, c34 = 0, c36 = 0, c39 = 0, c41 = 0, c42 = 0, c43 = 0, c44 = 0, c46 = 0, c47 = 0)$	$p(1 c9 = 0)$	0	-0.1
C20	$p(0.5 c9 > 0.2, c33 = 0, c34 = 0, c5 = 0)$	$p(0 c6 > 0)$	$p(1 c9 = 0)$	0	0.1
C21	0	0	0	0	0
C22	0	$p(0.5 c6 = 0, c9 < 0.4, c16 = 0, c17 = 0, c20 = 0, c31 = 0, c30 = 0, c32 = 0, c33 = 0, c34 = 0, c36 = 0, c39 = 0, c41 = 0, c42 = 0, c43 = 0, c44 = 0, c46 = 0, c47 = 0)$	0	0	0
C23	$p(0.2 c9 > 0.4, c33 = 0, c34 = 0, c5 = 0)$	$p(0.2 c6 = 0, c9 < 0.3, c16 = 0, c17 = 0, c20 = 0, c31 = 0, c30 = 0, c32 = 0, c33 = 0, c34 = 0, c36 = 0, c39 = 0, c41 = 0, c42 = 0, c43 = 0, c44 = 0, c46 = 0, c47 = 0)$	0	0	0
C24	$p(0.2 c9 > 0.1, c33 = 0, c34 = 0, c5 = 0)$	0	0	0	0
C25	0	0	0	0	-0.1
C26	0	0	0	0	0
C27	$p(0.2 c9 > 0.2, c33 = 0, c34 = 0, c5 = 0)$	0	0	0	0

Fig. 7. The adjacency matrix for the initial teacher submodel in Fig. 6.

reach its final state. Fig. 8a is the result of the first learning step, Fig. 8b is the result of the second step, and Fig. 8c is the result of last step.

As a result, the improved FCM of the teacher submodel contains driving knowledge which consists of not only prior knowledge of experts but also the modified and the new knowledge achieved by self-learning. So, the teacher submodel is intelligent enough to reason the answer and evaluate the students' actions, that is, it can guide students' study process.

4.3 Game Process

Using this system, students can study automobile driving skills and traffic laws. The system constructs a learner submodel for each student, which has the same structure as the teacher submodel. Before the study process, the weights between concepts in the learner submodel are all initialized to zero. In the game process, the weights are adjusted continuously by interaction between the student and system. The specific game process is as follows:

- Step 1. Generate the game scene, that is, generate street map, traffic light, passerby, road-rock, and disturbance autos.
- Step 2. Generate the right game guidance by the teacher submodel.
- Step 3. Get the operating sequence of the student.
- Step 4. Judge the veracity of the student's operation according to the game-based learning algorithm.
- Step 5. If the error is a bit big, the right guidance to the student is presented.
- Step 6. Update the FCM of the student.

5 EVALUATION

Two aspects of experiments should be carried out. The first aspect is to evaluate the self-learning ability of the model,

$$\begin{bmatrix} C1 & 0 & 0 & 0 & 0 & \dots \\ C2 & 0 & 0 & 0 & 0 & \dots \\ C3 & 0 & 1 & 0 & 0 & \dots \\ C4 & 0 & 0 & 0 & 1 & \dots \\ C5 & 0 & 0.8 & 0 & 0 & \dots \\ C6 & 0 & 0 & 0 & 0 & \dots \\ C7 & 0 & 0 & 0 & 0 & \dots \\ C8 & 0.8 & 0.8 & 0.8 & 0.2 & \dots \\ C9 & 0.2 & 1 & 0 & 0 & \dots \\ C10 & 0 & 0 & 0 & 0 & \dots \\ C11 & 0 & 0 & 0 & 0 & \dots \\ C12 & 0 & 0 & 0 & 0 & \dots \\ C13 & 0 & 0 & 1 & 1 & \dots \\ C14 & 1 & 1 & 0 & 0 & \dots \\ C15 & 1 & 1 & 1 & 1 & \dots \\ C16 & 0 & 0 & 0 & 0 & \dots \end{bmatrix}$$

(a)

$$\begin{bmatrix} C1 & 0 & 1 & 1 & 0 & 0 & \dots \\ C2 & 0 & 1 & 1 & 0 & 0 & \dots \\ C3 & 0 & 0 & 0 & 0 & 0 & \dots \\ C4 & 1 & 0 & 0 & 1 & 0 & \dots \\ C5 & 0 & 0 & 0 & 0 & 0 & \dots \\ C6 & 0 & 0 & 0 & 0 & 0 & \dots \\ C7 & 0 & 0 & 0 & 0 & 0 & \dots \\ C8 & 0.2 & 0.2 & 0.2 & 0.2 & 0.4 & \dots \\ C9 & 0 & 0 & 0.1 & 0.1 & 0.1 & \dots \\ C10 & 0 & 0 & 0 & 0 & 0 & \dots \\ C11 & 0 & 0 & 0 & 0 & 0 & \dots \\ C12 & 0 & 0 & 0 & 0 & 0 & \dots \\ C13 & 1 & 1 & 1 & 1 & 1 & \dots \\ C14 & 0 & 0 & 0 & 0 & 0 & \dots \\ C15 & 1 & 1 & 1 & 1 & 1 & \dots \\ C16 & 0 & 0 & 0 & 0 & 0 & \dots \end{bmatrix}$$

(b)

$$\begin{bmatrix} C1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C4 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ C5 & 0 & 0.2 & 0 & 0.2 & 0 & 0.2 & 0 & 0.2 & 0 & 0 \\ C6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C8 & 1 & 1 & 1 & 1 & 1 & 0.8 & 0.8 & 0.2 & 0.2 & 0.2 \\ C9 & 0.35 & 0.35 & 0.25 & 0.25 & 0.15 & 0.15 & 0.05 & 0.05 & 0 & 0 \\ C10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(c)

Fig. 8. Global learning of the initial FCM. For this case, it needs five steps to finish the learning process. (a) Step 1 of global learning. (b) Step 2 of global learning. (c) Step 5 of global learning.

that is, the self-learning ability of the teacher submodel. Experiments of this aspect will be discussed in Section 5.1. The second aspect is to prove that the model has a good ability to guide the study process of the student. The experiments of this aspect will be discussed in Section 5.2. All the experiments are carried out in our driving training prototype system.

5.1 Experiments for Self-Learning

As Section 4.2.1 stated, the initial FCM of the teacher submodel is given by the driving experts. It may have conflicts or false knowledge in it. Since the improved FCM has the ability of self-learning, it can remove the false knowledge hidden in the initial FCM automatically. The

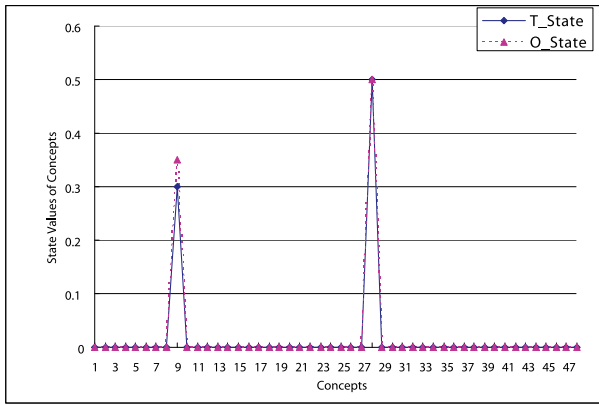


Fig. 9. The expectation output and actual output of concepts before local learning.

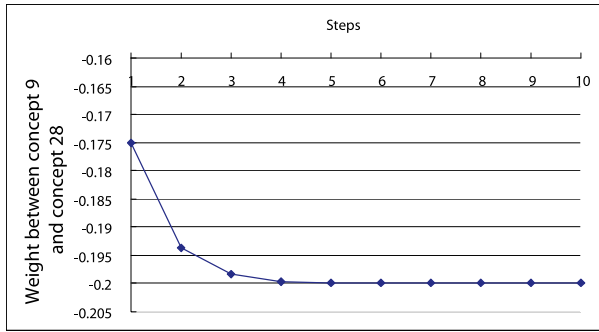


Fig. 10. The weight between C_9 and C_{28} varies with the local learning process.

self-learning process is composed of two steps, i.e., the local learning and the global learning, to get the local optimal FCM and global optimal FCM.

5.1.1 Experiments for Local Learning of Teacher Submodel

The aim of this experiment is to guide the local learning process between concepts in the FCM. Figs. 9, 10, and 11 show the local learning process of relation between C_9 and C_{28} , taking no account of the influence of other concepts.

Fig. 9 shows that the expectation output and actual output of C_9 before the local learning are almost equal, but those of C_{28} are quite different, which means that the relation between C_9 and C_{28} needs to be adjusted by local learning.

Fig. 10 shows the local learning process. In Fig. 10, the weight of connection between C_9 and C_{28} is changing along with the increase of the training process.

As a result, after the local learning, the relation between C_9 and C_{28} is right, which can be reflected by the fact that the actual output of C_9 and C_{28} is almost equal to the expected output, as shown in Fig. 11.

5.1.2 Experiments for Global Learning of the Teacher Submodel

In the global learning experiment, the automobile driving FCM obtained in the last experiment is used as the initial FCM. First, the initial values of concepts are set, such as setting the *initial speed of auto*, *other autos place*, the *crossing*, and so on. Second, assign the expectation values E_{C_i} according to the real ones and expert knowledge. Before

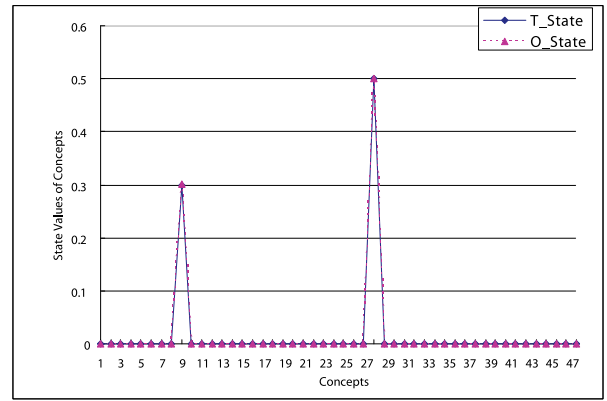


Fig. 11. The expectation output and actual output of concepts after local learning.

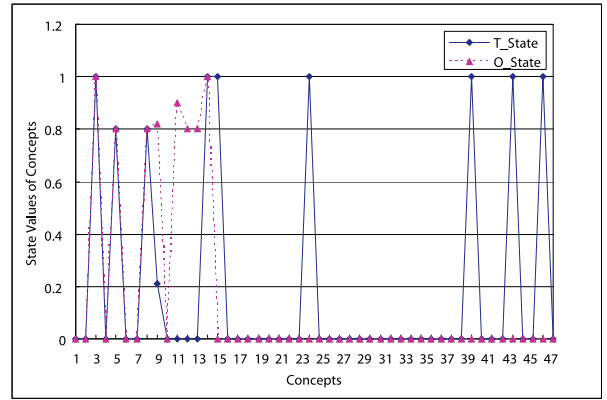


Fig. 12. The actual output of concepts before global learning.

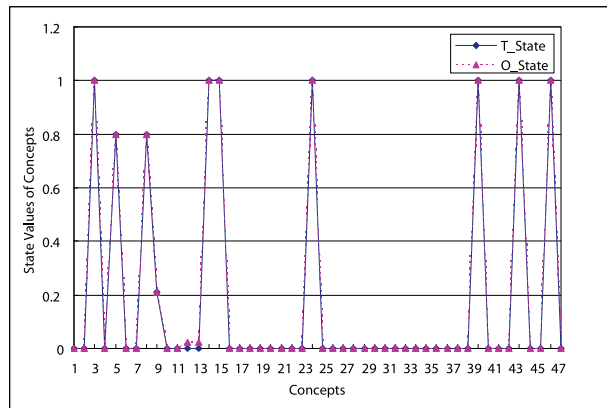


Fig. 13. The actual output of concepts after five steps of global learning.

the training process, the actual output V_{C_i} is different from E_{C_i} , and V_{C_i} is much closer to E_{C_i} after the global learning.

In Figs. 12, 13, and 14, T_state is the expected output of the concepts and O_State is the actual output of the concepts. The X-axis represents the 48 concepts (as shown in Fig. 6) and the Y-axis represents the output of state values.

Fig. 12 is the actual output of concepts before global learning. Fig. 13 is the actual output of concepts after learning five times. Fig. 14 is the actual output of concepts when global learning is completed. More specifically, Fig. 12 shows that there are many differences between the expected output and actual output of some concepts before global learning. The initial FCM is imperfect and is far from reality, justifying for the need for learning.

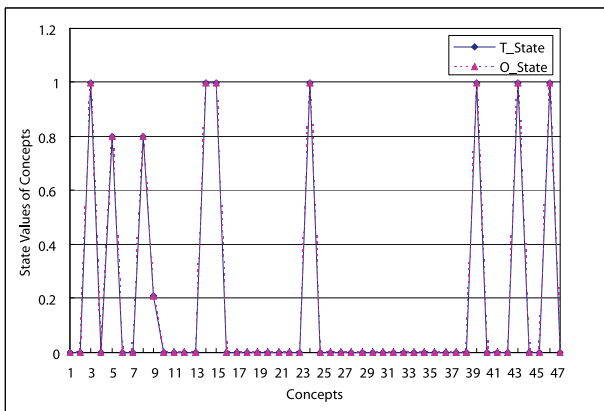


Fig. 14. The actual output of concepts when global learning is completed.

After learning for five times, the actual output is much closer to the expected output except for C_{12} and C_{13} (Fig. 13), which shows that our global learning process is quite effective. After the global learning process is complete, the actual output is almost consistent with the expected output.

5.2 Experiments for the Model's Learning Guidance Ability

To test the learning guidance ability of our game-based learning model using the FCM, we adopted contrast experiments between two groups of students on the driving training prototype system. After experiments, questionnaires filled by each student were analyzed to find out the usefulness of the system and the model [22].

5.2.1 Participants of the Experiments

The participants of the experiment were 20 persons who were studying driving skills or were going to study driving skills. They were randomly divided into the control group *A* and the experimental group *B* with 10 students in each group. Group *A* learned the concepts from the driving handbook directly, while group *B* was trained in the simulation system when they were studying from the driving handbook.

5.2.2 Experimental Design

There is much information to study in the driving study process. From the information, we selected some complex and related topics to construct six study cases: two simple ones, two mediocre ones, and two difficult ones. The students studied these cases one by one. The study process consisted of three rounds of study. The first and the second rounds were time-limited study, while the third round was a time-unlimited study. After each round of study, there was a test to evaluate the study effectiveness. During the study process, we recorded how long each student spent on each case. The study process did not end until the students got above 80 percent of the correct answers.

This experiment focused on whether the simulation system and the model can improve the efficiency of the students' study. There were only two groups, *A* and *B*, in experiments. Group *A* was used as the control group and group *B* was used as the experimental group. We made use of the t-test for computing the result of the experiment, which is shown in Table 3.

After the study experiments, we took advantage of a questionnaire to collect subjective evaluations of the system and the model from the students. Only group *B* studied with the help of the simulation driving system and was asked to fill in the questionnaire. The questions in the questionnaire were designed simply to see whether our design and implementation of the simulation system was effective or not in helping them study.

5.2.3 Experimental Procedures

In order to ensure the correctness of the evaluation of the driving training prototype system, the following steps were carried out [23]:

Step 1. Introduction.

The first step was to make students know the objective of the experiment and to make students familiar with the basic instructions about the driving training prototype system. To do this, we let the students of group *B* study an irrespective concept to the following test using the simulation system.

Step 2. Pretest.

The second step was that all the students of the two groups attended an individual test. The answers of each student were analyzed by statistical methods and the statistical results were compared with the post-test.

Step 3. First-round time-limited study and test.

In this step, group *A* and group *B* employed different methods of the leaning process. The students of group *A* did not use the simulation driving system and they only studied from the driving handbook. The students of group *B* studied from the handbook and practiced in the driving training prototype system. That is to say, group *B* used game-based learning, but group *A* did not. The time for each case was limited to 5 minutes, and after this round of study, the test was repeated.

Step 4. Second-round time-limited study and test.

This step repeated step 3 and the test result was recorded separately.

Step 5. Time-unlimited study and test.

To get the same study effectiveness in principle, different students need different times. In addition to the individual difference, the main reason might lie in different study methods. To compare the two groups' differences, this round of study was an unlimited time study. The student could not stop studying until he got above 80 percent of the correct answers in the test. During the process, the time each student spent was recorded for the final analysis.

Step 6. Survey.

The last step was that students fulfill an individual subjective questionnaire which collected their opinions about the experiments. The questionnaire is shown in Table 1.

5.2.4 Design of the Questionnaire

The questionnaire included 10 questions: questions 1 and 2 were about the knowledge of the participant, questions 3 and 5 were about using the system, and the other questions were about the learning effect of the system. By analyzing the scores students gave, the subjective estimate about the system can be achieved and the analysis is included in the following sections.

TABLE 1
Summary of Results from the Student Responses to a Series of Questions

No	Questions	Mean value	Standard deviation
1	Have you ever used driving training game before?	4.6	0.52
2	Are you familiar with the operation of driving training game?	4.6	0.52
3	Do you know that there is a teacher in the driving training system?	4.6	0.52
4	Does the teacher in the system help you thinking process for your study?	4.4	0.52
5	Does the teacher in the system interfere with your study?	1.7	0.48
6	Do you think that the teacher in the system help your study?	4.7	0.48
7	Does the game make your study more interesting?	4.3	0.62
8	Does the teacher help you avoid some mistakes or misconceptions?	4.6	0.52
9	Do you agree that system can improve the effectiveness of your study?	4.6	0.52
10	Do you agree that the system can improve the efficiency of your study?	4.6	0.52

5.2.5 Experimental Results

There are two aspects of the experimental results: subjective survey and objective evaluation, which are discussed in the following:

1. Results of subjective survey

The questionnaire used a five-point Likert Scale, with each question having five selections marked from one point to five points. One point means the least favorable to the statement. Five points means the most favorable to the statement. Each student filled in the questionnaire individually and gave each question a score. The summary of the survey is shown in Table 1. Through analyzing the results of the questionnaire, we know that the students' opinions about the system are very positive in general. In Table 1, except for the fifth question, most of the questions' mean value is above 4.5, which means the students agreed that the system was very helpful to improve their driving skills. Most of them thought that the system made studying interesting and that the teacher in the system guided them to study effectively. The fifth question surveyed if "the teacher" interfered the student, which is a negative question. In other words, the low score means a better result.

Furthermore, the standard deviation of each question is small, which means that all the students had a consensus opinion about this question.

2. Results of objective evaluation

The guidance ability of the model can be evaluated by the differences between the two groups' real study effectiveness. In the experiment, we consider the study effectiveness from two aspects. One is the test score and the other is the study time.

Table 2 summarizes the average results of the four tests. The pretest results show that two groups had similar

TABLE 2
Average the Test

Group	Pre-test	1 round test	2 round test	Post-test
Percentage of Correct Answers for group B(gamed)	20.5%	31.5%	39.0%	85.0%
Percentage of Correct Answers for group A(not gamed)	19.0%	25.0%	52.0%	86.0%

knowledge levels on driving skills. After the first round of study, the test results show that group A was a little better than group B. The reason is that group B spent some time familiarizing themselves with the game system. After the second round of study, group B was better than group A, obviously because of the help of the driving training prototype system. It should be noted that the result of the post-test does not mean anything because the two groups spent different amounts of time to achieve the goal.

Both gamed and nongamed students had significant improvement in the second round test compared with the pretest. However, it is very evident that the improvement of gamed students is greater than that of nongamed students. So, the driving training prototype system is effective in improving the study process of students effectively.

The test was designed so that each student must answer the questions correctly above 80 percent by continuous use of the study process. During the test, the time each student spent on each case was recorded. The time for each case consists of three parts: the first round and the second round, which are 5 minutes each, and the third round, which is different for different students. We measured how long the students spent studying to finish the test. By using an Independent Samples t-test Procedure, where $P < 0.05$ is considered to be significant, the *average*, *standard deviation*, and *P-value* are computed and listed in Table 3.

In Table 3, with the difficulty level increasing, the difference between the two groups extends more and more. For the simple cases, $P > 0.05$ means that the difference is small. For the mediocre cases, the average time group B spent is less than that of group A and $P < 0.05$ means that the difference is significant. For the difficult cases, group B did much better than group A and the P is much smaller than 0.05, which means that the difference is very significant.

TABLE 3
The Results of the T-Test

Diffi- culty level of the cases	Group	Number of stu- dents in the group	Average time of the students study a case (minute)	Stan- dard devia- tion	P-value
Simple	A	10	15.00	4.22	0.3446
	B	10	15.95	5.35	
Mediocre	A	10	19.20	6.18	0.0054
	B	10	15.90	4.71	
Difficult	A	10	27.00	6.0	0.00002
	B	10	18.60	7.6	

In general, Tables 2 and 3 show that the driving training system is helpful in improving the study process effectiveness.

5.2.6 Discussion of External Factors

Although the experimental results are satisfactory, some external factors, which may have influenced the experimental results, should be further discussed. Here, we only discuss the following three factors:

1. Difference of participants

It is inevitable that the knowledge of participants varies from person to person. For example, some participants have acquired some driving knowledge, so their experimental results are influenced by their prior knowledge.

2. Motivation of participants

During the experiment, each participant has different motivations. For example, some participants may be absent-minded in the experiments, which will influence the results.

3. Group of participants

If participants are not divided into groups randomly and different participants are not in a fairly equal manner, the groups will influence the results to a great extent.

The difference of participants is an objective factor, while the motivation of participants is a subjective factor. The influence of both is difficult to eliminate, so we tried to minimize the influence in our experiments.

Our method was to divide the participants into groups as randomly as possible. In the next step, we will increase the number of participants and groups, which may get more convincing results.

5.2.7 Comparison with Traditional Driving Simulation

Based on the proposed model, the prototype system is different from traditional driving simulations in the two main aspects as follows:

1. Efficiency

Since the proposed model is based on the improved FCM which has the abilities of reasoning and self-learning, it can automatically generate the scenario and deduce the answer to the generated scenario by the FCM's reasoning. But, for most of the traditional driving simulations, they need the programmer to design the scenario by hand. They also need to define the conditions for the scenario to judge the operator's actions. So, it is more efficient to develop the system based on our model than on the traditional methods.

2. Precision

It is for the same reason that our system may have lower precision than traditional systems. Because our model generates scenarios and reasons the answers automatically, it is certain that its precision is lower than those by hand. It requires a further study in order to improve the precision in sophisticated studying environments.

In conclusion, compared with traditional simulations, our model is highly automatic in generating study scenarios and reasoning the answers. But the procession of the system is lower than traditional simulations and needs to be improved.

6 CONCLUSIONS AND FUTURE WORK

This paper utilizes the *Hebbian Learning Rule* to improve the FCM to acquire new knowledge from data itself, and uses

the *Unbalance Degree* to improve the FCM to correct the false prior knowledge automatically.

A new game-based learning model is proposed based on the improved FCM. Compared with traditional game-based models, the proposed model can generate study scenarios and reason answers automatically, which can increase the efficiency of game-based learning system design and provide good guidance for students in their study process.

In order to show how to design a game-based learning system with the proposed model, a driving training prototype system was developed as a case study. Under the system's guidance, students studied driving knowledge more effectively, which is demonstrated by contrast experiments. The experimental results also show that the proposed model is available and effective.

The main contributions of this paper include:

1. Taking advantage of the *Hebbian learning rule* and *unbalance degree* to extend the FCM in order to equip it with the abilities of self-learning and knowledge acquisition from both data and prior knowledge, which makes the FCM more suitable for designing a game-based learning system.
2. Proposing a new guided game-based learning model based on the improved FCM including the teacher submodel, the learner submodel, and a set of learning mechanisms. The model provides a workable method to help design a game-based learning system.
3. Giving a case study of the proposed model in which a driving training prototype system was implemented according to the proposed model. Experimental results show that the proposed model is effective and valid in terms of controlling and guiding students' study process.

We will improve the submodels in our future work. Taking the teacher submodel as an example, decreasing its dependence on expert knowledge and increasing its adaptive ability to the changing environments are worth further investigation. These issues can lead to fruitful research outcomes in the future.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers and Professor Qing Li from Hong Kong City University for their constructive comments. Research work reported in this paper was partly supported by the key basic research program of Shanghai under grant no. 09JC1406200, by the National Science Foundation of China under grant nos. 91024012, 61071110, and 90612010, and by the Shanghai Leading Academic Discipline Project under grant no. J50103.

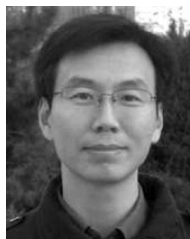
REFERENCES

- [1] J. Wolfe and D. Crookall, "Developing a Scientific Knowledge of Simulation/Gaming," *Simulation and Gaming*, vol. 29, no. 1, pp. 7-19, 1998.
- [2] M. Prensky, *Digital Game-Based Learning*. McGraw-Hill Trade, 2000.
- [3] S. Wu et al., "Game-Based Learning Scenes Design for Individual User in the Ubiquitous Learning Environment," *Proc. Third Int'l Conf. Technologies for E-Learning and Digital Entertainment (Edutainment '08)*, pp. 451-462, 2008.

- [4] T. Wang et al., "Developing a Prototype of Video Game-Based Learning Environment," *Proc. First IEEE Int'l Conf. Ubi-Media Computing and Workshops (U-Media '08)*, pp. 586-591, 2008.
- [5] P. Kearney, "Cognitive Assessment of Game-Based Learning: Colloquium," *British J. Educational Technology*, vol. 38, no. 3, pp. 529-531, 2007.
- [6] S. Tobias, "Interest, Prior Knowledge, and Learning," *Rev. of Educational Research*, vol. 64, no. 1, pp. 37-54, 1994.
- [7] J. Brophy, "Teacher Influences on Student Achievement," *Am. Psychologist*, vol. 41, no. 10, pp. 1069-1077, 1986.
- [8] Z.Q. Liu and R. Satur, "Contextual Fuzzy Cognitive Maps for Decision Support in Geographic Information Systems," *IEEE Trans. Fuzzy Systems*, vol. 7, no. 10, pp. 495-502, Oct. 1999.
- [9] J.B. Noha and K.C. Lee, "A Case-Based Reasoning Approach to Cognitive Maps-Driven Tacit Knowledge Management," *Expert Systems with Applications*, vol. 19, no. 4, pp. 249-259, 2000.
- [10] B. Kosko, *Fuzzy Engineering*. Prentice-Hall, 1997.
- [11] P. Elpiniki and P. Peter, "Two-Stage Learning Algorithm for Fuzzy Cognitive Maps," *Proc. Second IEEE Int'l Conf. Intelligent System*, pp. 82-87, 2004.
- [12] S. Wojciech et al., "Parallel Learning of Large Fuzzy Cognitive Maps," *Proc. Int'l Joint Conf. Neural Networks*, Aug. 2007.
- [13] N. Mateou et al., "A New Traversing and Execution Algorithm for Multilayered Fuzzy Cognitive Maps," *Proc. IEEE Int'l Conf. Fuzzy Systems*, pp. 2216-2223, 2008.
- [14] B. Ng, A. Si, R. Lau, and F. Li, "A Multi-Server Architecture for Distributed Virtual Walkthrough," *Proc. ACM Symp. Virtual Reality Software and Technology (VRST '02)*, pp. 163-170, 2002.
- [15] J. Chim et al., "On Caching and Prefetching of Virtual Objects in Distributed Virtual Environments," *Proc. ACM Multimedia*, pp. 171-180, 1998.
- [16] D.O. Hebb, *The Organization of Behavior: A Neurophysiological Theory*. Wiley, 1949.
- [17] A. Klopf, "Drive-Reinforcement Model of Single Neuron Function: An Alternative to the Hebbian Neuronal Model," *AIP Conf. Proc. 151: Neural Networks for Computing*, pp. 265-270, 1986.
- [18] A. Klopf, "Drive-Reinforcement Learning: A Real-Time Learning Mechanism for Unsupervised Learning," *Proc. IEEE First Int'l Conf. Neural Networks*, pp. 441-446, 1987.
- [19] I. Aleksander and H. Morton, *An Introduction to Neural Computing*. Chapman and Hall, 1990.
- [20] R. Linsker, "How to Generate Ordered Maps by Maximizing the Mutual Information between Input and Output Signals," *Neural Computation*, vol. 1, no. 3, pp. 402-411, 1989.
- [21] M.T. Hagan and H.B. Demuth, *Neural Network Design*, pp. 7-1-7-16. PWS, 1996.
- [22] Y.-M. Cheng, L.-S. Chen, and H.-C. Huang, "Building a General Purpose Pedagogical Agent in a Web-Based Multimedia Clinical Simulation System for Medical Education," *IEEE Trans. Learning Technologies*, vol. 2, no. 3, pp. 216-225, July-Sept. 2009.
- [23] R.M.C Garcia, C.D. Kloos, and M.C. Gil, "Game Based Spelling Learning," *Proc. 38th ASEE/IEEE Frontiers in Education Conf.*, pp. S3b11-S3b15, 2008.



Xiangfeng Luo received the master's and PhD degrees from the Hefei University of Technology in 2000 and 2003, respectively. He was a postdoctoral researcher with the China Knowledge Grid Research Group, Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), from 2003 to 2005. He is currently an associate professor in the School of Computers, Shanghai University. His main research interests include web content analysis, Semantic Networks, web knowledge flow, Semantic Grid, and Knowledge Grid. His publications have appeared in *Concurrency and Computation: Practice and Experience*, the *Journal of Systems and Software*, and the *Journal of Computer Science and Technology*.



Xiao Wei is currently working toward the PhD degree at Shanghai University. His main research interests include game-based learning, interactive computing, and web content analysis.



Jun Zhang received the bachelor's degree in 2008 from Shanghai University, where he is currently working toward the graduate degree in the School of Computers. His main research interests include online word relation discovery and topic detection and tracking.