Modeling Intelligent E-Learning Systems based on Adaptive Fuzzy Higher Order Petri Nets

Mohamed Amin
Minufiya University
Faculty of science, Dept. of Math.& Computer Science

Nabawia El-Ramly
Minufiya University
Faculty of science, Dept. of Math.& Computer Science

Doaa Shebl
Minufiya University
Faculty of science, Dept. of Math.& Computer Science

ABSTRACT

In this paper, we introduce adaptive and intelligent technologies in E-learning course development using Modular Object Oriented Dynamic Learning Environment (Moodle) based on Petri nets as a modeling formalism. Since classical Petri nets and fuzzy Petri nets are not adaptable according to the changes of the new incoming data such as the parameters of Moodle (static course material, interactive course material, activities), we introduce adaptive fuzzy higher order Petri net (AFHOPN) that is dynamically adjust the parameters. AFHOPN helps to describe and analyze the dynamic behavior, production inference of the intelligent E-learning systems and measure the learning rate.

Keywords

Higher order Petri nets, Fuzzy reasoning, E-learning systems.

1. INTRODUCTION

As the Internet and World Wide Web are rapidly developed, the technologies that support the educational processes come closer to the traditional educational system. Traditional learning portfolio assessment relies on man-made data collection and a writing-centered learning process. Difficulties in data storage, search, and management after long-term implementation have become problematic in developing and implementing portfolio assessment. A recent research demonstrated that both instructors and learners have very positive perceptions towards using e-learning as a teaching tool [1]. According to [2] adaptive web-based educational systems provide an alternative to the traditional " just-put-it-on-the-web" in the development of web-based educational courseware. A web-based learning portfolio can be collected, stored, and managed automatically by computers when learners interact with an e-learning platform. Therefore, learning performance assessment using a Web-based learning portfolio has received significant attention recently [3]. In [4], web-based educational systems attempt to be more adaptive by building a model of the goals, preferences and knowledge of each individual student and using this model throughout the interaction with the system.

There exist a wide variety of divers adaptive and intelligent web-based educational system. The 'rules' that are used to describe the creation of such systems are not yet fully standardized, and the criteria that need to be used pedagogically effective rules-sets (i.e adaptation parameters). Many experimental educational hypermedia systems have been created, each to their own unique specifications [5]. As yet, however, no combined effort has been made to extract the common design paradigm from these systems. In [6] dynamic fuzzy Petri nets (DFPN's) model is introduced to increase the flexibility of the tutoring a gent behavior (software assistant for a single user, who may be an expert in e-learning course). A compete standardized course generation using DFPN's usually needs some auxiliary materials which are determined by the students' behavior are introduced in [7]. This makes the course design process more complex and more challenges.

Since, fuzzy Petri nets (FPN's) and fuzzy higher order Petri nets (FHOPN's) are not intended to act as a learning tool, but rather they serve to process uncertain data and provide results between true and false that may be further interpreted by external elements of a system. In this paper, we introduce an adaptive fuzzy higher order Petri nets (AFHOPN's) that has the capability of dynamic adjustment of parameters and allows a structural representation of knowledge and has got systematic procedure for supporting the fuzzy reasoning. After that, we introduce an intelligent e-learning systems using Modular Object-Oriented Dynamic Learning Environment (Moodle) for creating dynamic online learning communities and generating automatic course development. This approach helps the teachers to perform precise formative assessments according to e-learning tools and to devote themselves for teaching, since they save significant time in performing learning evaluation.

The rest of the paper is structured as follows. section 2 introduces the formal description of the AFHOPN proposed model and the fuzzy knowledge representation. Section 3 proposes an intelligent E-learning systems. Section 4 gives Modular Object Oriented Dynamic Learning Environment for developing E-learning course based on AFHOPN proposed model. Finally, in section 5, we draw the conclusion.

2. AFHOPN PROPOSED MODEL

Petri nets (PN's) have become a widely accepted formalism for modeling and analyzing asynchronous system with concurrent and parallel activities in variety of application domains[8], [9].

PN's can be combined with different techniques and theories such as fuzzy sets [10], neural networks (NN's) [11], [12] and object-oriented programming. These modified PN's are widely used in manufacturing systems [13], robots , knowledge based systems [14], and process control. In order to deal with fuzzy in formation, Fuzzy Petri Net (FPN's) introduced in [10].

2.1 The Formal Definition of AFHOPN

Definition 1. An Adaptive Fuzzy Higher-Order Petri Net can be defined as a 11 tuple : AFHOPN = (P, T, D , I, O, W_I, W_O, cf, th, α, β), where P ={ $P_1, P_2,\dots, P_n$} is a finite set of places, | P | = n > 0; T ={$t_1, t_2,\dots, t_m$} is a finite set of transitions, |T | = m >

0 ; D ={d_1, d_2,…, d_n} is a finite set of propositions, |D |= |P |, P ∩ T ∩ D=ϕ ; I:T→$\bigcup_{i=1}^{n} P^i$ is the input function representing a mapping from transitions to some or all elements in bags of their input places, and $\bigcup_{i=1}^{n} P^i = P^1 \cup P^2 \cup \ldots \cup P^n$; O:T→P is the output function representing a mapping from transitions to bags of their output places; cf:T→[0,1] is an association function represents the certainty factor of the transitions; th:T→[0,1] is an association function represents the threshold value of the transitions; α:P→ [0,1] is an association function represents the token truth value in place pi and denoted by $\alpha(p_i) \in [0,1]$, αi represents the fuzzy belief of place p_i i.e. $\alpha i = \alpha(p_i)$; β:P→D is an association function represents a bijective mapping from places to proposition, If $\alpha(p_i) = y_i$, $y_i \in [0,1]$ and $\beta(p_i) = d_i$, then the degree of truth of proposition d_i is y_i. Function W_I:I→N is an input weight function, which assigns to each k^{th}-order arc, k-dimensional vector whose components represent the occurrence of places in this arc's hold set, where N represents the set of vectors of real values (between zero and one) (i.e N∈ [0,1]), W_O:O → V is an output weight function which assigns to each output arc, where V∈ [0,1].

We denote $I^{(j)}_{\{r(i)\}k}$ as the k^{th}-order input arc of transition t_j from its hold set $H^{(j)}_{\{r(i)\}k}$ = { $p_{r(1)}$, $p_{r(2)}$, …, $p_{r(k)}$}, i.e. $H^{(j)}_{\{r(i)\}k} \subseteq I(t_j)$, where k = 1, 2, …, n ; j = 1, 2, … , m ; n = | P |; m = | T |, the corresponding weight vector of input arc $I^{(j)}_{\{r(i)\}k}$ is $W^{(j)}_{\{r(i)\}k}$ = [$N_{r(1)}$, $N_{r(2)}$, …, $N_{r(k)}$], where $N_{r(i)}$ is the occurrence of place $P_{r(i)}$ in $H^{(j)}_{\{r(i)\}k}$, and the output weight function W_O from the transition tj to the output place P_l represented by $V_l(t_j)$ or v_{jl}. The transition with k input arcs has k thresholds. The threshold is assigned to transition for each input arc represents a lower bound on the degree of propositions that the hold set of the arc should possess in order to fire the transition. Accordingly, we use $th_{j\leftarrow\{r(i)\}k}$ to denote the threshold of transition $t_{j\leftarrow\{r(i)\}k}$ when it fired by the input arc $I^{(j)}_{\{r(i)\}k}$.

2.2 Implementation of AFHOPN Model

Here, we show that each type of fuzzy production rule (FPR) can be modeled by AFHOPN with the aggregation function δ. In order to give the definition of enabled and firing of the transition, we introduce the extended fuzzy load of the arc $I^{(j)}_{\{r(i)\}k}$.

Definition 2 (extended fuzzy load γ). Let the transition t_k has the n^{th}-order input arc $I^{(k)}_{\{r(i)\}n}$ with hold set $H^{(k)}_{\{r(i)\}n} = \{ p_{r(1)}, p_{r(2)}, \ldots, p_{r(n)} \}$, the extended fuzzy load $\gamma^{(k)}_{\{r(i)\}n}$ is computed as the value of fuzzy belief of place $P_r(i)$ in its hold set multiplied by its corresponding weight vector($W^{(k)}_{\{r(i)\}n}$) of $I^{(j)}_{\{r(i)\}k}$, where $W^{(k)}_{\{r(i)\}n}$ = [$N_{r(1)}$, $N_{r(2)}$, …, $N_{r(n)}$], i.e $\gamma^{(k)}_{\{r(i)\}n}$ =[$\alpha_{r(1)}$, $\alpha_{r(2)}$, …,$\alpha_{r(n)}$] ∘ [$N_{r(1)}$, $N_{r(2)}$, …, $N_{r(n)}$] =[$\alpha_{r(1)}$. $N_{r(1)}$, $\alpha_{r(2)}$. $N_{r(2)}$,…,$\alpha_{r(n)}$. $N_{r(n)}$] where, $\alpha_r(i)$ is the fuzzy belief of place $P_r(i)$, $N_r(i)$ is the occurrence of $P_r(i)$ ($P_r(i) \in H^{(k)}_{\{r(i)\}n}$) , and ∘ is the array multiplication element by element .

Definition 3 (Enabled). A transition tk is enabled if $\exists\, I^{(k)}_{\{r(i)\}n}$ s.t min($\gamma^{(k)}_{r(1)}$, $\gamma^{(k)}_{r(2)}$,…, $\gamma^{(k)}_{r(n)}$) $\geq th_{k\leftarrow\{r(i)\}n}$.

Definition 4 (Firing rule). An enabled transition $t_{k\leftarrow\{r(i)\}n}$ may or may not fire. If the transition $t_{k\leftarrow\{r(i)\}n}$ is fired, then tokens are removed from the hold set $H^{(k)}_{\{r(i)\}n} = \{ p_{r(1)}, p_{r(2)}, \ldots, p_{r(n)} \}$ and a token is deposited to each of its output places.

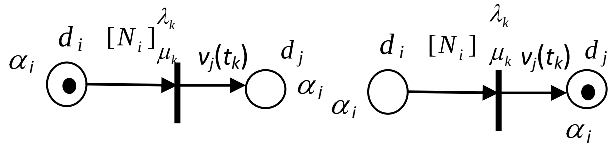
The fuzzy belief of the output place is computed as follows:

$$\alpha(P_j) = v_j(t_k).cf^{'}(t_k) \text{, where } cf^{'}(t_k) = \delta \cdot cf(t_k)$$

the value of the function δ is dependent on the three types of FPR.

TYPE 1: A simple FPR

IF d_i THEN d_j ; with cf = μ_k , $th_{k\leftarrow i} = \lambda_k$, $W_i^{(k)} = N_i$, and W_O = $v_j(t_k)$. The simple FPR can be modeled as shown in Fig.1.(a). The transition tk has only one first-order input arc $I_i^{(k)}$, and its hold set $H_i^{(k)} = \{ p_i \}$, *if* $\gamma_i^{(k)} \geq \lambda_k$, *then* the transition $t_{k\leftarrow i}$ can fire as in Fig.1.(b). Now, the value of δ can be computed as $\delta = \alpha_i \cdot N_i = \gamma_i^{(k)}$.

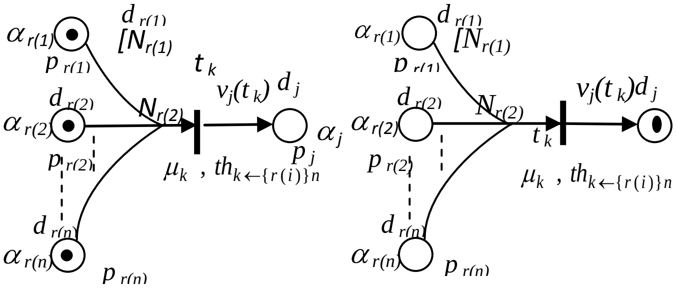


(a) Before firing of $t_{k\leftarrow i}$ (b) After firing of $t_{k\leftarrow i}$

Fig.1. AFHOPN represents TYPE 1 of FPR.

TYPE 2: A Composite Conjunctive Rule

IF $d_{r(1)}$ AND $d_{r(2)}$ AND … AND $d_{r(n)}$ THEN d_j ; with cf = μ_k , $th_{k\leftarrow\{r(i)\}n} = \lambda_k$, $W^{(k)}_{\{r(i)\}n}$ = [$N_{r(1)}$, $N_{r(2)}$, …, $N_{r(n)}$] , and W_O = vj(tk). The consequence d_j can not result until the propositions $d_{r(1)}$, $d_{r(2)}$,…, and $d_{r(n)}$ are presented with suitable truth values. If one proposition $d_{r(i)}$ does not exist (true) in the left side of rule, then the consequence dj not result at the right side of rule, where the transition in AFHOPN fires only by one input arc. All propositions $d_{r(1)}$, $d_{r(2)}$,…, and $d_{r(n)}$ must be linked on one n^{th}-order arc as shown in Fig. 2.(a). In this case, the transition tk has one n^{th}-order arc $I^{(k)}_{\{r(i)\}n}$, and its hold set $H^{(k)}_{\{r(i)\}n} = \{p_{r(1)}, p_{r(2)}, \ldots, p_{r(n)}\}$. If min ($\gamma^{(k)}_{r(1)}$, $\gamma^{(k)}_{r(2)}$, …, $\gamma^{(k)}_{r(n)}$) ≥ λ_k, then the transition $t_{k \leftarrow \{r(i)\}n}$ can fire and tokens removed from the hold set to the output place transition as shown in Fig. 2.(b). Now, the value of δ can be computed as follows : δ = min ($\alpha_{r(1)}$. $N_{r(1)}$, $\alpha_{r(2)}$. $N_{r(2)}$,…,$\alpha_{r(n)}$. $N_{r(n)}$) = min($\gamma^{(k)}_{r(1)}$,$\gamma^{(k)}_{r(2)}$, …,$\gamma^{(k)}_{r(n)}$)



(a) Before firing of $t_{k \leftarrow r(i)}$ (b) After firing of $t_{k \leftarrow r(i)}$

Fig. 2. AFHOPN represents TYPE 2 of FPR.

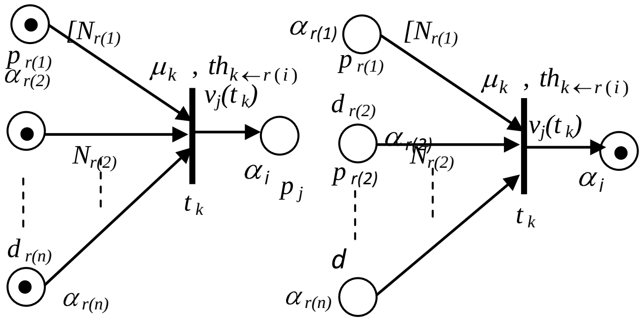
TYPE 3: A Composite Disjunctive Rule

IF $d_{r(1)}$ OR $d_{r(2)}$ OR … OR $d_{r(n)}$ THEN d_j with cf = μ_k and thresholds $th_{k \leftarrow r(1)} = \lambda_1$, $th_{k \leftarrow r(2)} = \lambda_2$, …, $th_{k \leftarrow r(n)} = \lambda_n$, W_I =[$N_{r(1)}$, $N_{r(2)}$, …, $N_{r(n)}$] , and W_O = $v_j(t_k)$, where $th_{k \leftarrow r(i)}, i = 1, 2, \ldots, n$ is associated with transition t_k for the input arc $I^{(k)}_{r(i)}$ that has hold set $H^{(k)}_{r(i)} = \{p_{r(i)}\}$. For this rule, there are two cases:

Case 1. When all premises $d_{r(i)}$ are a simple proposition and the consequence d_j may result by any one of premises, if the proposition $d_{r(i)}$ is presented with suitable fuzzy truth value $\alpha_{r(i)}$, then the proposition d_j can be concluded and its fuzzy truth value α_j can be aggregated, where the rule has n propositions in premises and each proposition can produce the proposition dj in the consequences. In Fig. 3.(a), we note that the transition has n first-order input arcs, and if the transition fired by one arc only the value of δ can be computed as in Type 1 of FPR. On the other hand, if the transition fired by (s ≤n) arcs, then we can compute δ as:

δ = max ($\gamma^{(k)}_{r(1)}$,$\gamma^{(k)}_{r(2)}$,…,$\gamma^{(k)}_{r(s)}$), s =1, 2, …, n.

$\alpha_{r(1)}$



(a) Before firing of $t_{k \leftarrow r(i)}$ (b) After firing of $t_{k \leftarrow r(i)}$

Fig. 3. AFHOPN represents TYPE 3 of FPR.

Case 2. When one or more premises is composite conjunctive premise, as shown as in Fig. 4. , if a transition fired by one of its first-order arc only, the value δ can be computed as in TYPE 1 of FPR. If the transition fired by one of its qth-order arc only, then the value of δ can be computed as in type 2 of FPR. But if the transition fired by s ≤ n arcs (first or qth-order), then the value of δ can be computed as : δ = max ($\gamma^{(k)}_{r(1)}$, …, min ($\gamma^{(k)}_{r(i1)}$, … $\gamma^{(k)}_{r(iq)}$) , …, $\gamma^{(k)}_{r(n)}$).

2.3 Fuzzy Reasoning Algorithm and The Updating Weights Algorithm

Here, we develop the fuzzy reasoning algorithm for computing fuzzy beliefs at places and introduce an algorithm for updating the input and output weights.

Algorithm A: Fuzzy Reasoning Algorithm

Input: the truth values of a set of antecedent propositions.

Output: the truth values of a set of consequences propositions (goal places).

Step 1 Build the set of user input places P_I .

Step 2 Build the set of initially enabled transitions $T_{initial}$.

do

{ Step 3 Find current enabled transitions $T_{current}$(R)

according to definition 5.

Step 4 Calculate new truth values of a set Consequences propositions produced by fired transitions according to definition 6.

Step 5 Make token transmission. Assume P is one of the output places of a fired t:

5.1 If a place has only one input transition; then add a token to P with the truth value which is produced by its input transition.

5.2 If a place has one input transition with n (n>1) input arcs, and it fired by s (s≤n) of its input arcs, then add to P with the maximum truth value produced by the transition w.r.t. all s input arcs.

5.3 If a place has m(m>1) input transitions (of type that illustrated in step 5.1 or 5.2) and k (k≤m) of its input transitions fired, then select the transition, which will give the maximum output and add a token to output place *P* with the truth value produced by this transition.

Step 6 Let $T = T - T_{current}$.}

Step 7 while($T_{current} = \varphi$).

After applying the fuzzy reasoning algorithm, the truth values of a set of antecedence and consequence are changed. So, the parameters (weights) must be adjusted according to the new truth values of the input and the output places. This lead us to introduce an algorithm for updating the input and the output weights.

The Widrow-Holf Learning law (Least Mean Square) can applied as

$$W^{(k)}_{\{r(i)\}n}\ (x+1) = W^{(k)}_{\{r(i)\}n}\ (x)+(\delta\ \gamma^{(k)}_{\{r(i)\}n}\ -\alpha(P_j)) \qquad (1)$$

where $\gamma^{(k)}_{\{r(i)\}n}$ is denotes the extended fuzzy load of transition tk, $W^{(k)}_{\{r(i)\}n}$(x) is corresponding the input weight vector of input arc $I^{(k)}_{\{r(i)\}n}$ in a step x, and $W^{(k)}_{\{r(i)\}n}$ (x+1) is the input weight vector of input arc $I^{(k)}_{\{r(i)\}n}$ in a step (x+1).

Algorithm B: Updating Input and Output Weights

Input: the truth values of a set of antecedent propositions and the truth values of a set of consequences propositions.

Output: input ($W^{(k)}_{\{r(i)\}n}$)and output (W_O)weights

```
Step 1 Build the set of user input places $P_I$ .
Step 2 Build the set of initially enabled transitions $T_{initial}$ .
do
{ Step 3 calculate the input weights as above (1).
Step 4 calculate the output weights $W_O = \delta$ (cf', $\alpha(P_j)$)
according to the types of firing rules. }
 Step 5 while( $T_{curren}t = \varphi$ )
```

3. INTELLIGENT E-LEARNING SYSTEM

In the recent years, educational developers have begun to focus on the design of learning management systems, in an efforts to support e-learning in many different disciplines, context and fields of practice. These general-purpose systems have sought to enable a range of learner interactions with textual and audio-visual material-through reading , listing and viewing, pursuing a variety of set tasks and participating in discussions (synchronous, asynchronous). As well, developers have addressed the challenge of providing so-called "intelligent" support of such activities and of supplying information to educators about learners' progression through them. However, though such learning management systems sometimes provide interesting learning opportunities, they are rarely underpinned by a powerful, explanatorily coherent theory learning. Consequently, whilst learners are routinely channeled through a range of e-learning activities, much significant learning is more likely to occur by chance than by design.

3.1 Problem Statements Online Learning System

Consider, two learners discussing a concept about AVL-Trees on the Web. First learner prefers to begin with written concept, more explanation subject, and does not like drawing. Second one prefers to begin with drawn figures about concept, more example about subject, and does not like written concept. The problem comes when the first learner explains AVL-Trees concept to the second learner. Obviously, the second learner will find some difficulties to understand the concept. Consequently he will loose confidence in the first learner and maybe leave the discussion. Moreover, the problem will be more complex if the two learners have two different knowledge levels.

3.2 Existing Solution

The existing web-based tutoring systems can support this group with two learning systems: Cooperative systems without adaptive, and Adaptive systems without cooperative.

- **Cooperative Systems Without adaptive** :

The tools of these systems are e-mail, chat-room, video Conference, two learners. These systems Just transfer information, do not interact with learners, do not support learners with extra information, and do not let acquire any new knowledge from learner.

- **Adaptive Systems Without Cooperation**

The tools of these systems are MetaDoc**,** ELM-ART, internet**,** and only one learner. These systems are These systems are Interact with learner, adapt and support learners with extra information, do not let acquire any new knowledge from learner, do not support interaction between a group of learner, and do not adapt their own knowledge.

- **Problem Related to these Systems**

Lack of sufficient interactions between learners, Lack of knowledge about learners such as their knowledge levels, learning styles, and past performance, Lack of confidence of the learner toward the system, and Lack of motivation building and positive reinforcement. A uniform teaching style supposed to suit all learners.

In web learning system, the design of multi-user applications to support the synchronous collaborative learning environment offers many more challenges: communication, coordination, and awareness and use the model WYSIWIS (what you see is what I see). But, in the Intelligent Web Learning system, The adaptation aspect have to be taken into consideration. So, we apply a new trend that we call ”What We See Is What We Need to See” (WWSIWWNS). The WWSIWWNS means that both a learner and helper can see two different user-interface adapted to their needs.

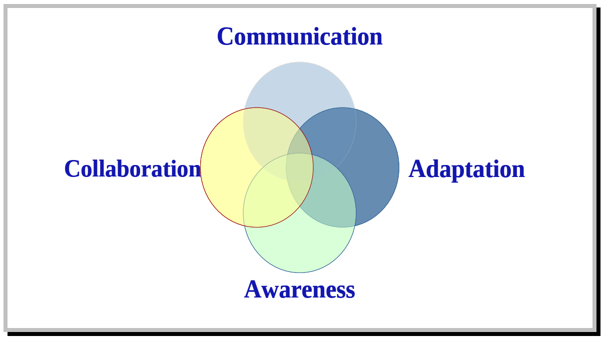


Fig. 4. Intelligent E- Learning system

4. E-LEARNING COURSE DEVELOPMENT USING MOODLE

Moodle (Modular Object-Oriented Dynamic Learning Environment) is used as a tool to enhance learning system that enables us to create powerful, flexible, and engaging on line learning experiences. Moodle is an open source course management system for creating dynamic (Adaptive) online learning communities and for supplementing face-to-face learning.

By using Moodle, CMS(Course Management Systems) give educators tools to create a course web site and provide access control so only enrolled students can view it. Aside from access control, CMS offer a wide variety of tools that can make your course more effective. They provide an easy way to upload and share materials, hold online discussions and chats, give quizzes and surveys, gather and review assignments, and record grades. Moodle can enables us to add five kinds of static course material. This is course material that a student reads, but not interact with:

- A text page.
- A web page.
- A linking to any on the web (including material on your Moodle site).
- A view into one of the directories course.
- A label that displays any text or image

Also, Moodle can add six types of interactive course material, that allow to the student to interact with, by answering questions, entering text, uploading files:(assignment, choice, journal, lesson, quiz, and survey). Moreover, Moodle offers five kinds of activities where students interact with each other (chat, forum, glossary, wiki, and workshop).

Also, Moodle can support SCORM(Sharable Content Object Reference Model). Most web content consists of simple hyperlinks from one page to another. SCORM enables complex directed learning experiences that go far beyond what can be done with simple hyperlinked web content.

4.1 The Moodle Architecture

Moodle runs on any web server that supports the PHP programming language, and a database. It works best, and there is more support, when running on the Apache web server with a MySQL database. The Moodle learning management system resides in three places on our web host:

- The application occupies one directory, with many subdirectories for various modules.
- Data files students and teachers upload , such as photos and assignments submitted by students, reside in the Moodle data directory.
- Course material that you create with Moodle (web pages, quizzes, workshops, lessons, etc.), grades, user information, and user logs resides in the Moodle database.

4.2 ADDIE Reasoning Algorithm by AFHOPN

Algorithm C:ADDIE(Analysis- Design- Development- Implementation- Evaluation)

Input : the goal proposition d_k

Output : the learning content structure after the goal

proposition is input

```
Step1: Analysis
    1.1 Describe the learners.
    1.2 Define the production rules
Step 2 : Design
    2.1 Select the approach .
    2.2 Generate Tokens.
    2.3 Create Social Materials.
Step 3 :Development
    3.1 Resolving Production Rules.
```

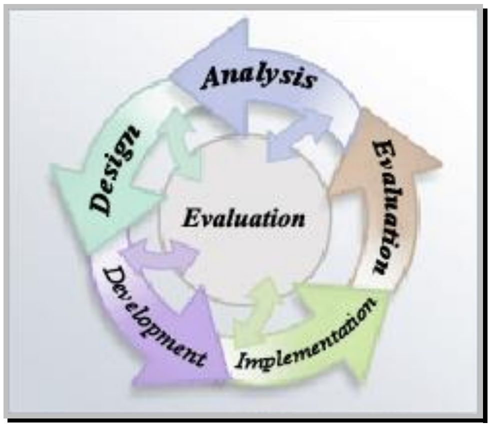


Fig. 5 .Steps of ADDIE reasoning algorithm

Step 4 :Implementation

4.1 prepare resources.

4.2 Design E-course conform to SCORM

Step 5: Evaluation

5.1 Manage the process (calculate a new parameters).

5.2 Evaluate E-learning course which implemented by

Moodle

Example 1: Here, by using a simple example, a process of forming a learning sequence is demonstrated.

Step1: Analysis, we suppose that, the system has recorded about five learners' behavior. PR1: IF P1 AND P2 AND P3 AND P4 AND P5 THEN P6 .

Step 2 : Design, AFHOPN model for this expert system may be represented as shown in fig 6. a set of output data (fuzzy truth values of output variables) can be calculated according to the input data (fuzzy truth values of input variables). The (WI's) weights in our fuzzy production rules play an important role in adaptive web learning (WI's weights represented the learning rate (lR) from Moodle (i.e. using interactive course material (assignment, choice, lesson, quiz, …) and chat, forum, …)). I(Grade), I(RR) represent, represent the input value of final test grade, learning rate from Moodle, where I(Grade) ={1, 0.72, 0.53, 0.47, 0}and I(lR)={1, 0.75, 0.61, 0.66, 0}

Step 3 :Development

Here, we need to calculate learning factor α(P_6)Assume the firing conditions satisfied , After firing the transition t_1, firstly, we calculate the new cf(denoted by cf'):

$\delta = \min(\gamma^{(k)}_{r(1)}, \gamma^{(k)}_{r(2)}, \ldots, \gamma^{(k)}_{r(n)}) = [0.48 \quad 0.54 \quad 0.323 \quad 0.310 \quad 0.42\,] = 0.310$

then cf'(t_1)= δ.cf(t_1)= (0.310) .(0.5) = 0.155

$\alpha(P6) = W^{\overset{1}{6}} .\ cf'(t_1)$

$= (0.85).(0.155) = 0.1318.$

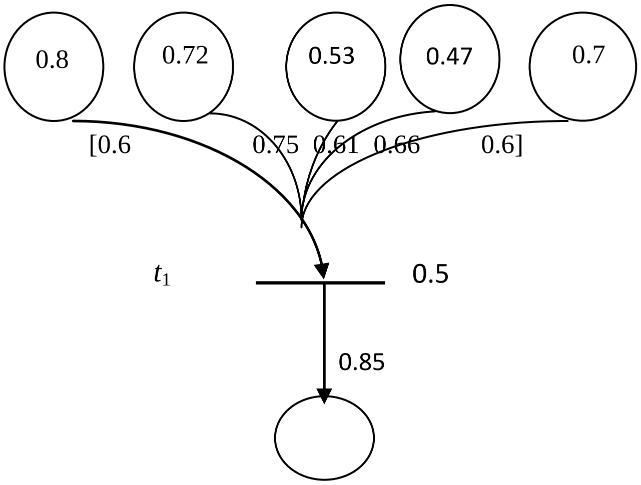


Fig. 6. An AFHOPN example by using lR

By using the algebraic equation we will get the truth value of the input and the output place

$$\alpha(P_{i+1}) = \begin{bmatrix} 0.32 \\ 0.18 \\ 0.207 \\ 0.16 \\ 0.28 \\ 0.1318 \end{bmatrix}$$

Step 4 :Implementation

Fig.7. show the Interface of the course which implemented by Moodle conform to SCORM.

Step 5: Evaluation

After training the given data we need to adjust the weight therefore we apply the algorithm B for updating weights and Fig. 11 show the new truth values from according to Example 1:

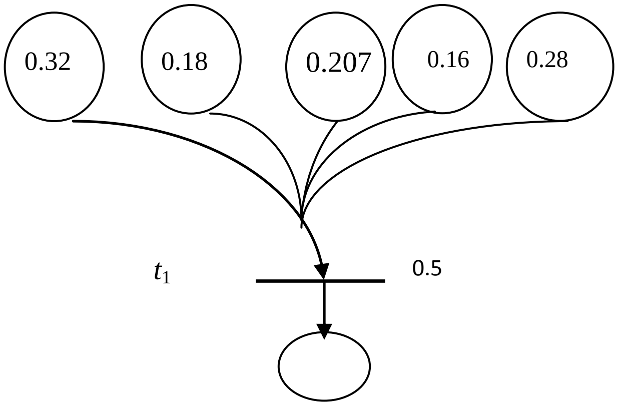


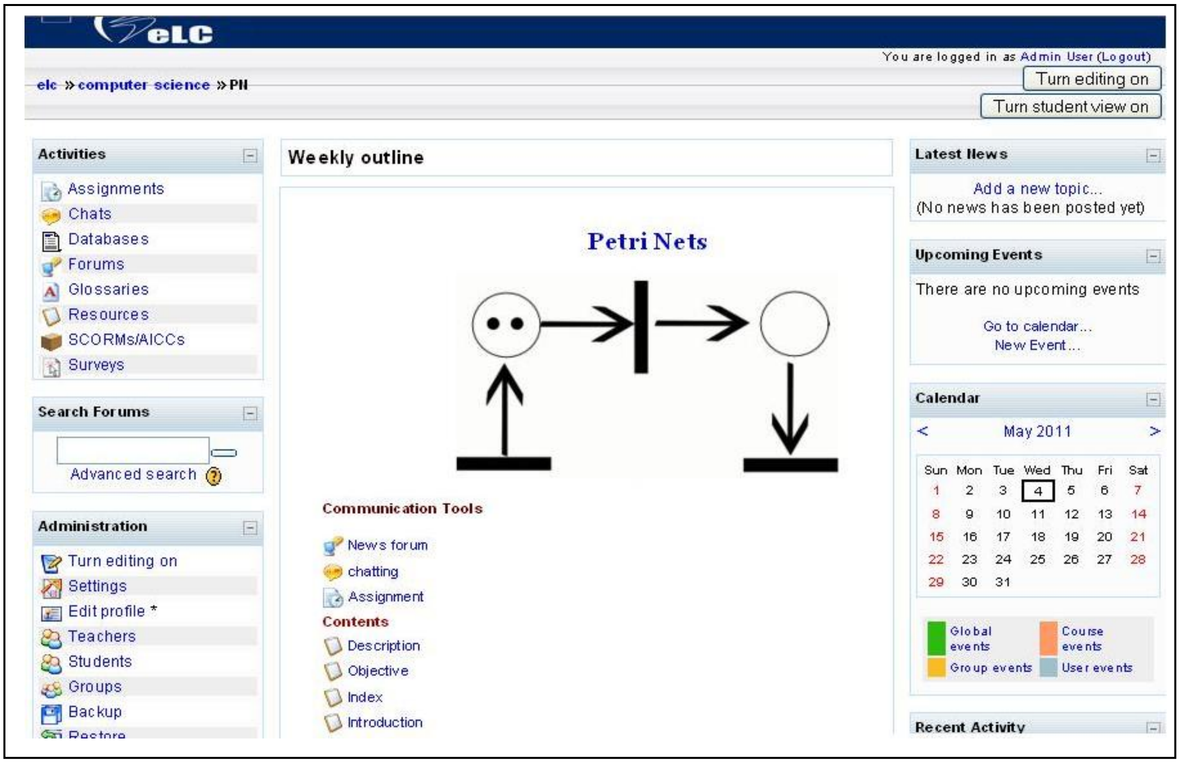
Fig. 8. An AFHOPN example by using RR

To calculate the input weight (learning rate) by

$$W^{(k)}_{\{r(i)\}n}\ (x+1) = W^{(k)}_{\{r(i)\}n}\ (x) + (\delta\ \ \gamma^{(k)}_{\{r(i)\}n} - \alpha(P_j))$$

$$W^{(1)}_{1,2,3,4,5}\ (x+1) = \begin{bmatrix} 0.71 \\ 0.78 \\ 0.69 \\ 0.7 \\ 0.8 \\ 0.92 \end{bmatrix}$$

We note that, the values of new input weights are greater than its old . Therefore, the learning rate is increased after training



You are logged in as Admin User (Logout)

(No news has been posted yet)

There are no upcoming events

< May 2011 >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Fig. 7. Interface of the course which implemented by Moodle conform to SCORM

5. CONCLUSION

We introduced a class of Petri nets , the adaptive fuzzy higher order Petri nets, that is dynamically adjusted the parameters, allows a structural representation of knowledge and has got systematic procedure for supporting fuzzy reasoning with the features: it is suitable for vague and dynamic knowledge, based on transition firing rules. adaptive fuzzy higher order Petri nets use as a dynamic tool for measuring the learning rate. Modular Object-Oriented Dynamic Learning Environment (Moodle) are introduced for creating dynamic online learning communities and generating automatic course development. Moodle helps the teachers to perform precise formative assessments according to e-learning tools and to devote themselves for teaching, since they save significant time in performing learning evaluation.

6. ACKNOWLEDGMENTS

We are grateful to the anonymous referees for their constructive suggestions concerning improvement in the presentation of this paper.

7. REFERENCES

[1] Liaw, S., Huang, H., and Chen, G. 2007. Surveying instructor and learner attitudes towards e-learning, Computers and Education, vol. 49, pp. 1066-1080.

[2] Brusilovsky, P. and Miller, P. 2001. Course Delivery Systems for the virtual University. In DELLA SENTA, T.&TSCHANG, T.(Eds.) Access to knowledge: new information technologies and emergence of the virtual University. Amsterdam. Elsevier Science.

[3] Chang, C. 2002. Building a Web-based learning portfolio for authentic assessment, in Proc. Int. Conf. Comput. Educ., vol. 1, pp. 129–133.

[4] Brusilovsky, P. and Pyelo, C. 2003. Adaptive and Intelligent Web-based Educational Systems. International Journal of Artificial Intelligence in Education, vol. 13, pp. 159-172.

[5] Brown, E., Cristea, A., and Stewart, C. 2005. Pattern in authoring of Adaptive Educational Hypermedia: A taxonomy of learning styles. Educational Technology and Society, vol. 8, pp. 77-90.

[6] Chen, J., Huang, Y., and Chu, W. 2005. Applying dynamic fuzzy Petri net to web learning system. Interactive Environments, vol. 13 ,no. 3, pp. 159-178.

[7] Huang Y. M., Chen J. N., Huang, T.C., L.eng, Y. L., and Kuo, Y.H., 2008. Standardized course generation process using dynamic fuzzy Petri nets, Expert Systems with Applications, vol. 34, pp. 72-86.

[8] Murata, T. 1989. Petri Nets: Properties, Analysis and Applications," Proc. IEEE, vol. 77, pp.541-580.

[9] Peterson, J. 1981. Petri Net Theory and Modeling of Systems, Englewood Cliffs, NJ: Prentice-Hall.

[10] Chen S., Ke J., and Chang, J. 1990. Knowledge representation using fuzzy Petri nets, IEEE Trans. Knowl. Data Eng. Vol. 2, no.3, pp. 311-319.

[11] Ahson, S. 1995. Petri net models of fuzzy neural networks, IEEE Trans. Syst., Man Cybern. Vol. 25, pp. 962-932.

[12] Chow, T., and Li, J.Y.1997. Higher-order Petri net models based on artificial neural networks , Artificial Intelligence, vol.29, no.1,pp.289-300.

[13] Li, Z., C.Zhou, M., and Wu, N. 2008. A survey and comparison of Petri net-based deadlock preventation policies for flexible manufacturing systems", *IEEE* Trans. Syst. , Man, Cybern.C, Appl. Rev., pp.137-188.

[14] Li X., Y, W., and Lara-Rosano, F. 2000. Dynamic knowledge inference and learning under adaptive fuzzy Petri net framework," IEEE Trans. Syst., Man Cybern-Part C: Appl Rev, vol. 30, no. 4, pp. 442-450.