

Thesis summarization

# **A User Modeling System for Adaptive Learning**

# A User Modeling System for Adaptive Learning

*Send with full respect:*

*Short report:* A User Modeling System for Adaptive Learning

*Supervisor:* Pr. Dong Thi Bich Thuy

*Researcher:* Nguyen Phuoc Loc

## Abstract

Softwares and information systems nowadays tend to personalization services but it is necessary to have full of knowledge about users in order to serve them as well as possible. So user information storage and management systems so-called user modeling systems (UMS) become more important and user information that is constructed as concrete model gets more valuable.

The thesis focuses on adaptive learning context and so, user is learner and user model is learner model. The thesis aims to build up a user modeling system with full of basic modeling functions such as collecting and managing user information. Moreover, such user modeling system has strong inference mechanism based on solid user model.

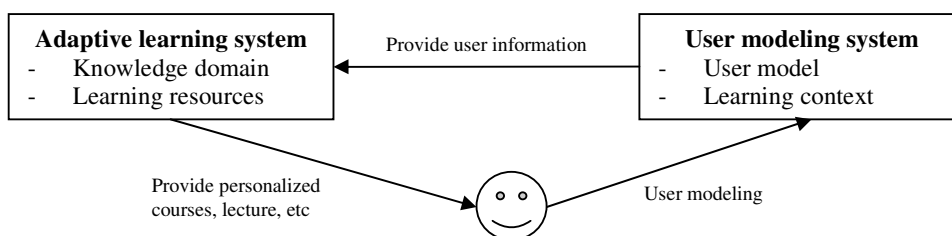
The summarization has four chapters:

- Chapter 1: Introduction to existing user modeling systems and thesis' objectives
- Chapter 2: The architecture of proposed user model and user modeling system
- Chapter 3: Three sub-models of proposed user model
  - o 3.1: Knowledge sub-model denoted *K* sub-model
  - o 3.2: Learning style sub-model denoted *LS* sub-model
  - o 3.3: Learning history sub-model *LH* sub-model
- Chapter 4: Evaluating adaptive learning model
- Chapter 5: Conclusion and future trend

## 1. Introduction

User model is the collection of user's properties such as knowledge, characteristics, interests, background, demography, etc. Information is wide and the importance level of such information varies with regard to context, for example, knowledge, learning styles, goals and backgrounds are the most important in learning context – the main context of thesis. Users also depend on context and so users are learners and customers in learning context and commercial context, respectively. Terms such as user model, learner model, and student model have the same meaning in thesis.

Systems that build up and manage user model is called user modeling system (UMS), please imagine that user model is similar to database and user modeling system is similar to database management system (DBMS). User modeling system provides user information to adaptive system so that adaptive system gives the best service to users. In e-learning, adaptive system supplies learner teaching methods, courses, lectures, tests and exercises which are appropriate to each individual, based on her/his personal information provided by user modeling system.



**Figure 1.1.** User modeling system

Existing user modeling systems are classified into four groups:

- User modeling shell is now outdated because it is component attached to adaptive system and so it is not specialized user modeling system.
- User modeling server works in specialized and independent manner as database management systems but it is restricted on inferring new potential information from user model.
- Agent-based user modeling system works by interaction of agents, therefore each agent manages a piece of user information. Agents are defined as software modules that operate independently and interact together. Agent-based user modeling system is distributed system.
- Mobile user model storages user information on mobile devices such as smart phone, mobile phone, etc. These systems update user information in instant way and get high interaction with users but are limited to capacity.

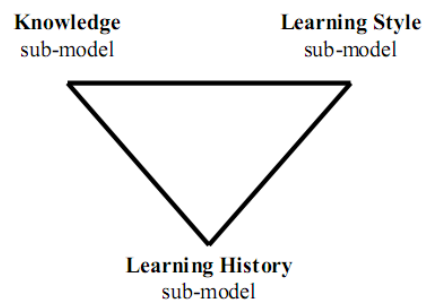
Each user modeling system has own strong points and drawbacks but in general, these systems focus on information storage and provision and so there is lack of strong inference mechanism. Modern user modeling systems differ from database management system by the ability of inference; collective user information is not much enough to need of adaptive systems and so it is required to exploit new potential information from existing information in user model. Moreover, it is necessary to choose essential users' attributes in learning context because information about user is very much; we cannot describe and model all of such information. Therefore, thesis has two objectives:

- The first objective is to build up a user modeling system incorporated with inference mechanism based on solid algorithms and mathematical tools, which aims to draw new potential information about users.
- The second objective is to select and model the most essential characteristics about learners in learning context because the accuracy of inference mechanism depends on the importance level of information. That information is too much or less important will decrease performance of inference process. So the second objective supports the first objective.

## 2. Architecture of user modeling system

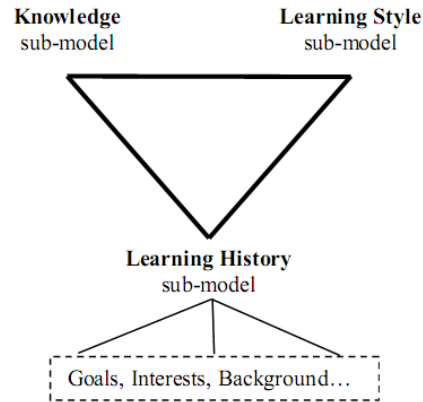
This research proposes a learner model that consists of three essential kinds of information about learners such as knowledge (*K*), learning style (*LS*) and learning history (*LH*) which are fine-tuned from a lot of user information. Such three characteristics are both mutually independent and coherent in order to form a triangle and so this learner model is called Triangular Learner Model (TLM). TLM will cover the whole of user's information required by learning adaptation process and give the best support to adaptive learning. The reasons for such assertion are:

- Knowledge, learning styles and learning history are prerequisite for modeling learner.
- While learning history and knowledge change themselves frequently, learning styles are relatively stable. The combination of them ensures the integrity of information about learner.
- User knowledge is domain specific information and learning styles are personal traits. The combination of them supports user modeling system to take full advantages of both domain specific information and domain independent information in user model.



**Figure 2.1.** Triangular Learner Model

Knowledge, learning styles and learning history are sub-models of TLM. TLM is designed according to triangular model so that it is easy to describe and construct other user other information such as interests and goals. In other words, TLM can be extended to interpret more detailed about learner by attaching more learners' characteristics such as interests, background, goals... into the learning history (LH) sub-model. So learning history sub-model is the most important one, which used to initialize two other sub-models such knowledge and learning style.

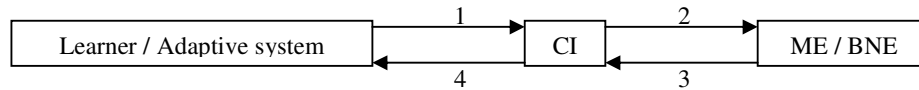


**Figure 2.2.** extended Triangular Learner Model

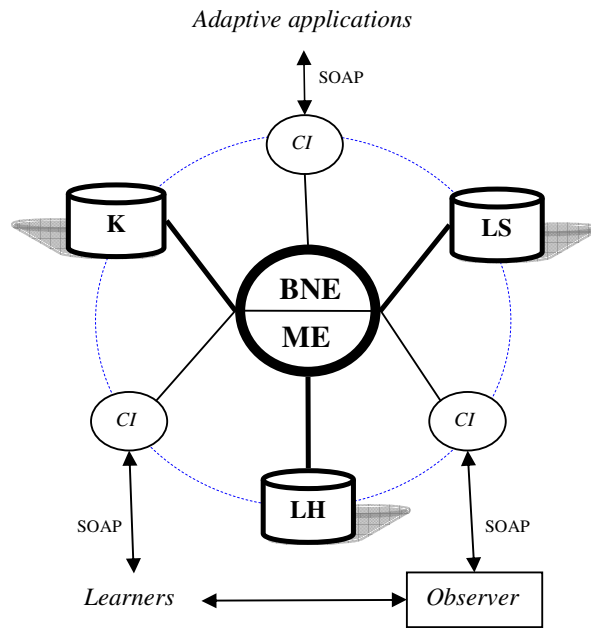
TLM finishes the second objective and the first objective is achieved by the proposal of user modeling system so-called Zebra: constructing TLM, managing TLM, and inferring new information from TLM. The core of Zebra is the composition of two engines: mining engine (ME) and belief network engine (BNE).

- Mining engine (ME) is responsible for collecting learners' data, monitoring their actions, structuring and updating TLM. Mining engine totally uses mining technique to perform modeling tasks. Mining engine is very important when it manages whole TLM but it focuses mainly on learning history sub-model. ME also provides input information to BNE. Mining engine has three other important functionalities that are to discover some other characteristics beyond knowledge and learning styles such as interests, goals, learning context, etc and to support learning concept recommendation and to support collaborative learning.
- Belief network engine (BNE) is responsible for inferring new personal traits from TLM by using deduction mechanism available in belief network. This engine applies both Bayesian network and hidden Markov into its tasks. Two sub-models: knowledge & learning style are managed by this engine. BNE does not work as actively as ME but it is more complicated than ME.

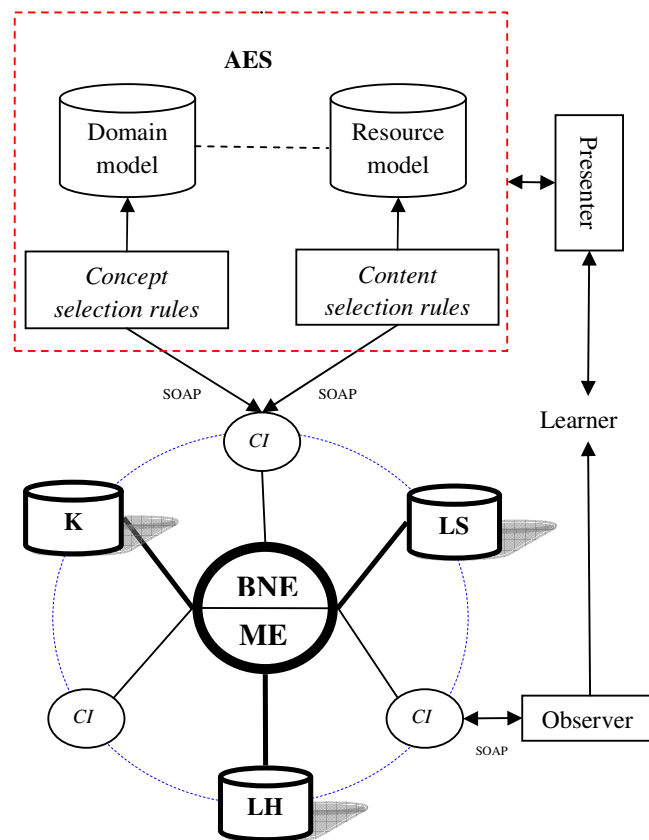
Zebra provides communication interfaces (CI) that allows users and adaptive systems to see or modify restrictedly their TLM. Because user modeling system always interacts with adaptive system and TLM cannot be modified directly from outside, CI operates as communication port. Following is the adaptive process:



1. Adaptive system send user information query to CI via protocols such as SOAP, HTTP, RMI, etc.
2. CI analyzes query request into input parameters and sends such parameters to respective ME or BNE.
3. ME or BNE executes query according to specified parameters and send results back CI.
4. CI sends results back adaptive system.



**Figure 2.3.** The architecture of Zebra



**Figure 2.4.** The interaction between adaptive system AES and user modeling system Zebra

### 3. Three sub-models in Triangular Learner Model

The previous chapter gives us a general architecture of Triangular Learner Model (TLM) and its user modeling system Zebra. TLM is composed of three sub-models such as knowledge sub-model, learning style sub-model and learning history sub-model. This chapter is the most detailed one that describes thoroughly these sub-models together with their attributes, constructing method and inference mechanism.

#### 3.1. Knowledge sub-model (K)

Knowledge is study result and very important to adaptive system and so it is the evaluation of adaptive system. A preeminent adaptive system will improve learners' knowledge after they study on on-line course via such adaptive system. The thesis gives the proposal of knowledge sub-model along with the ways to construct and enhance knowledge sub-model.

#### Proposed knowledge model: combination of overlay model and Bayesian network

The thesis proposes the combination of overlay model and Bayesian network. In overlay model, the domain is decomposed into a set of knowledge elements and each element is a lecture, exercise or test and represented by an integer number that measure user knowledge. Each learner is modeled as a sub-set of masteries over those elements and such sub-set is called overlay model.

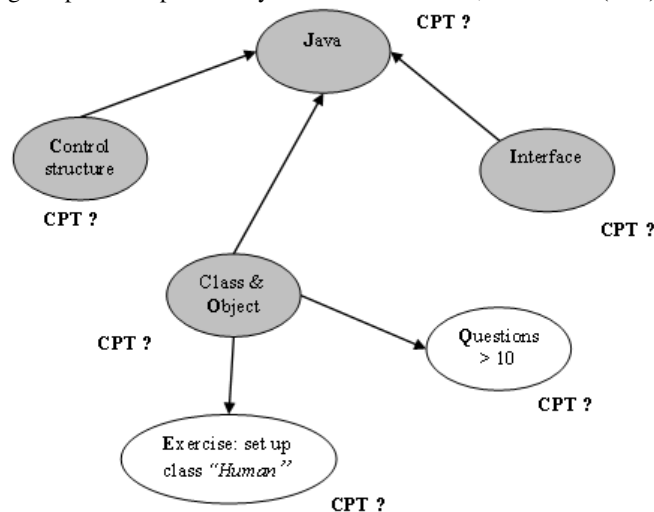
Bayesian network [Neapolitan 2003] is a directed acyclic graph (DAG) composed of a set of nodes and a set of directed arcs. Each arc indicates the relationship between two nodes and the strength of this relationship is quantified by the conditional probability table (CPT). The inference mechanism in Bayesian network is based on Bayes' theorem. The combination between overlay model and BN is done through following steps:

1. The structure of overlay model is translated into Bayesian network, each user knowledge element becomes a node in Bayesian network. Evaluation knowledge such as test or exercise becomes evidence node and normal knowledge such as lecture or lesson becomes hypothesis node or hidden node.
2. The relationship between domain elements in overlay model becomes a conditional dependence assertion signified by CPT of each node in Bayesian network.

Following is an example of combination between overlay model and Bayesian network. Suppose Java course for students is constituted of 6 knowledge elements:

- A top-most course *Java* (*J*) and three lectures: *control structure* (*C*), *class & object* (*O*), *interface* (*I*).
- An exercise: *set up class Human* (*E*).
- A test that students need answer accurately more than 10 answers: *Questions > 10* (*Q*).

The problem needs solved is that how much knowledge a student gains after she / he finished Java course when student knowledge is marginal posterior probability of node *J* with  $J = 1$ , denoted  $Pr(J=1)$ .



**Figure 3.1.** Combination of Bayesian network and overlay model (evidence nodes are unshaded, otherwise, hidden nodes are shaded)

$J, C, O, I$  are four hidden nodes (shaded),  $E$  and  $Q$  are two evidence nodes (unshaded). Because the strength of each arc quantified by a CPT is not determined yet, it is impossible to calculate the marginal posterior probability. The thesis proposes two-step process to determine CPT:

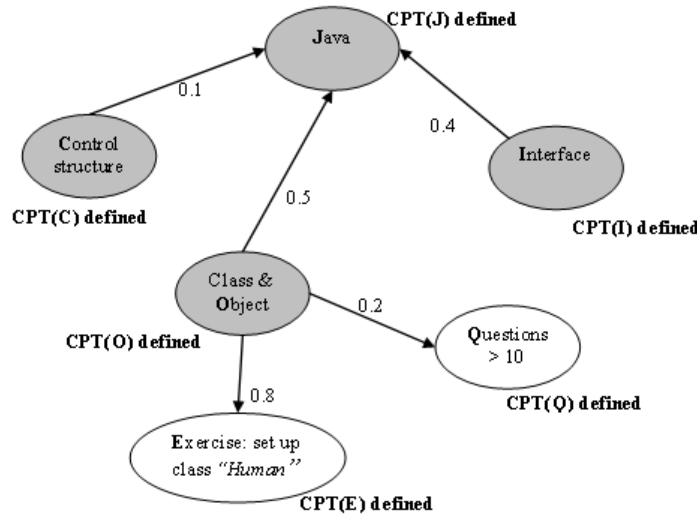
1. Each arc is assigned by the weight representing the importance level of a lesson, lecture, test or exercise.
2. Union formula is used to create CPT based on these weights.

Suppose weights of lectures *control structure* ( $C$ ), *class & object* ( $O$ ), *interface* ( $I$ ) are  $w_1=0.1$ ,  $w_2=0.5$ ,  $w_3=0.4$ , respectively. Lectures *class & object* ( $O$ ) are the most important, in consecutive, there are *interface* ( $I$ ) and *control structure* ( $C$ ). The weights of exercise  $E$  and test  $Q$  are  $0.8$  and  $0.2$ , respectively. The CPT of Java course is determined via following Union formula:

$$Pr(J \mid C, O, I) = w_1 * h_1 + w_2 * h_2 + w_3 * h_3$$

(Please section III.1.1.3 in thesis for the proof of Union formula)

$$\text{Where } h_1 = \begin{cases} 1 & \text{if } C = J \\ 0 & \text{otherwise} \end{cases} \quad h_2 = \begin{cases} 1 & \text{if } O = J \\ 0 & \text{otherwise} \end{cases} \quad h_3 = \begin{cases} 1 & \text{if } I = J \\ 0 & \text{otherwise} \end{cases}$$



**Figure 3.2.** Bayesian overlay model is specified completely with full of CPT (s)

C	O	I	Pr(J = 1)	Pr(J = 0) 1 - p(J = 1)
1	1	1	<b>1.0</b> ( $0.1*1 + 0.5*1 + 0.4*1$ )	<b>0.0</b>
1	1	0	<b>0.6</b> ( $0.1*1 + 0.5*1 + 0.4*0$ )	<b>0.4</b>
1	0	1	<b>0.5</b> ( $0.1*1 + 0.5*0 + 0.4*1$ )	<b>0.5</b>
1	0	0	<b>0.1</b> ( $0.1*1 + 0.5*0 + 0.4*0$ )	<b>0.9</b>
0	1	1	<b>0.9</b> ( $0.1*0 + 0.5*1 + 0.4*1$ )	<b>0.1</b>
0	1	0	<b>0.5</b> ( $0.1*0 + 0.5*1 + 0.4*0$ )	<b>0.5</b>
0	0	1	<b>0.4</b> ( $0.1*0 + 0.5*0 + 0.4*1$ )	<b>0.4</b>
0	0	0	<b>0.0</b> ( $0.1*0 + 0.5*0 + 0.4*0$ )	<b>1.0</b>

**Table 3.1.** CPT of node Java ( $J$ )

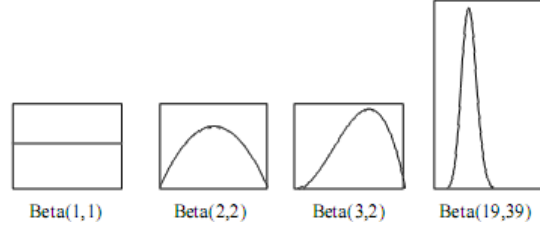
When Bayesian network is specified completely with full of CPT (s), it is easy to compute the posterior marginal probability  $Pr(J=1)$ . Knowledge sub-model is improved by two methods:

- Parameter learning: CPT (s) – Bayesian parameters, determined via the combination of weight specification and Union formula, needs enhanced after Zebra operates in long time process. This is a kind of parameter learning. Research applies Expectation Maximization (EM) algorithm [Neapolitan 2003], [Dellaert 2002] and proposes Maximum Likelihood Estimation (MLE) algorithm for learning parameters.
- Structure learning: structures of nodes, arcs and CPT (s) can be very different from Bayesian sub-model derived from the combination of overlay model and Bayesian network. Thesis proposes an optimal algorithm that applies dynamic Bayesian network into structure learning.

### Improving knowledge sub-model via EM algorithm

EM technique assumes that probability density function  $\beta(f)$  for each CPT conforms beta distribution:

$$\beta(f) = \frac{\Gamma(N)}{\Gamma(a)\Gamma(b)} f^{a-1} (1-f)^{b-1}$$



**Figure 3.3.** Beta distribution

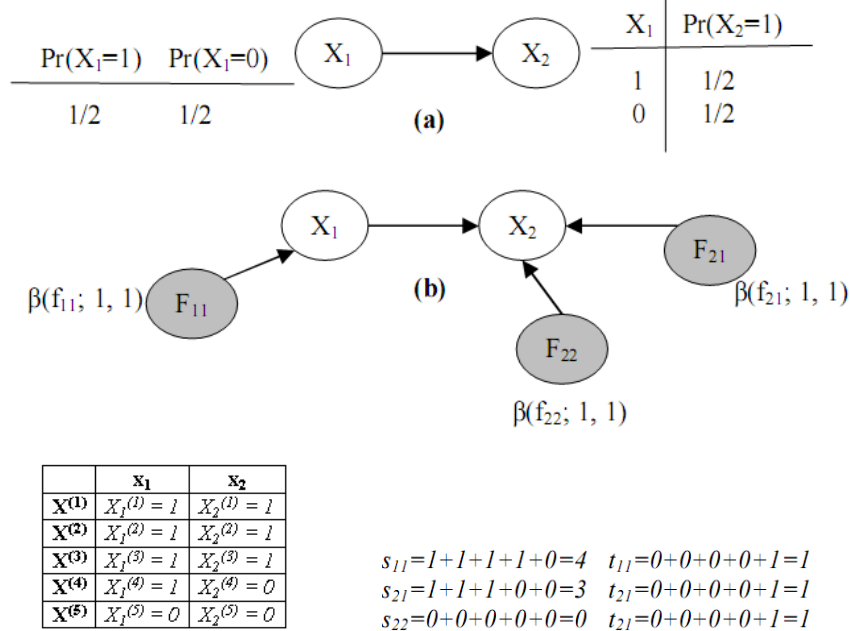
Each node in Bayesian network is considered a random variable. The probability of  $X$  is:

$$\Pr(X = 1) = E(\beta(f)) = \frac{a}{a+b}$$

Where “ $a$ ” denotes the number of successful outcomes (for example,  $f=1$ ) in  $N = a+b$  trials. The probability of  $X$  after  $M = s + t$  trials is:

$$\Pr(X = 1|M \text{ trials}) = E(\beta(f)|M \text{ trials}) = \frac{a+s}{a+b+s+t} = \frac{a+s}{N+M}$$

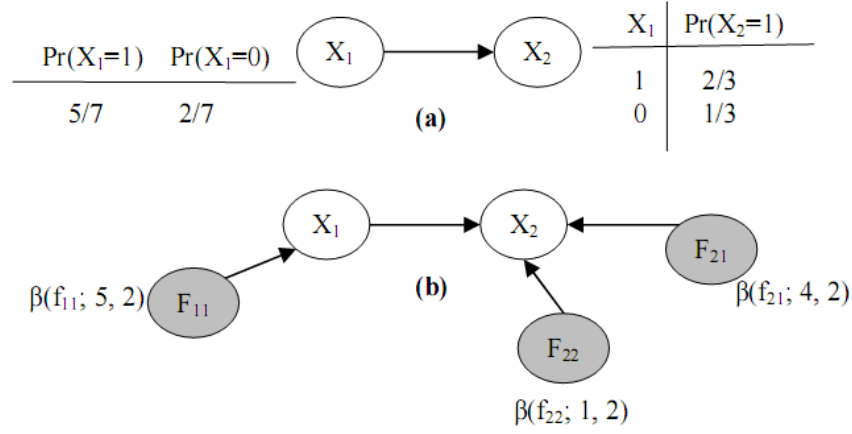
Where “ $s$ ” denotes the number of successful outcomes in  $M = s + t$  trials. Suppose following Bayesian network is specified via 5 trials  $X^{(1)}, X^{(2)}, X^{(3)}, X^{(4)}, X^{(5)}$ :



**Figure 3.4.** Bayesian network, CPT and trials

EM algorithm accumulates the number of successful outcomes after each trial in order to change the probability of each node  $X_i$ .





$$\begin{aligned}\beta(f_{11}|E) &= \beta(f_{11}; a_{11}+s_{11}, b_{11}+t_{11}) = \beta(f_{11}; 1+4, 1+1) = \beta(f_{11}; 5, 2) \\ \beta(f_{21}|E) &= \beta(f_{21}; a_{21}+s_{21}, b_{21}+t_{21}) = \beta(f_{21}; 1+3, 1+1) = \beta(f_{21}; 4, 2) \\ \beta(f_{22}|E) &= \beta(f_{22}; a_{22}+s_{22}, b_{22}+t_{22}) = \beta(f_{22}; 1+0, 1+1) = \beta(f_{22}; 1, 2)\end{aligned}$$

$$\begin{aligned}s_{11} &= 1+1+1+1+0=4 & t_{11} &= 0+0+0+0+1=1 \\ s_{21} &= 1+1+1+0+0=3 & t_{21} &= 0+0+0+0+1=1 \\ s_{22} &= 0+0+0+0+0=0 & t_{22} &= 0+0+0+0+1=1\end{aligned}$$

**Figure 3.5.** Bayesian network and its CPT (s) learned via EM algorithm

If there are missing values in training data, please see section III.1.3.2 for EM algorithm in incomplete data. EM algorithm is simple and very effective after each iteration. In case of huge training data, maximum likelihood estimation (MLE) algorithm is more appropriate to learning parameters.

#### Improving knowledge sub-model via MLE algorithm

The basic idea of MLE algorithm is to find out the maximum value of likelihood function. Given beta function  $B(x, y) = \Gamma(x) \Gamma(y) / \Gamma(x + y)$  where  $\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt$  is gamma function. The likelihood function of distribution beta is:

$$L(a, b) = \prod_{i=1}^n f(x_i, a, b) = \prod_{i=1}^n \frac{1}{B(a, b)} x_i^{a-1} (1-x_i)^{b-1} = \frac{1}{B^n(a, b)} \prod_{i=1}^n x_i^{a-1} \prod_{i=1}^n (1-x_i)^{b-1}$$

The likelihood is taken logarithm as following:

$$\ln L(a, b) = -n \ln(B(a, b)) + (a-1) \sum_{i=1}^n \ln x_i + (b-1) \sum_{i=1}^n \ln(1-x_i)$$

The goal of MLE algorithm is to find out two parameters  $a$  and  $b$  so that the log likelihood function gets maximal. After taking the first partial derivatives of log likelihood function with subject to  $a$  and  $b$ , thesis proves that parameters  $a$  and  $b$  are satisfied following set of equations:

$$\begin{cases} e^a \sum_{k=1}^b \frac{(-1)^k}{k} \binom{b}{k} = \frac{1}{n} \sum_{i=1}^n \ln x_i \\ e^b \sum_{k=1}^a \frac{(-1)^k}{k} \binom{a}{k} = \frac{1}{n} \sum_{i=1}^n \ln(1-x_i) \end{cases} \Leftrightarrow \begin{cases} F_1(a, b) = L_1 \\ F_2(a, b) = L_2 \end{cases}$$

Where  $F_1(a, b) = e^a \sum_{k=1}^b \frac{(-1)^k}{k} \binom{b}{k}$ ,  $L_1 = \frac{1}{n} \sum_{i=1}^n \ln x_i$ ,  $F_2(a, b) = e^b \sum_{k=1}^a \frac{(-1)^k}{k} \binom{a}{k}$ ,  $L_2 = \frac{1}{n} \sum_{i=1}^n \ln(1-x_i)$

This thesis proposes an iterative algorithm to find out approximate solutions  $a$  and  $b$  of above set of equations. The iterative algorithm is based on the proposition “The range of variables  $a$  and  $b$  is from 1 to  $n$  where  $n$  is the whole positive number and not greater than the number of evidences in training data”. Each pair values  $(a_i, b_i)$  which are values of variables  $a$  and  $b$  are fed to  $F_1, F_2$  at each iteration. Two biases  $\Delta_1 = F_1(a_i, b_i) - L_1$  and  $\Delta_2 = F_2(a_i, b_i) - L_2$  are computed. The normal bias is the root of sum of the second power  $\Delta_1$  and the second power of  $\Delta_2$ :  $\Delta = \sqrt{\Delta_1^2 + \Delta_2^2}$ . The pair  $(\hat{a}, \hat{b})$  whose normal bias  $\Delta$  is minimum are chosen as the parameter estimators. The algorithm is described as below:

```

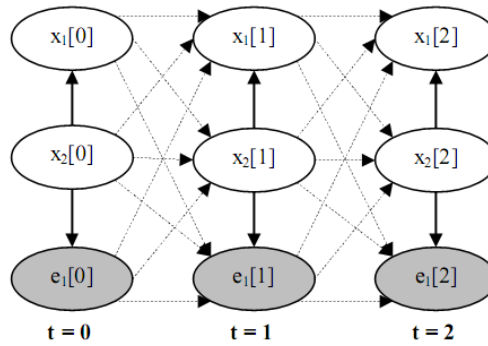
min  $\Delta = +\infty$ ;
 $\hat{a} = \hat{b} = 1$  (uniform distribution)
For  $a = 1$  to  $n$  do
  For  $b = 1$  to  $n$  do
     $\Delta_1 = F_1(a, b) - L_1$ 
     $\Delta_2 = F_2(a, b) - L_2$ 
     $\Delta = \sqrt{\Delta_1^2 + \Delta_2^2}$ 
    If  $\Delta < \min \Delta$  then
      min  $\Delta = \Delta$ 
       $\hat{a} = a$ 
       $\hat{b} = b$ 
    End If
  End for  $a$ 
End For  $b$ 
( $\hat{a}$  and  $\hat{b}$  are optimal parameters)

```

### Improving knowledge sub-model via dynamic Bayesian network

The strong point of static Bayesian network is simple and its inference mechanism is very effective but it cannot model temporal relationships between variables. In e-learning context, it is very necessary to monitor chronologically users' process of gaining knowledge. Dynamic Bayesian network (DBN) [Neapolitan 2003] consists of a series of  $G_t$  where  $G_t$  is static Bayesian network at time point  $t$ :

- An initial Bayesian network  $G_0$  at time point 0 containing a set of nodes, arcs and CPT (s).
- Bayesian network  $G_t$  at time point  $t$  contains not only a set of nodes, arcs and CPT (s) but also a transition probability distribution  $Pr_{\rightarrow}(G_t | G_{t-1})$  representing the transition (state) from  $G_{t-1}$  to  $G_t$ . If the temporal process go through  $n$  transition (states), there will be  $n - 1$  transition probability distributions.



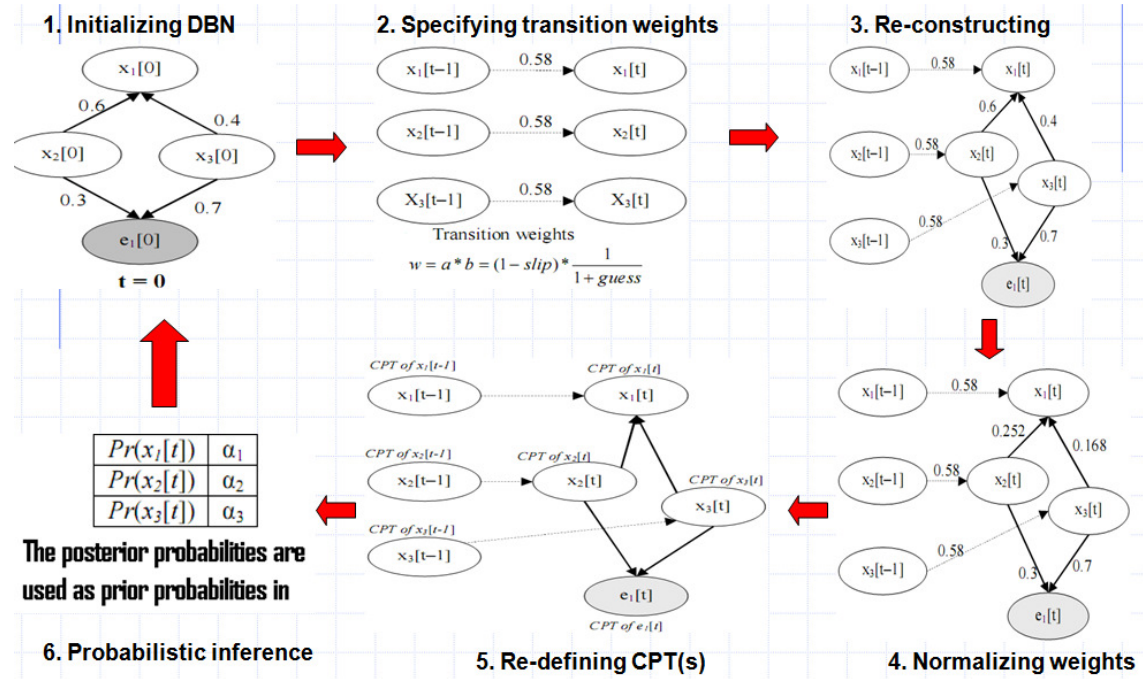
**Figure 3.6.** Dynamic Bayesian with time points  $t = 0, 1, 2$

The size of dynamic Bayesian network (DBN) gets huge after long time process, which causes a boom of combinations. This is the drawback of DBN. To overcome this drawback of, thesis proposes the new algorithm to construct DBN so that the size of DBN are kept intact (not changed) when the process continues for a long time, based on the Markov property “given the current time point  $t$ , the conditional probability of next time point  $t+1$  is only relevant to the current time point  $t$ ”. At every time point  $t$ , the algorithm only specifies  $G_{t-1}$ ,  $G_t$  and transition probability  $Pr(G_t | G_{t-1})$ . The algorithm has a lot of iterations; each iteration includes 6 accumulated steps:

1. Initializing DBN
2. Specifying transition weights
3. Re-constructing DBN
4. Normalizing weights of dependencies
5. Re-defining CPT (s)
6. Probabilistic inference

After each iteration, DBN is evolutionary and inference mechanism gets more precise, please see section III.4.1.2 in thesis with more details. At step 2, the concepts *slip* and *guess* are defined:

- *Slip* is the probability that learner does know a particular subject but there is solid evidence convincing that she/he doesn't understand it,  $Pr(not X_t | X_{t-1})$ .
- *Guess* is the probability that learner does not know a particular subject but there is solid evidence convincing that she/he does understand it,  $Pr(X_t | not X_{t-1})$ .
- *Transition weight* is derived from concepts *slip* and *guess*:  $(1 - slip) \frac{1}{1 + guess}$ . Please see section III.4.1.2 in thesis for proof of this formula.



**Figure 3.7.** The new algorithm to construct dynamic Bayesian network

In general, the knowledge sub-model  $K$  is the combination of overlay model and Bayesian network. The structure of  $K$  model is designed by specialist according to overlay model and its implementation conforms overlay model (see section III.1.2 in thesis, [De Bra, Calvi 1998] and [De Bra, Smits, Stash 2006]) but it applies Bayesian network into inference mechanism. Moreover, thesis proposes three methods to improve  $K$  model: learning parameters via EM algorithm, learning parameters via MLE algorithm, learning structure and modeling  $K$  model chronologically by dynamic Bayesian network.

### 3.2. Learning style sub-model (LS)

Learning styles [Stash, Cristea, De Bra 2005] are defined as the composite of characteristic cognitive, affective and psychological factors that serve as relatively stable indicators of how a learner perceives, interacts with and responds to the learning environment. There are many psychological theories relevant to learning style (LS) such as Dunn and Dunn, Witkin, Riding, Myers-Briggs, Kolb, Honey – Mumford, Felder – Silverman. Thesis choose hidden Markov model (HMM) [Dugad, Desai 1996] to represent the psychological theory Honey – Mumford and Felder – Silverman. According to Honey – Mumford and Felder – Silverman, learning styles are classified into dual pairs as following:

- Verbal / Visual: verbal person prefer to perceive materials as text and visual person prefer to perceive materials as pictures.
- Activist / Reflector: activists are open-minded and comprehend new information by doing something with it. Reflectors prefer to think about new information first before acting on it.
- Theorist / Pragmatist: Theorists think things through in logical steps, assimilate different facts into coherent theory. Pragmatists have practical mind, prefer to try and test techniques relevant to problems.

Hidden Markov model (HMM) has inference mechanism consisting of 5 components  $\langle S, \Pi, A, \Theta, B \rangle$ :

- A finite set of state  $S = \{s_1, s_2, \dots, s_n\}$ . A stochastic process  $P$  is a set of states  $P = \{x_1, x_2, \dots, x_k\}$  với  $x_i \in S$ . Process  $P$  must meet fully the Markov property, namely, given the current state  $x_k$  of process  $P$ , the conditional probability of next state  $x_{k+1}$  is only relevant to current state  $x_k$ . In other words,  $Pr(x_k | x_0, x_1, \dots, x_{k-1}) = Pr(x_k | x_{k-1})$ . In hidden Markov model,  $S$  is hidden set.
- Initial state distribution for every state,  $\Pi = (\pi_1, \pi_2, \dots, \pi_n)$  where  $\pi_i = Pr(s_i)$  is initial probability for state  $s_i$ .
- The transition probability matrix  $A = (a_{ij})$  where  $a_{ij}$  is the probability of transitioning from any given state  $s_i$  to some next state is  $s_j$ . We have  $\forall s_i \in S, \sum_{s_j \in S} a_{ij} = 1$ .
- A finite set of possible observations  $\Theta = \{\theta_1, \theta_2, \dots, \theta_m\}$ .
- Observation probability matrix  $B = (b_{ij})$  where  $b_{ij}$  is the probability that state  $s_i$  occurs given an observation  $\theta_j$ .

Given a stochastic process, hidden Markov model finds out hidden states from observations. Thesis applies hidden Markov model into discovering users' learning styles via observations about theirs learning process. Psychological theories Honey – Mumford and Felder – Silverman is constructed by hidden Markov model with 5 components  $\langle S, \Pi, A, \Theta, B \rangle$ . Suppose three pairs of learning styles are chosen such as *Verbal / Visual*, *Activist / Reflector*, *Theorist / Pragmatist*, we build up three models  $\Delta_1 = \langle S_1, \Theta_1, A_1, B_1, \Pi_1 \rangle$ ,  $\Delta_2 = \langle S_2, \Theta_2, A_2, B_2, \Pi_2 \rangle$  and  $\Delta_3 = \langle S_3, \Theta_3, A_3, B_3, \Pi_3 \rangle$ , followed by 5 steps:

1. Defining *states*: each state is identical to a leaning style. Thesis proposes a method to determine user's learning styles by finding out such states.  
 $S_1 = \{\text{verbal}, \text{visual}\}$ ,  $S_2 = \{\text{activist}, \text{reflector}\}$ ,  $S_3 = \{\text{theorist}, \text{pragmatist}\}$ .
2. Defining *initial state distributions*: Uniform probability distribution is used for each  $\Pi_i$ .  
 $\Pi_1 = \{0.5, 0.5\}$ ;  $Pr(\text{verbal}) = Pr(\text{visual}) = 0.5$   
 $\Pi_2 = \{0.5, 0.5\}$ ;  $Pr(\text{activist}) = Pr(\text{reflector}) = 0.5$   
 $\Pi_3 = \{0.5, 0.5\}$ ;  $Pr(\text{theorist}) = Pr(\text{pragmatist}) = 0.5$
3. Defining *transition probability matrices*: Suppose that learners tend to keep their styles; so the conditional probability of a current state on previous state is high if both current state and previous state have the same value and otherwise.

	Verbal	visual		activist	reflector		theorist	pragmatist
verbal	0.7	0.3	activist	0.7	0.3	theorist	0.7	0.3
visual	0.3	0.7	reflector	0.3	0.7	pragmatist	0.3	0.7

**Table 3.2.** Transition probability matrices  $A_1, A_2, A_3$

4. Defining *observations*. There is a relationship between learning object learned by users and their learning styles. Three attributes are assigned to each learning object such as lecture, example, etc.
  - Format attribute indicating the format of learning object has three values: *text*, *picture*, and *video*.
  - Type attribute telling the type of learning object has four values: *theory*, *example*, *exercise*, and *puzzle*.
  - Interactive attribute indicates the “interactive” level of learning object. The more interactive learning object is, the more learners interact together in their learning path. This attribute has three values corresponding to three levels: *low*, *medium*, *high*.

Whenever a student selects a learning object (LO), it raises observations depending on the attributes of learning object. We must account for the values of the attributes selected. For example, if a student selects a LO which has *format* attribute being *text*, *type* attribute being *theory*, *interactive* attribute being *low*, there are considerable observations: *text*, *theory*, *low* (interaction). So, it is possible to infer that she/he is a theorist.

The dimension *Verbal/Visual* is involved in format attribute. The dimensions *Activist/ Reflector* and *Theorist/ Pragmatist* relate to both *type* attribute and *interactive* attribute. So we have:

- $\Theta_1 = \{ \textit{text}, \textit{picture}, \textit{video} \}$
- $\Theta_2 = \{ \textit{theory}, \textit{example}, \textit{exercise}, \textit{puzzle}, \textit{low} \text{ (interaction)}, \textit{medium} \text{ (interaction)}, \textit{high} \text{ (interaction)} \}$
- $\Theta_3 = \{ \textit{theory}, \textit{example}, \textit{exercise}, \textit{puzzle}, \textit{low} \text{ (interaction)}, \textit{medium} \text{ (interaction)}, \textit{high} \text{ (interaction)} \}$

5. Defining *observation probability matrices*. Different observations (attributes of LO) effect on states (learning styles) in different degrees. Thesis specifies these degrees by weights. The larger the weight is, the higher the effectiveness of observation is. As example, for verbal person, the weights of observations: *text*, *picture*, *video* are in descending order because they prefer to *text* learning material. Otherwise, for visual person, the weights of observations: *text*, *picture*, *video* are in ascending. Weights are normalized into conditional probabilities and observation probability matrix is consists of these probabilities, for example,  $Pr(\textit{text} \mid \textit{verbal}) = 0.6$ ,  $Pr(\textit{picture} \mid \textit{verbal}) = 0.3$  and  $Pr(\textit{video} \mid \textit{verbal}) = 0.1$ .

Observation probability matrices  $B_1$ ,  $B_2$  and  $B_3$  whose rows are learning styles and columns are observations are specified as following:

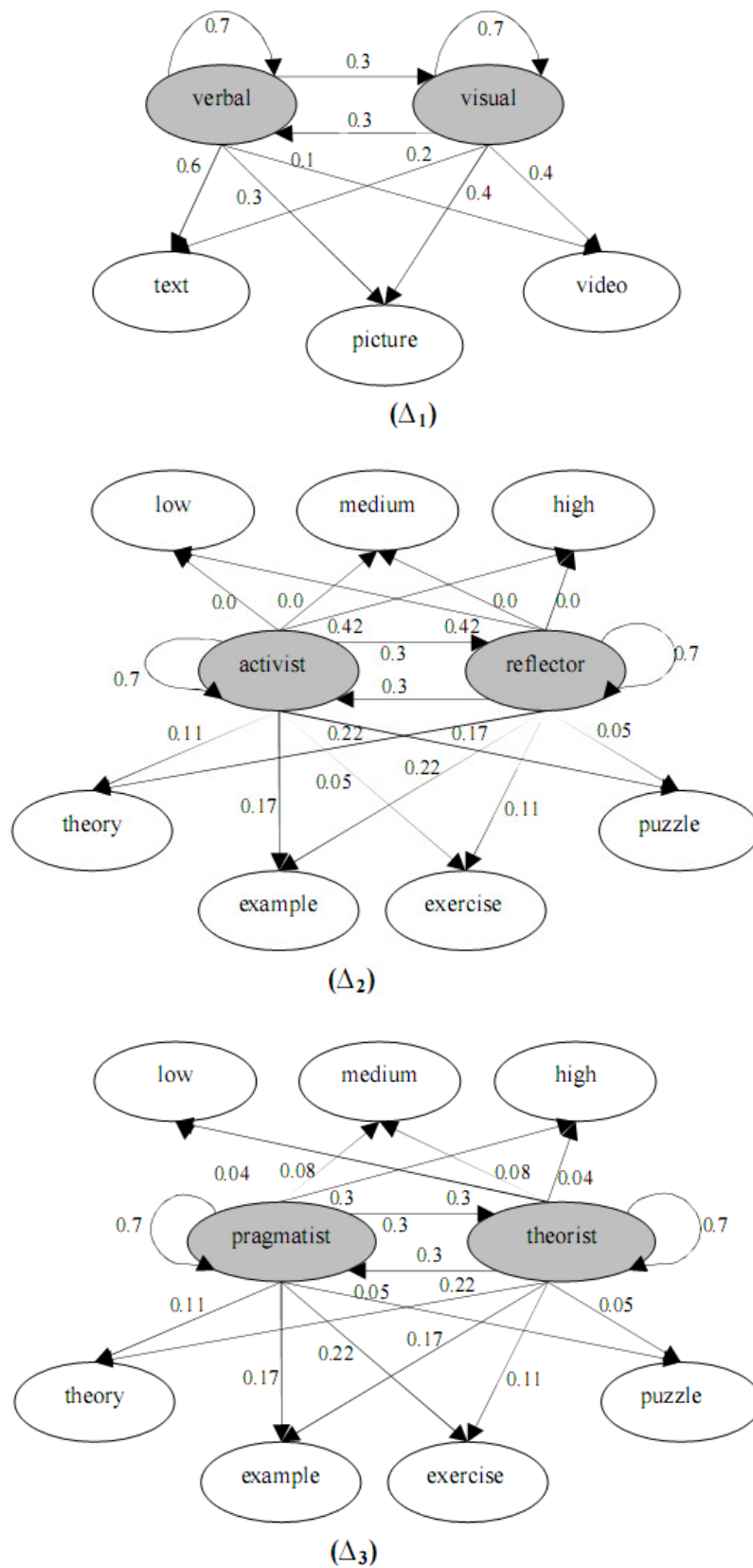
	<i>text</i>	<i>picture</i>	<i>video</i>
<i>verbal</i>	0.6	0.3	0.1
<i>visual</i>	0.2	0.4	0.4

	<i>theory</i>	<i>example</i>	<i>exercise</i>	<i>puzzle</i>	<i>low</i>	<i>medium</i>	<i>high</i>
<i>activist</i>	0.11	0.17	0.05	0.22	0	0	0.42
<i>reflector</i>	0.17	0.22	0.11	0.05	0.42	0	0

	<i>theory</i>	<i>example</i>	<i>exercise</i>	<i>puzzle</i>	<i>low</i>	<i>medium</i>	<i>high</i>
<i>pragmatist</i>	0.11	0.17	0.22	0.05	0.04	0.08	0.3
<i>theorist</i>	0.22	0.17	0.11	0.05	0.3	0.08	0.04

**Table 3.3.** Observation probability matrices  $B_1$ ,  $B_2$  and  $B_3$

After specifying 5 aforementioned steps, three hidden Markov models  $\Delta_1$ ,  $\Delta_2$ ,  $\Delta_3$  are built up as following:



**Figure 3.8.** Hidden Markov models for three type learning styles: *Verbal/Visual, Activist/Reflector, Theorist/Pragmatist*

After hidden Markov models are built up, Viterbi algorithm is applied into finding out users' learning styles from their learning material choices and their learning processes. Suppose learner chooses 3 learning objects  $LO_1$ ,  $LO_2$ , and  $LO_3$  which have following attributes:

	Format	Type	Interactive
$LO_1$	picture	theory	not assigned
$LO_2$	text	example	not assigned
$LO_3$	text	not assigned	low

**Table 3.4.** Learning objects selected

Observations sequences are extracted from selected learning objects.

Hidden Markov models	Observation sequences
$\Delta_1$ : Dạng thức <i>Verbal / Visual</i>	picture $\rightarrow$ text $\rightarrow$ text
$\Delta_2$ : Dạng thức <i>Activist / Reflector</i>	theory $\rightarrow$ example $\rightarrow$ low
$\Delta_3$ : Dạng thức <i>Theorist / Pragmatist</i>	theory $\rightarrow$ example $\rightarrow$ low

**Table 3.5.** Observation sequences

Viterbi algorithm is used to find out corresponding sequence of state transitions that is most suitable to have produced such sequence of observations.

Markov model	Sequence of observations	Sequence of state transition	Learning style (LS)
$\Delta_1$	picture $\rightarrow$ text $\rightarrow$ text	visual $\rightarrow$ verbal	verbal
$\Delta_2$	theory $\rightarrow$ example $\rightarrow$ low	reflector $\rightarrow$ reflector $\rightarrow$ reflector	reflector
$\Delta_3$	theory $\rightarrow$ example $\rightarrow$ low	theorist $\rightarrow$ theorist $\rightarrow$ theorist	theorist

It is easy to deduce that this student is a verbal, reflective and theoretical person.

In general, thesis proposes five steps to realize and apply HMM into two learning style models: Honey-Mumford and Felder-Silverman, in which styles are considered states and user's selected learning objects are tracked as observations. From observations, learners' learning styles are discovered based on hidden Markov model. The algorithm is very flexible, which is extended into other learning style models different from Honey – Mumford and Felder – Silverman, in which, each learning style modal (pair / dimension) is represented as an hidden Markov model and the number of model is not limited and the number of learning object attributes is also optional.

### 3.3. Learning history sub-model (LH)

Learning history (LH) sub-model is at the bottom of Triangular Learner Model (TLM) because it is the most important one which is used to construct knowledge sub-model and learning style sub-model. Learning history (LH) is defined as a transcript of all learners' actions such as learning materials access, duration of computer use, doing exercise, taking an examination, doing test, communicating with teachers or classmates, etc. This sub-model has four main functions:

1. Providing necessary information for two remaining sub-models: learning style sub-model and knowledge sub-model described in previous sections so that they perform inference tasks. For example, knowledge sub-model needs learning evidences like learner's results of test, frequency of accessing lectures and etc so as to assess learner's mastery of concrete knowledge item or concept.
2. Supporting learning concept recommendation.
3. Mining learners' educational data in order to discover other learners' characteristics such as interests, background, goals, etc.
4. Supporting collaborative learning through constructing learner groups.

Mining engine (ME) structures and manages *LH* sub-model by data mining techniques [Han, Kamber 2006]. This summarization focuses on function 2, 3, 4. By function 1, *LH* sub-model is often archived in XML file format.

### Learning concept recommendation based on sequential pattern mining

Learning activities such as doing exercises, accessing learning materials, etc are represented as sequential database (or learning sequences) and sequential patterns are mined from this database. Sequential pattern represents the learning path that learners should follow in their learning process. For example, if learner is studying concept  $A$  and the sequential pattern is  $B \rightarrow C \rightarrow A \rightarrow E$ , adaptive learning system will recommend her/him to study from concept  $B$  to  $C$ ,  $A$  and  $E$  in successive. Please see section III.3.1.1 in thesis for more details in mining sequential pattern [Agrawal (Rakesh), Srikant 1995]. Thesis proposes a special technique that breaks mined sequential pattern so as to perform recommendation task.

Suppose there are domain concepts (subjects) in Java course: **data type**, **package**, **class & OOP**, **selection structure**, **virtual machine**, **loop structure**, **control structure**, and **interface** which in turn denoted as  $d, p, o, s, v, l, c, f$ . Study activities that students access learning material relating such domain concepts in sessions are represented as the learning sequential database as following:

Student	Session	Concept accessed	ID	Learning sequences	Length
1	Aug 5 10:20:01	o	1	$\langle of \rangle$	2
1	Aug 5 10:26:12	f			
2	Aug 6 08:20:01	d, p	2	$\langle (dp)o(slc) \rangle$	6
2	Aug 6 14:15:01	o			
2	Aug 6 15:00:00	s, l, c			
3	Aug 7 12:30:00	o, v, c	3	$\langle (ovc) \rangle$	3
4	Aug 8 07:14:20	o	4	$\langle o(sc)f \rangle$	4
4	Aug 8 07:40:25	s, c			
4	Aug 8 10:17:20	f			
5	Aug 8 10:26:15	f	5	$\langle f \rangle$	1

**Table 3.7.** Learning sequential database

Suppose sequential pattern  $\langle osc(sc) \rangle$  is mined from learning sequential database, we have learning path: “*class & OOP*” – “*selection structure*” – “*control structure*” – “*selection structure, control structure*”. If a student is studying the concept “*class & OOP*”, the adaptive learning system should recommend which next concepts in above patterns. So the patterns should be broken into association rules with their confidence. For example, breaking above pattern  $\langle osc(sc) \rangle$  follows 4 steps:

1. Breaking entire  $\langle osc(sc) \rangle$  into large itemsets such as  $o, s, c, (sc)$  and determining all possible large 2-sequences which are 2-arrangement of all large itemsets following the criterion: order of 2-sequences must comply with the order of sequential pattern. There are six large 2-sequences:  $\langle os \rangle, \langle oc \rangle, \langle o(sc) \rangle, \langle sc \rangle, \langle s(sc) \rangle, \langle c(sc) \rangle$ .
2. Thus, we have six rules derived from these large 2-sequences in form: “*left-hand itemset*  $\rightarrow$  *right-hand itemset*”, for example, rule “ $s \rightarrow c$ ” derived from 2-sequence  $\langle sc \rangle$ .
3. Computing the confidences of such rules and sorting them according to these measures. The confidence of a rule is the ratio of the support of 2-sequences and the support of left-hand large itemset,  $confidence(x \rightarrow y) = support(\langle xy \rangle) / support(\langle x \rangle)$ . The rules whose confidences are less than a pre-defined threshold are removed in order to ensure that remains are strong rules. We called these rules as *sequential rules* and these confidences as *sequential confidences*.

sequential rules	sequential confidences
$o \rightarrow s$	40%
$o \rightarrow c$	40%
$o \rightarrow sc$	40%
$s \rightarrow c$	0%
$s \rightarrow sc$	0%
$c \rightarrow sc$	0%



**Table 3.8.** Sequential rules

4. If student choose the concept (itemset)  $x$ , system will find the rules whose *left-hand itemset* must contain  $x$  and whose confidence is highest. After that, concepts in the *right-hand itemsets* (concepts) of this rule are recommended to learner. For example, if learner is studying concept “*class & OOP*”, system find out two rules  $o \rightarrow s$  and  $o \rightarrow c$  and both of them contain concept  $o$  (*class & OOP*) and their confidences are highest (40%); so concepts  $s$  (*selection structure*) and  $c$  (*control structure*) are recommended to learners.

This methodology is similar to mining association rules but it achieves high performance and precise prediction since it proposes learner learning path derived from result of sequential pattern mining process.

### Discovering user interests by document classification

Interest is personal trait not modeled in standard Triangular Learner Model (TLM) but it is described in extended TLM because of its importance in adaptive process. Learning history sub-model is responsible for discovering user interests by mining techniques. Thesis only proposes a mining technique so-called document classification but it assures that there will have many other mining techniques and so thesis gives an open research. The proposed technique is based on two points of view:

- The series of user access in his/her history are modeled as documents. So user is referred indirectly to as “document”.
- User interests are classes such documents are belong to.

Our approach includes four following steps:

1. Documents in training corpus are represented according to vector model. Each element of vector is product of term frequency and inverse document frequency. However the inverse document frequency can be removed from each element for convenience.
2. Classifying training corpus by applying decision tree or support vector machine or neural network. Regardless of using any classification technique, the classification rules or classifiers must be extracted. Please see section III.3.2 for classification techniques in detailed.
3. Mining user’s access history to find maximum frequent itemsets. Each itemset is considered an interesting document and its member items are considered as terms. Such interesting documents are modeled as vectors.
4. Applying classifiers (see step 2) into these interesting documents (see step 3) order to choose which classes are most suitable to these interesting documents. Such classes are user interests.

Following is the example of discovering user interests by document classification. Given a set of classes  $C = \{\text{computer science, math}\}$ , a set of terms  $T = \{\text{computer, programming language, algorithm, derivative}\}$  and the corpus  $D = \{\text{doc1.txt, doc2.txt, doc3.txt, doc4.txt, doc5.txt}\}$ . Term frequencies of documents are shown in following table:

	<i>computer</i>	<i>programming language</i>	<i>algorithm</i>	<i>derivative</i>	<i>class</i>
<i>doc1.txt</i>	5	3	1	1	computer
<i>doc2.txt</i>	5	5	40	5	math
<i>doc3.txt</i>	20	5	20	55	math
<i>doc4.txt</i>	20	55	5	20	computer
<i>doc5.txt</i>	15	15	4	0.3	math
<i>doc6.txt</i>	35	10	45	10	computer

**Table 3.9.** Term frequencies of documents

	<i>computer</i>	<i>programming language</i>	<i>algorithm</i>	<i>derivative</i>	<i>class</i>
<i>doc1.txt</i>	0.5	0.3	0.1	0.1	computer
<i>doc2.txt</i>	0.05	0.05	0.4	0.5	math
<i>doc3.txt</i>	0.2	0.05	0.2	0.55	math
<i>doc4.txt</i>	0.2	0.55	0.05	0.2	computer
<i>doc5.txt</i>	0.15	0.15	0.4	0.3	math
<i>doc6.txt</i>	0.35	0.1	0.45	0.1	computer

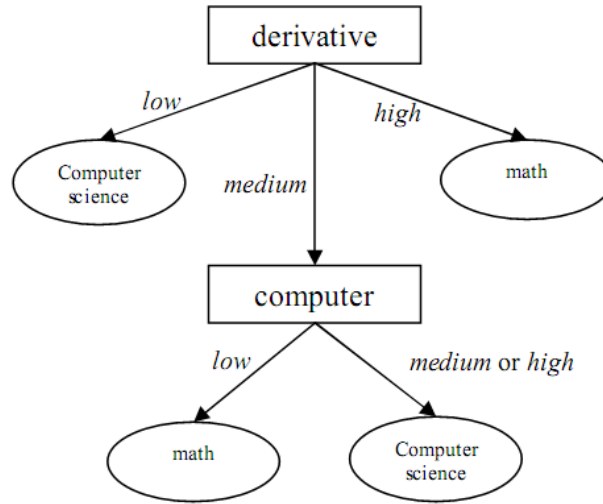
**Table 3.10.** Normalized term frequencies

Suppose document classifier is decision tree, all term frequencies are changed from real number into nominal value:  $0 \leq \text{frequency} < 0.2$ : low;  $0.2 \leq \text{frequency} < 0.5$ : medium;  $0.5 \leq \text{frequency}$ : high

	<i>computer</i>	<i>programming language</i>	<i>algorithm</i>	<i>derivative</i>	<b>class</b>
<i>doc1.txt</i>	high	medium	low	low	computer
<i>doc2.txt</i>	low	low	medium	high	Math
<i>doc3.txt</i>	medium	low	medium	high	Math
<i>doc4.txt</i>	medium	high	low	medium	computer
<i>doc5.txt</i>	low	low	medium	medium	Math
<i>doc6.txt</i>	medium	low	medium	low	computer

**Table 3.11.** Nominal term frequencies

The decision tree is generated from nominal term frequencies as following



**Figure 3.9.** Decision tree

We can extract classification rules from this decision tree:

<i>Rule 1</i>	If frequency of term “derivative” is <i>low</i> then document belongs to class <i>computer science</i>
<i>Rule 2</i>	If frequency of term “derivative” is <i>medium</i> and frequency of term “computer” is <i>medium</i> or <i>high</i> then document belongs to class <i>computer science</i>
<i>Rule 3</i>	If frequency of term “derivative” is <i>medium</i> and frequency of term “computer” is <i>low</i> then document belongs to class <i>math</i> .
<i>Rule 4</i>	If frequency of term “derivative” is <i>high</i> then document belongs to class <i>math</i>

**Table 3.12.** Classification rules deriving from decision tree induction

Suppose in some library or website, user does her/his search for her/his interesting books, documents, etc. Given there is a set of key words or terms {computer, programming language, algorithm, derivative} that user U often looking for, her/his searching history is shown in following table:

<i>Date</i>	<i>Keywords (terms) searched</i>
Aug 28 10:20:01	computer, programming language, algorithm, derivative
Aug 28 13:00:00	computer, programming language, derivative, algorithm
Aug 29 8:15:01	Computer
Aug 30 8:15:06	Computer

**Table 3.13.** User’s searching history

The maximum frequent itemsets are mined from searching history when we consider searching history as training corpus (dataset).

$N_o$	Maximum frequent itemset
1	(computer = 4, programming language = 2, algorithm = 2, derivative = 2)

**Table 3.14.** The maximum frequent itemset in vector model whose components are term frequencies

The maximum frequent itemset is considered as “interesting document” and the class of this document is user’s interest. “Interesting document” is normalized as following.

$N_o$	Interesting document
1	(computer = 0.4, programming language = 0.2, algorithm = 0.2, derivative = 0.2)

**Table 3.15.** Interesting document vector is normalized

$N_o$	Interesting document
1	(computer = <i>medium</i> , programming language = <i>medium</i> , algorithm = <i>medium</i> , derivative = <i>medium</i> )

**Table 3.16.** Nominal interesting document vector

Applying classification rule 2 in table 3.10, the interesting document belongs to class computer science because the frequency of both terms “*derivative*” and “*computer*” are *medium*. So we can state that user has only one interest: *computer science*.

In general this methodology is based on the point of view that learner is modeled indirectly as a “document” and the class of this document is her/his interest. The method is the combination of classification technique and information retrieval technique.

### Constructing user groups or user communities

User model in thesis aims to individualization, which is appropriate to personalized adaptive system; in addition, there is a demand to provide adaptation to a group or community of users. Consequently, all users in the same group will profit from the same learning materials, teaching methods. Of course, individual or community (group) adaptation has particular features and demands but the preeminent adaptive system should support both of them. Group adaptation has more advantages than individual adaptation in some situations:

- Common features in a group which are the common information of all members in such group are relatively stable, so it is easy for adaptive systems to perform accurately adaptive tasks.
- If a new user logins system, she/he will be classified into a group and it is easy to recognize common features. Therefore, adaptive system will give her/his adaptation immediately and there is no complicated inference mechanism.
- In the collaborative learning, users need to learn or discuss together. It is very useful if the collaborative learning is restricted in a group of similar users. Therefore, it is convenient for users that have common characteristics (knowledge, goal, interest, etc) to learn together because they do not come up against an obstacle when interacting together.

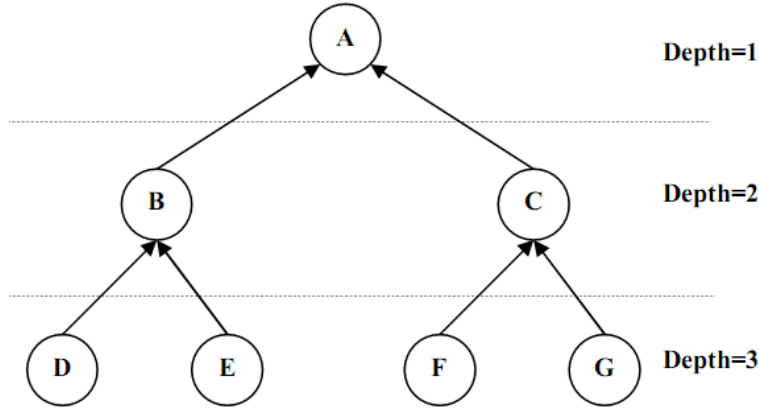
Because it is necessary to specify user groups, thesis proposes clustering technique to determine such groups, especially clustering for overlay model and Bayesian network. Please see sections III.3.3.1 and III.3.3.3 for clustering techniques along with similarity and dissimilarity measures. This summarization focuses on clustering techniques for overlay model and Bayesian network.

There are many applicable clustering techniques [Han, Kamber 2006] but there is a problem: “how to compute such measure in case that user model is an overlay model which is in form of domain graph”. Thesis proposes special technique based on graph depth.

Suppose there are two overlay models  $U_1=G_1=<V_1,E_1>$  và  $U_2=G_2=<V_2,E_2>$  where  $V_1, V_2$  are sets of nodes and  $E_1, E_2$  are sets of arcs. Graphs  $G_1$  and  $G_2$  are in form of tree in which each directed arc represents the prerequisite relationship of two nodes. If there is an arc from node A to node B, user must master over A before learning B.

$V_i = (v_{i1}, v_{i2}, \dots, v_{in})$  is also considered as a vector whose elements are integer numbers representing user’s masteries of knowledge items.

Let  $depth(v_{ij})$  is the depth level of node j of graph model  $G_i$ . Note that the depth level of root node is 1.



**Figure 3.10.** Graph and depth

The dissimilarity (or distance) of two graph models  $G_1$  and  $G_2$  is defined as below

$$dissim(G_1, G_2) = distance(G_1, G_2) = \sum_{j=1}^n \left| \frac{v_{1j} - v_{2j}}{depth(v_{1j})} \right|$$

The meaning of this formulation is “the high level concept (node) is the aggregation of low level (basic) concepts”. For example, three graphs  $G_1$ ,  $G_2$ ,  $G_3$  have the same structure shown in figure 3.10, the values of their nodes are shown in following table:

	A	B	C	D	E	F	G
$G_1$	2	1	1	0	3	2	1
$G_2$	1	1	0	1	4	5	4
$G_3$	2	1	1	1	4	5	4

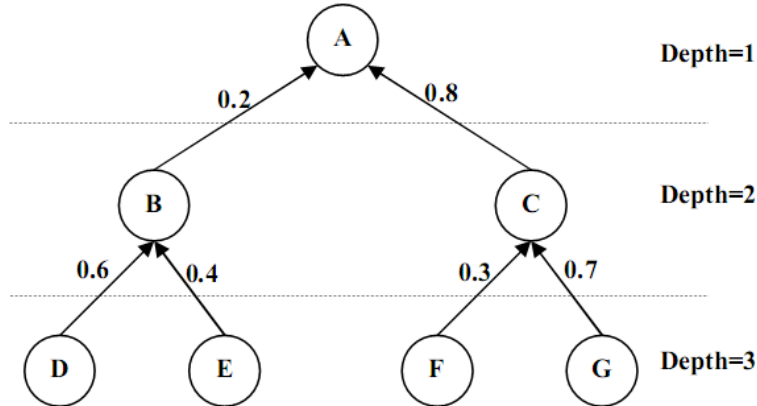
**Table 3.17.** Values of graph nodes ( $G_1$ ,  $G_2$ ,  $G_3$ )

The dissimilarity (or distance) between  $G_3$  and  $G_1$ ,  $G_2$ , respectively are computed as below:

$$dissim(G_1, G_3) = \left| \frac{2-2}{1} \right| + \left| \frac{1-1}{2} \right| + \left| \frac{1-1}{2} \right| + \left| \frac{0-1}{3} \right| + \left| \frac{3-4}{3} \right| + \left| \frac{2-5}{3} \right| + \left| \frac{1-4}{3} \right| = 2.66$$

$$dissim(G_2, G_3) = \left| \frac{1-2}{1} \right| + \left| \frac{1-1}{2} \right| + \left| \frac{0-1}{2} \right| + \left| \frac{1-1}{3} \right| + \left| \frac{4-4}{3} \right| + \left| \frac{5-5}{3} \right| + \left| \frac{4-4}{3} \right| = 1.5$$

So  $G_2$  is more similar to  $G_3$  than  $G_1$  is. In case that arcs in graph are weighted, graph model is shown in following figure



**Figure 3.11.** Graph model and weighted arcs

Let  $weight(v_{ij})$  be the weight of arc from node  $j$  (of graph model  $i$ ) to its parent. I consider that  $weight(v_{ij})$  is the weight at node  $v_{ij}$ , the dissimilarity (or distance) of two graph models  $G_1$  and  $G_2$  is re-defined as below:

$$dissim(G_1, G_2) = distance(G_1, G_2) = \sum_{j=1}^n \left| \frac{v_{1j} - v_{2j}}{depth(v_{1j})} * weight(v_{1j}) \right|$$

Note that the weight at root node equals 1. For example, the dissimilarity (or distance) between  $G_3$  and  $G_1, G_2$ , respectively are re-computed as below:

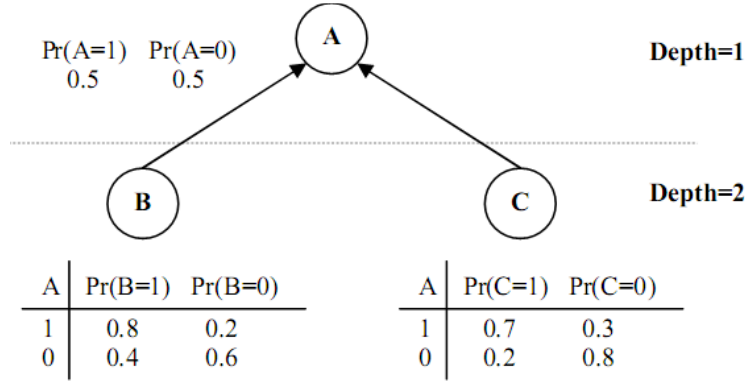
$$dissim(G_1, G_3) =$$

$$\left| \frac{2-2}{1} \right| + \left| \frac{1-1}{2} * 0.2 \right| + \left| \frac{1-1}{2} * 0.8 \right| + \left| \frac{0-1}{3} * 0.6 \right| + \left| \frac{3-4}{3} * 0.4 \right| + \left| \frac{2-5}{3} * 0.3 \right| + \left| \frac{1-4}{3} * 0.7 \right| = 1.33$$

$$dissim(G_2, G_3) =$$

$$\left| \frac{1-2}{1} \right| + \left| \frac{1-1}{2} * 0.2 \right| + \left| \frac{0-1}{2} * 0.8 \right| + \left| \frac{1-1}{3} * 0.6 \right| + \left| \frac{4-4}{3} * 0.4 \right| + \left| \frac{5-5}{3} * 0.3 \right| + \left| \frac{4-4}{3} * 0.7 \right| = 1.4$$

Now  $G_1$  is more similar to  $G_3$  than  $G_2$  is. In case that graph model is Bayesian network, the dissimilarity-calculation formula is more complicated. Bayesian network is considered as graph and the graph depth has the same definition but the weight of arc is replaced by conditional probability table (CPT). So the value at each node is replaced by marginal posterior probability.



**Figure 3.12.** Bayesian network, CPT and depth

Let  $Pr(v_{ij})$  be the marginal probability of node  $j$  in network  $G_i$ , the dissimilarity (or distance) between two Bayesian network  $G_1$  and  $G_2$  is defined as below:

$$dissim(G_1, G_2) = distance(G_1, G_2) = \sum_{j=1}^n \left| \frac{\Pr(v_{1j}) - \Pr(v_{2j})}{depth(v_{1j})} \right|$$

Please see section III.1.1.1 in thesis for inference mechanism in Bayesian network and marginal posterior probability.

## 4. Evaluating adaptive learning model

As aforementioned, user modeling system is the heart of adaptive learning system. There are a lot of theories and practical methods including the approach proposed in this thesis to build up adaptive system and user modeling system. Each method has strong points and drawbacks and so it is very useful to evaluate these methods in order to determine which model is appropriate to which situation because each method tailors to concrete conditions and contexts. For example, studying via internet website is very different from studying at a course with support of network. This research focuses on how to evaluate adaptive learning system with regard to user modeling system in e-learning or distance learning context when there is no separation between adaptive learning system and user modeling system. We can consider the corporation between adaptive system and user modeling system as an integrated model so-called adaptive learning model. Thus, this research has two goals:

- Firstly, research proposes criterions to evaluate adaptive learning model.
- Secondly, research gives some scenarios as an example that applies criterions above into performing evaluation task in concrete situations.

This thesis proposes three criterions of evaluation:

- Criterion  $\alpha$  so-called system criterion tells us how adaptive learning system works with/without user modeling system. For example, when modeling server applies Bayesian network into build up learner model, criterion  $\alpha$  measures the performance of adaptive system with or without the support of Bayesian network. In general, this criterion answers two following questions:
  - o How adaptation is performed in adaptive system with/without the support of modeling server.
  - o Whether the whole user knowledge is computed more accurately with the support of user model, for example Bayesian network.
- Criterion  $\beta$  so-called academic criterion tells us how well modeling server help users to study. This criterion surveys users' study result. The higher criterion  $\beta$  is, the better study result is.
- Criterion  $\gamma$  so-called adaptation criterion or satisfaction criterion measures the quality of adaptation function of learning system with the support of modeling server. After every student gives feedbacks or comments on adaptive system, these feedbacks are collected and analyzed; hence, criterion  $\gamma$  is calculated based on these feedbacks in order to estimate level of students' satisfaction from adaptive system. The higher criterion  $\gamma$  is, the better quality of adaptation is.

There are two ways to calculate criterion  $\alpha$  such as using hypothesis testing and using regression model. By using hypothesis testing, students are classified into two groups with and without the support of user modeling system (for example, Zebra system) and the knowledge variance of each group is calculated. The null hypothesis is stated that two variances are equal. Rejecting null hypothesis is equivalent to criterion  $\alpha$  getting Boolean value *true*, indicating the preeminence of user modeling system. Please see section IV.2.1.1 in thesis for more details about calculating criterion  $\alpha$  by using hypothesis testing. By using regression technique, if an inference method is good, its predictive value, namely the whole knowledge user achieves, and all partial knowledge items user study at every stage on learning path will satisfy well a function or equation. The process to calculate  $\alpha$  according to regression technique has four steps:

1. The regression functions relevant to user knowledge are determined for estimated groups with and without the support of user modeling system.
2. The estimated user knowledges are calculated based on these regression functions.
3. The prediction errors are defined as the deviation between estimated knowledges and real knowledges.
4. The measure  $\alpha$  is simple inverse of square sum of all prediction errors.

Please see section IV.2.1.1 in thesis for more details about calculating criterion  $\alpha$  by using regression technique. Given two groups with and without the support of user modeling system (for example, Zebra system), the measure  $\beta$  for each group is computed as accumulative probability of assumption user in such group has mastered over course. Note that the accumulative probability conforms normal distribution and is calculated based on the sample variance and sample mean of knowledge vector of each group.

Suppose a questionnaire is built up by expert and it is composed of  $n$  questions. Users in each group rate on each question where rating value may be binary satisfied and unsatisfied. By the simplest way, criterion  $\gamma$  is defined as the ratio of the number of satisfied users to the whole number of users. In enhance method, the rating values range in an interval, for example  $[0...5]$ , where value 0 and 5 indicates least and most satisfied and so users' ratings composes a rating matrix. There are three steps to compress rating matrix and to calculate  $\gamma$ :

1. Firstly, rating matrix is "shrunk" by projecting it onto its eigenvectors.
2. Secondly, the mean  $\mu_i$  of each column is calculated.
3. Finally, the mean vector of this matrix is composed of all column mean  $\mu_i$  and the criterion  $\gamma$  is the module of such mean vector.

In general, the higher criterion  $\gamma$  is, the better quality of adaptation is. Please see section IV.2.1.1 in thesis for more details about calculating criterion  $\gamma$ .

Evaluation scenario is the example for demonstrating how to calculate and apply aforementioned criterions into evaluating the quality of adaptive learning model. E-learning cannot replace face-to-face teaching and it should exist parallel and support traditional education. Thus, this scenario makes the comparison between face-to-face learning manner and distance learning manner. This evaluation scenario is divided into three main acts in which students and teacher play the roles of actors.

1. *Study act*: Teacher teaches and students learn in both face-to-face manner and e-learning manner via website. Suppose students are classified into three groups  $A$ ,  $B$  and  $C$ . Group  $A$  and  $B$  represent face-to-face manner and e-learning manner via website, respectively. Especially, group  $C$  represents e-learning manner with support of user model, namely Bayesian network.
2. *Feedback act*: Students give feedbacks to teacher and teacher collects and analyzes them.
3. *Evaluation act* is done by teacher; thus, criterions  $\alpha$ ,  $\beta$  and  $\gamma$  are calculated according to data collected from two above acts. The quality of adaptive learning in groups  $A$ ,  $B$  and  $C$  are determined based on such criterions.

Study act has 5 scenes:

1. Teacher builds up school's curriculums and set up adaptive e-learning website with/without the support of user modeling system.
2. Teacher teaches and students in group  $A$ ,  $B$  and  $C$  learn by face-to-face manner.
3. Students in group  $B$  and  $C$  go on website and study by themselves. Teacher monitors them and put up important notice.
4. Students in group  $A$ ,  $B$  and  $C$  do tests and exercises via website.
5. Teacher evaluates students based on their test results.

Teacher's role in study act:

- Teaching face-to-face in traditional manner.
- Building up knowledge domain and creating web resources for this domain such as defining html lesions, tests, exercises, etc.
- Creating user model, for example, creating Bayesian network and its weights for knowledge domain.
- Setting up user modeling system and e-learning adaptive website.
- Monitoring students' learning process.
- Sending test results and school report to students.

Students' role in study act:

- Students in group  $A$ ,  $B$  and  $C$  go to class to study in face-to-face manner.
- Students in group  $B$  and  $C$  learn themselves on adaptive learning web sites. Note that website / learning materials are adapted to each student based on their knowledge and characteristics.
- Students in group  $A$ ,  $B$  and  $C$  do tests / exercise via website.

Feedback act has 3 scenes:

1. Teacher creates the questionnaire to survey students' feeling about both adaptive learning website and curriculum such as very satisfied, satisfied and not satisfied.
2. Students answer or rate on such questions online.
3. Teacher collects students' feedbacks and analyzes them.

Evaluation act has 2 scenes:

1. Teacher calculates three criterions based on students' feedback and test results.
2. Teacher makes the decision about the quality of face-to-face teaching manner and e-learning manner with/without support of user modeling system.

In general, there are three criterions such as system criterion  $\alpha$ , academy criterion  $\beta$  and adaptation criterion  $\gamma$ . That two of three criterions, concretely  $\alpha$  and  $\beta$ , assessing user knowledge implicates that evaluation of adaptive learning model focuses on the effect of education which is ability to help student to improve their knowledge although adaptation and personalization is significant topic in adaptive learning. You can recognize that the education never goes beyond the main goal that increases amount of human knowledge. Evaluation scenario, an example for demonstrating how to determine these criterions, indicates that study is lifelong process for everyone and so, classes and courses are short movies in this lifelong process. Both students and teachers are actors and their roles can mutually interchange, for example, teaching is the best way to learn and student is the best teacher of teacher.

## 5. Conclusion and future trend

In general, thesis aims to build up a user modeling system in learning context. Firstly, learner model so-called Triangular Learner Model (TLM) consisting of 3 sub-models such as knowledge (K), learning style (LS) and learning history (LH) which are associated together so as to form a triangular.

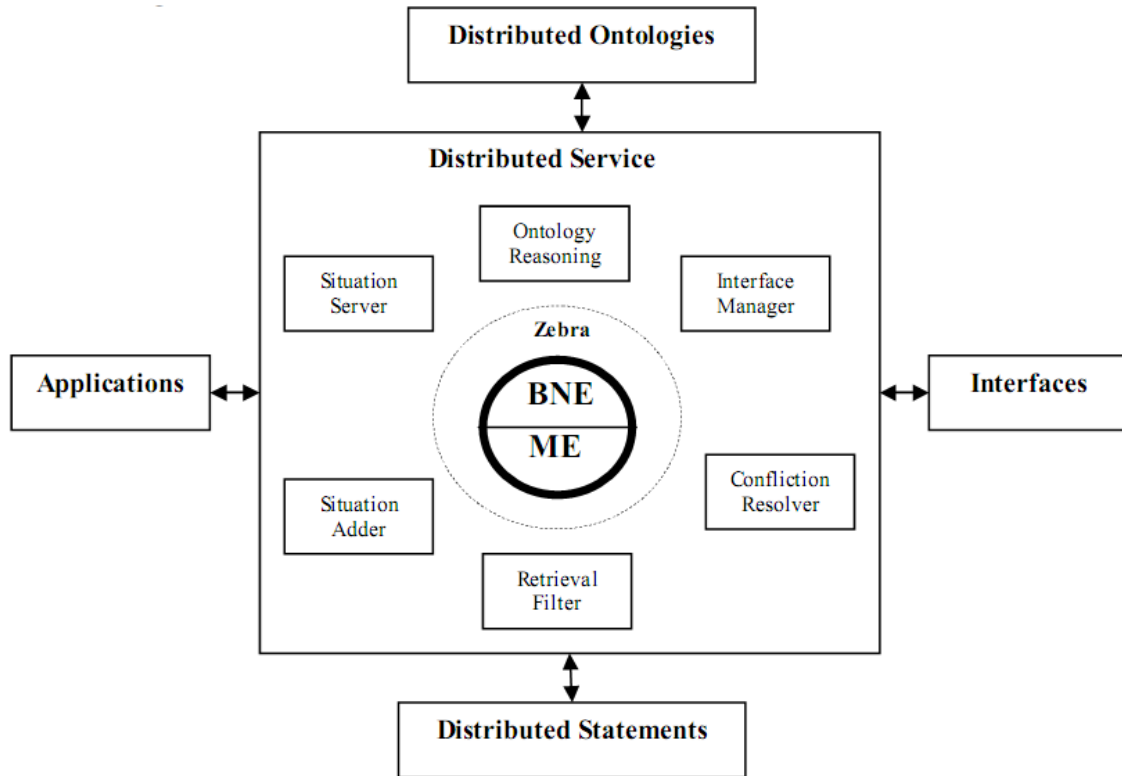
- Knowledge sub-model (*K*) is the combination of overlay model and Bayesian network together with weight specification. *K* sub-model is evolved via techniques such as parameter learning via EM algorithm, parameter learning via MLE algorithm, structure learning via dynamic Bayesian network.
- Learning style sub-model (*LS*): applying hidden Markov model into psychological theories so as to discover user's learning styles.
- Learning history sub-model (*LH*) is the most important one which archives learners' study activities and has XML file format. *LH* sub-model is used to construct *K* sub-model and *LS* sub-model and to discover extended features such as user interests, user group, etc.

User modeling system Zebra manages Triangular Learner Model (TLM) and performs inference mechanism by two engines:

- Mining engine ME manages whole TLM but focuses on learning history sub-model (*LH*) and uses mainly data mining techniques.
- Belief network engine BNE manages and performs inference mechanism on knowledge sub-model (*K*) and learning style sub-model (*LS*) by applying Bayesian network and Markov model.

Zebra system in the future will support two additional functions:

- Computerized Adaptive Test – CAT [Baker 2001]: each learner receives different tests and knowledge evaluation model is improved more and more with the most optimal terminated condition so as to assess learner's knowledge as accurate as possible.
- Ubiquitous environment [Heckmann 2005] support: the user modeling system interacts with users at anywhere and is totally transparent; it means that users need not make sense information technology infrastructure and the architecture of user modeling system and so, users only take advantages of profits from ubiquitous service. Zebra user modeling system is integrated into ubiquitous environment according to assumed architecture as following:

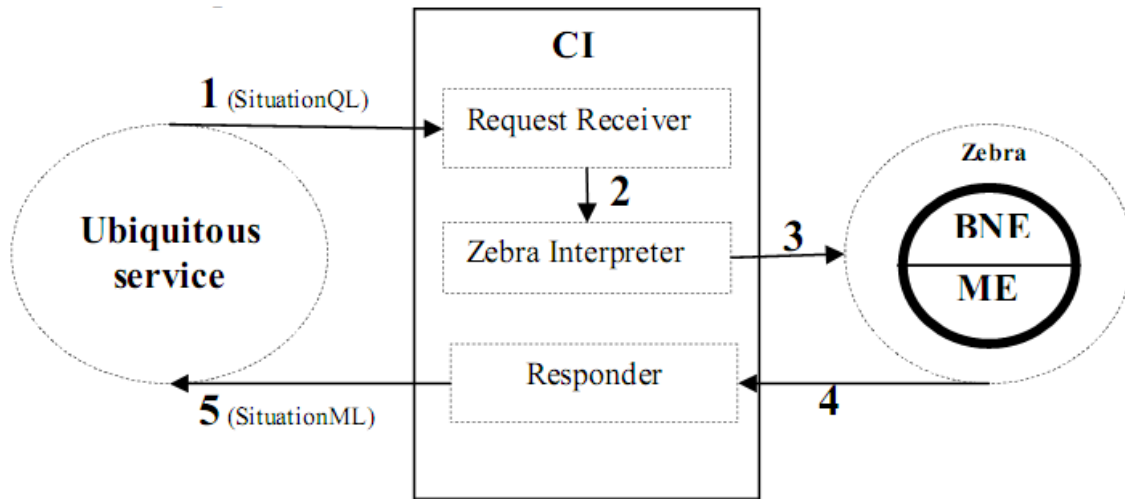


**Figure 4.1.** Incorporating Zebra into ubiquitous service  
(Please see section IV.2.5 in thesis for more details)



Therefore, the ubiquitous service will interact with Zebra by special CI by request-response protocol including four following steps:

1. Ubiquitous service sends one request statement in form of *SituationQL* to *CI*.
2. *CI* interprets this statement into the data structure which inference engines of Zebra such as mining engine & belief network engine are aware of.
3. The request in form of data structure that Zebra knows is sent to Zebra.
4. Inference engines perform some concrete deduction tasks in order to take out some new assumptions, information, personal recommendations about user. This output is sent back *CI*.
5. *CI* interprets such output into a XML-file in form of *SituationML* which is readily understandable for ubiquitous service and sends it to ubiquitous service.



**Figure 4.2.** Interaction between Zebra and ubiquitous service  
(Please see section IV.2.5 in thesis)

## Reference

1. [Agrawal (Rakesh), Srikant 1995]. R. Agrawal and R. Srikant. Mining sequential patterns. In Proc. 1995 Int. Conf. Data Engineering (ICDE95), pp.314, Taipei, Taiwan, Mar.1995.
2. [Baker 2001]. Frank B. Baker. The basics of item response theory. Published by the ERIC Clearinghouse on Assessment Evaluation 2001.
3. [De Bra, Calvi 1998]. Paul De Bra, Licia Calvi. AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, vol. 4, pp. 115-139, Taylor Graham Publishers, 1998.
4. [De Bra, Smits, Stash 2006]. Paul De Bra, David Smits, Natalia Stash. The Design of AHA!. Proceedings of the ACM Hypertext Conference, Odense, Denmark, August 23-25, 2006 pp. 133.
5. [Dellaert 2002]. Frank Dellaert. The Expectation Maximization Algorithm. College of Computing, Georgia Institute of Technology. Technical Report number GIT-GVU-02-20, February 2002.
6. [Dugad, Desai 1996]. R. Dugad, U. B. Desai. A tutorial on Hidden Markov models. Signal Processing and Artificial Neural Networks Laboratory, Dept of Electrical Engineering, Indian Institute of Technology, Bombay Technical Report No.: SPANN-96.1, 1996.
7. [Fink 2004]. Josef Fink, Gutachter: Prof. Dr. Alfred Kobsa, Prof. Dr. Rainer Unland. User Modeling Servers - Requirements, Design, and Evaluation. Universität Duisburg-Essen, Standort Essen, Fachbereich 6 Mathematik. Amsterdam, Netherlands: IOS Press (2004).

8. [Fröschl 2005]. Christoph Fröschl. User Modeling and User Profiling in Adaptive E-learning Systems. Masters Thesis at Graz University of Technology 2005. An approach for a service-based personalization solution for the research project AdeLE (Adaptive e-Learning with Eye-Tracking).
9. [Han, Kamber 2006]. Jiawei Han and Micheline Kamber. Data Mining: Concepts and Techniques. Second Edition. © 2006 by Elsevier Inc.
10. [Hauger, Köck 2007]. David Hauger and Mirjam Köck. State of the Art of Adaptivity in E-Learning Platforms. Proceedings of the 15th Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS 2007). ISBN: 978-3-86010-907-6 (2007).
11. [Heckmann 2005]. Dominikus Heckmann. Ubiquitous User Modeling. PhD Thesis at Universität des Saarlandes. Volume 297 Dissertationen zur Künstlichen Intelligenz.
12. [Kobsa 2006]. Alfred Kobsa. Generic User Modeling Systems. User Modeling and User-Adapted Interaction 2006 (UMUAI-2006).
13. [Kolb 1999]. D. A. Kolb. The Kolb Learning Style Inventory. Version 3. Boston: Hay Group, 1999.
14. [Neapolitan 2003]. Richard E. Neapolitan. Learning Bayesian Networks. Northeastern Illinois University Chicago, Illinois 2003.
15. [Stash, Cristea, De Bra 2005]. Natalia Stash, Alexandra Cristea, Paul De Bra. Explicit Intelligence in Adaptive Hypermedia: Generic Adaptation Languages for Learning Preferences and Styles. In Proceedings of HT2005 CIAH Workshop, Salzburg, Austria, 2005.
16. [Wolf 2003]. Christian Wolf. iWeaver: Towards "Learning Style"-based e-Learning in Computer Science Education. Australasian Computing Education Conference (ACE2003), Adelaide, Australia. Conferences in Research and Practice in Information Technology, Vol.20.
17. [Wolf 2007]. Christian Wolf. Construction of an Adaptive E-learning Environment to Address Learning Styles and an Investigation of the Effect of Media Choice. PhD by Project: RMIT University - Design and Social Context Portfolio, School of Education (Research) - Melbourne - Australia (January 2007).

*The summarization is complete; the next section is Vietnamese version*

Bản tóm tắt luận văn

**Hệ thống mô hình hóa người học  
hỗ trợ học thích nghi trong đào tạo từ xa**

# Hệ thống mô hình hóa người học

## hỗ trợ học thích nghi trong đào tạo từ xa

Kính gửi:

*Bản tóm tắt:* Hệ thống mô hình hóa người học hỗ trợ học thích nghi trong đào tạo từ xa

*Giáo viên hướng dẫn:* PGS. TS. Đồng Thị Bích Thủy

*Nghiên cứu sinh:* Nguyễn Phước Lộc

### Tóm tắt

Các phần mềm hiện nay đang có xu hướng cá nhân hóa nhưng muốn phục vụ tốt nhất nhu cầu người dùng thì cần biết đầy đủ thông tin về họ. Vì thế các hệ thu thập và quản lý thông tin người dùng – được gọi là hệ mô hình hóa càng trở nên quan trọng. Và thông tin về người dùng, được tổ chức dưới mô hình cụ thể, càng trở nên quý giá.

Luận văn tập trung vào ngữ cảnh học thích nghi, người dùng là người học, mô hình người dùng là mô hình người học. Luận văn hướng đến xây dựng một hệ thống mô hình hóa người học có đầy đủ các chức năng cơ bản như thu thập và cung cấp thông tin người học cùng với cơ chế suy diễn mạnh mẽ trên mô hình người học vững chắc.

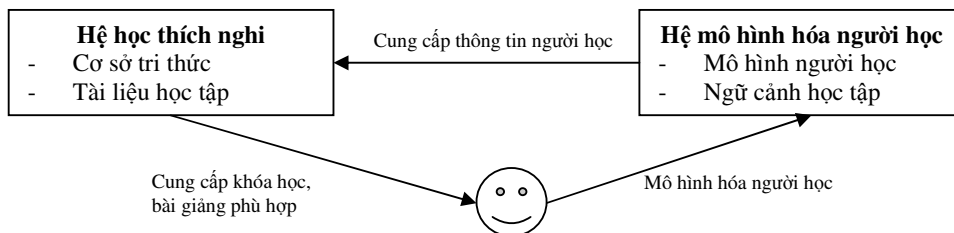
Bản tóm tắt gồm 4 chương:

- Chương 1: Giới thiệu các hệ mô hình hóa hiện có và mục tiêu luận văn
- Chương 2: Kiến trúc mô hình người học và hệ mô hình hóa người học
- Chương 3: Ba thành phần của mô hình người học
  - o 3.1: Thành phần tri thức  $K$
  - o 3.2: Thành phần cách học  $LS$
  - o 3.3: Thành phần lược sử học  $LH$
- Chương 4: Kết luận và hướng phát triển

### 1. Giới thiệu

Mô hình người dùng (user model) là tập thuộc tính về người dùng như: tri thức, tính cách, sở thích, học vấn, thông tin cá nhân... Các thông tin này rất rộng và tùy theo ngữ cảnh, mức quan trọng sẽ thay đổi, ví dụ trong lĩnh vực học thích nghi – ngữ cảnh chính của luận văn, thông tin về tri thức (knowledge), cách học (learning styles), mục tiêu (goals), nền tảng (backgrounds) quan trọng nhất. Người dùng cũng phụ thuộc vào ngữ cảnh, trong học thích nghi người dùng là người học, trong thương mại điện tử người dùng là khách hàng. Các thuật ngữ mô hình người dùng, mô hình người học (learner model), mô hình sinh viên (student model) cùng ý nghĩa trong luận văn.

Các hệ thống xây dựng và quản trị mô hình người dùng được gọi hệ mô hình hóa (user modeling system – UMS), hãy tưởng tượng mô hình người dùng như cơ sở dữ liệu và hệ mô hình hóa là hệ quản trị cơ sở dữ liệu. Hệ mô hình hóa cung cấp thông tin về người dùng cho hệ thích nghi để hệ thích nghi phục vụ tốt nhu cầu người dùng. Trong học điện tử (e-learning), hệ học thích nghi (adaptive learning system) sẽ cung cấp những phương pháp giảng dạy, khóa học, bài giảng, bài kiểm tra, bài tập phù hợp với từng đối tượng người học dựa trên thông tin về họ được cung cấp bởi hệ mô hình hóa.



**Hình 1.1.** Hệ mô hình hóa người dùng

Các hệ mô hình hóa hiện tại [Fröschl 2005], [Kobsa 2006] chia thành bốn nhóm:

- Shell mô hình hóa (user modeling shell): hệ này lỗi thời vì các shell chỉ là một thành phần (component) mô hình hóa gắn vào hệ thích nghi nên chưa mang tính chuyên dụng.
- Trình chủ mô hình hóa (user modeling server): hệ này hoạt động như hệ quản trị cơ sở dữ liệu với tính độc lập và chuyên dụng nhưng khả năng suy diễn thông tin tiềm ẩn về người dùng bị hạn chế.
- Hệ mô hình hóa dựa trên tác tử (agent-based user model): các mô hình người dùng được xây dựng như các tác tử (agent) tương tác với nhau. Tác tử được định nghĩa như những đơn thể (module) phần mềm hoạt động độc lập và tương tác với nhau. Các hệ này có khả năng phân tán.
- Mô hình người dùng di động (mobile user model): lưu trữ mô hình người dùng trên thiết bị di động như smart phone, mobile phone. Hệ này tương tác cao với người dùng, cập nhật tức thời nhưng hạn chế về dung lượng.

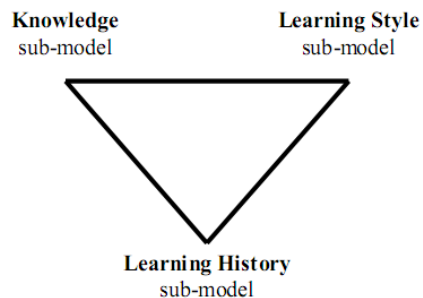
Mỗi hệ mô hình hóa đều có ưu nhược điểm cùng đặc trưng riêng nhưng nhìn chung, tập trung vào khả năng cung cấp thông tin và lưu trữ nên thiếu cơ chế suy diễn mạnh mẽ. Những hệ mô hình hóa hiện đại khác biệt với hệ quản trị cơ sở dữ liệu ở khả năng suy diễn; thông tin được thu thập về người dùng chưa đủ đáp ứng nhu cầu ngày càng cao của hệ thích nghi, các thông tin tiềm ẩn cần được khai thác, nghĩa là cần được suy diễn từ thông tin thu thập vốn có. Hơn nữa thông tin về người dùng rất nhiều nên cần chọn lọc những thuộc tính cần thiết nhất trong ngữ cảnh học thích nghi. Nhìn chung, luận văn nhằm giải quyết vấn đề chọn lọc thông tin và suy diễn thông tin mới từ người học. Vì vậy luận văn gồm hai mục tiêu

- Vì vậy mục tiêu thứ nhất của luận văn là xây dựng một hệ mô hình hóa có cơ chế suy diễn thông tin tiềm ẩn về người dùng trên cơ sở toán học cùng các giải thuật chặt chẽ.
- Mục tiêu thứ hai, nghiên cứu sẽ chọn lọc và mô hình hóa những đặc trưng thiết yếu về người học trong ngữ cảnh học thích nghi vì sự chính xác của cơ chế suy diễn phụ thuộc vào mức quan trọng của thông tin. Thông tin nhiều hay không quan trọng sẽ giảm hiệu năng tiến trình suy diễn. Do đó mục tiêu thứ hai hỗ trợ cho mục tiêu thứ nhất.

## 2. Kiến trúc hệ mô hình hóa người học

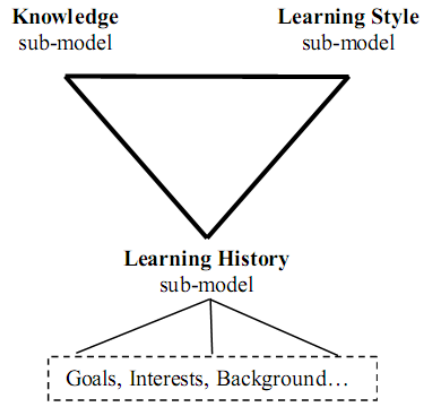
Luận văn đề xuất mô hình người dùng gồm 3 thành phần (hay mô hình con – sub-model) được chất lọc từ nhiều thuộc tính của người học: *tri thức* – knowledge (*K*), *cách học* – learning style (*LS*), *lược sử học* – learning history (*LH*). Ba thành phần này vừa độc lập vừa liên kết chặt để tạo thành một tam giác, vì vậy mô hình người dùng này được gọi mô hình tam giác (Triangular Learner Model – TLM). TLM với 3 thành phần tuy tinh gọn nhưng bao quát hết đặc trưng người dùng trong ngữ cảnh học thích nghi, bởi vì:

- *Tri thức (K)*, *cách học (LH)*, *lược sử học (LS)* là những thông tin cơ bản nhất về người học.
- *Tri thức*, *lược sử học* thay đổi thường xuyên nhưng *cách học* tương đối ổn định. Sự phối hợp giữa chúng bảo đảm tính toàn vẹn của thông tin người học.
- *Tri thức* là thông tin chuyên môn nhưng *cách học* là đặc trưng cá nhân. Sự phối hợp giữa chúng tận dụng tối đa lợi ích từ thông tin chuyên môn và không chuyên môn.



**Hình 2.1.** Mô hình người học TLM

TLM được thiết kế theo mô hình tam giác để dễ dàng mở rộng cho việc mô tả những đặc trưng khác của người dùng: sở thích (interests), mục tiêu (goals). Các đặc trưng mở rộng được dẫn xuất từ thành phần *lược sử học (LH)*. Vì vậy lược sử học là thành phần quan trọng nhất, được dùng để khởi tạo hai thành phần *tri thức*, *cách học* và xây dựng các đặc trưng mở rộng nêu trên bằng kỹ thuật khai khoáng.

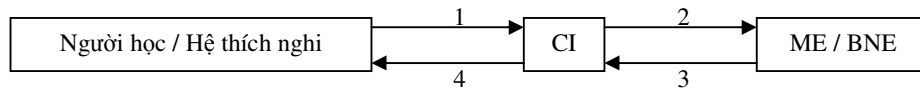


**Hình 2.2.** Mô hình TLM mở rộng

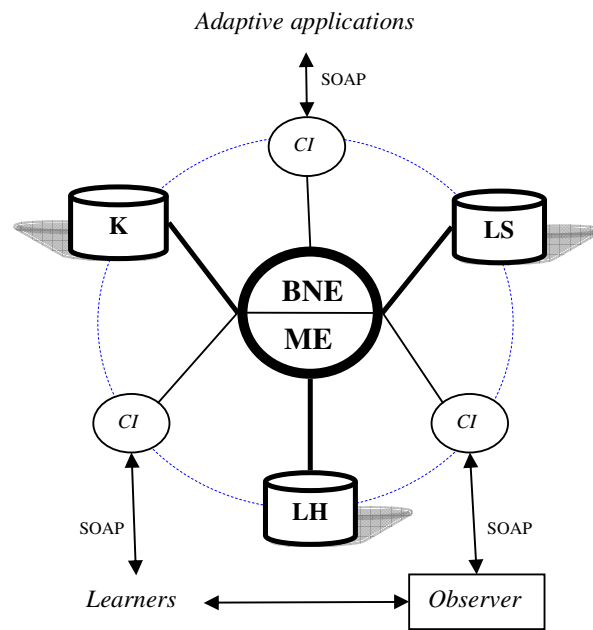
Mô hình TLM đã hoàn thành mục tiêu thứ hai của luận văn, mục tiêu thứ nhất đạt được bằng đề xuất hệ mô hình hóa Zebra: xây dựng & quản trị TLM và suy diễn thông tin mới từ TLM. Zebra hợp thành từ 2 guồng máy: guồng máy khai khoáng (mining engine – ME) và guồng máy mạng xác tín (belief network engine – BNE):

- Guồng máy ME: thu thập dữ liệu học tập của người dùng, theo dõi hoạt động học tập của họ, cấu trúc và cập nhật TLM. ME hoàn toàn sử dụng kỹ thuật khai khoáng. Vai trò ME rất quan trọng, bao quát cả TLM nhưng chủ yếu tập trung vào thành phần *lược sử học*. ME còn cung cấp thông tin cho guồng máy BNE, là đầu vào của BNE. ME còn có 3 chức năng khác bao gồm: phát hiện đặc trưng mở rộng (sở thích, mục tiêu), hỗ trợ tư vấn học tập và hỗ trợ học cộng tác.
- Guồng máy BNE: chịu trách nhiệm suy diễn các đặc trưng người dùng cụ thể là *tri thức (K)* và *cách học (LS)*. BNE sử dụng mạng xác tín như mạng Bayesian và mô hình Markov ẩn cho cơ chế suy diễn. BNE không hoạt động nhiều như ME nhưng phức tạp hơn.

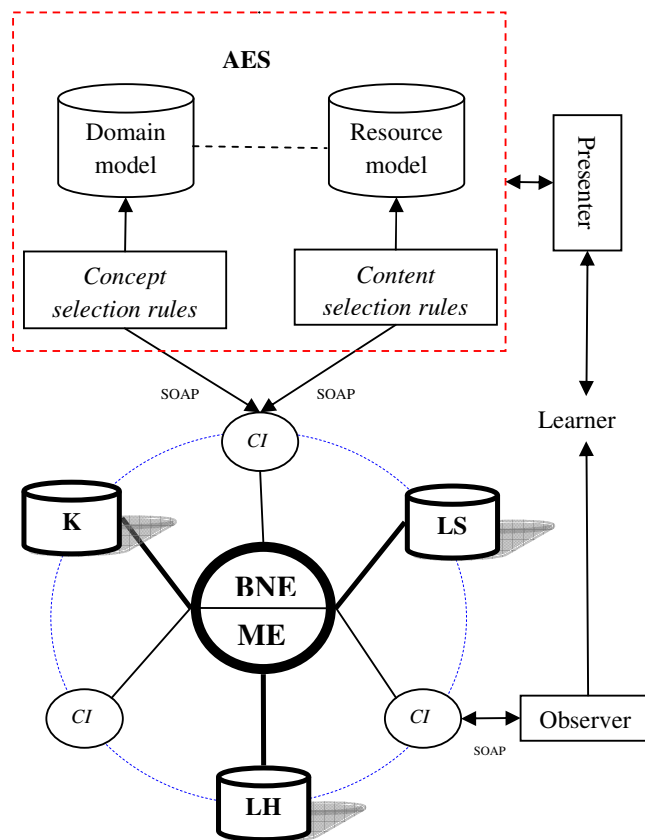
Ngoài ra Zebra còn cung cấp các giao tiếp (communication interfaces – CI) cho phép người học hay hệ thích nghi truy vấn hoặc cập nhật TLM. Do hệ mô hình hóa luôn tương tác với ứng dụng thích nghi (adaptive system / application) và TLM không thể bị thay đổi trực tiếp từ bên ngoài nên CI hoạt động giống như cổng giao tiếp. Quy trình ứng dụng thích nghi hay người học truy vấn hoặc cập nhật TLM:



1. Hệ thích nghi gửi yêu cầu đến CI thông qua các giao thức như SOAP, HTTP, RMI.
2. CI phân tích yêu cầu thành tham số đầu vào và gửi cho ME hay BNE.
3. ME hay BNE thực hiện yêu cầu và gửi kết quả cho CI.
4. CI trả kết quả cho ứng dụng thích nghi.



**Hình 2.3.** Kiến trúc hệ mô hình hóa Zebra



**Hình 2.4.** Tương tác hệ mô hình hóa Zebra với hệ học thích nghi AES



### 3. Ba thành phần của mô hình người học (TLM)

Trong chương 2, bản tóm tắt đã nét qua mô hình TLM gồm 3 thành phần hay mô hình con tạo thành một tam giác: Tri thức (K), cách học (LS), lược sử học (LH). Chương này sẽ tập trung khảo sát từng thành phần cùng với các đặc trưng, cách xây dựng và cơ chế suy diễn của chúng.

#### 3.1. Thành phần tri thức (K)

Tri thức là kết quả học tập, nên quan trọng đối với hệ thích nghi và là thước đo hệ thích nghi; một hệ thích nghi tốt là sau khi học, tri thức người học tăng lên. Luận văn đề xuất mô hình tri thức cùng những phương thức xây dựng và cải tiến mô hình tri thức.

#### Mô hình đề xuất: kết hợp mô hình chồng và mạng Bayesian

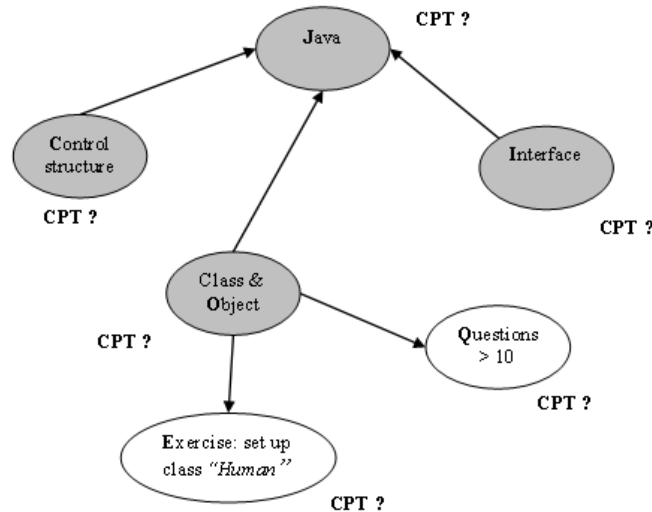
Luận văn đề xuất kết hợp mô hình chồng (overlay) với mạng Bayesian để xây dựng mô hình tri thức. Trong mô hình chồng, tri thức toàn thể (domain) được biểu diễn như tập các phần tử tri thức. Mỗi phần tử là bài giảng, bài tập, bài kiểm tra... và được đại diện bởi một số nguyên đo lường mức độ hiểu biết của người học. Mỗi người học được mô hình hóa như tập con của tri thức toàn thể. Mô hình chồng về cơ bản là đồ thị biểu diễn được đầy đủ tri thức người học với tất cả cấp độ liên kết tri thức từ đơn giản đến phức tạp nhưng mô hình chồng không có khả năng suy diễn và đánh giá kết quả đạt được tri thức người học. Do đó luận văn đề xuất sự kết hợp giữa mô hình chồng và mạng Bayesian để cung cấp cơ chế suy diễn (mục tiêu thứ nhất bài báo) cho mô hình chồng. Mạng Bayesian [14] là một đồ thị hữu hướng phi chỉ trình gồm một tập nút và một tập cạnh, mỗi quan hệ giữa 2 nút thể hiện qua cạnh, độ mạnh của mỗi quan hệ được định lượng qua bảng xác suất điều kiện (CPT). Cơ chế suy diễn trong mạng Bayesian dựa trên định luật xác suất Bayesian. Sự kết hợp mô hình chồng và mạng Bayesian như sau:

- Cấu trúc chồng (overlay) được diễn dịch thành mạng Bayesian, theo đó mỗi phần tử trong mô hình chồng thành một nút trong mạng Bayesian. Các tri thức kiểm định (bài kiểm tra, bài tập) trở thành nút bằng chứng và các tri thức bình thường thành nút giả thuyết hay nút ẩn.
- Mỗi quan hệ giữa các phần tử tri thức trở thành cạnh trong mạng Bayesian.

Ví dụ sau đây mô tả sự kết hợp mô hình chồng và mạng Bayesian, giả sử một khóa học Java căn bản cho sinh viên gồm 6 phần tử tri thức:

- Một khóa học Java ( $J$ ) và ba bài giảng: *control structure* ( $C$ ), *class & object* ( $O$ ), *interface* ( $I$ ).
- Một bài tập: *set up class Human* ( $E$ ).
- Một bài kiểm tra mà sinh viên cần trả lời đúng hơn 10 câu hỏi: *Questions > 10* ( $Q$ ).

Vấn đề cần giải quyết: kết thúc khóa học Java, sau khi hoàn thành bài tập  $E$  và bài kiểm tra  $Q$ , tri thức sinh viên được ước lượng bao nhiêu? Khi đó tri thức sinh viên là xác suất hậu biên (marginal posterior probability) của nút  $J$  khi  $J = 1$ , ký hiệu  $Pr(J=1)$ .



**Hình 3.1.** Sự kết hợp mô hình chồng và mạng Bayesian  
(nút bằng chứng không tô đậm, nút ẩn tô đậm)

$J, C, O, I$  là 4 nút ẩn (tô đậm),  $E$  và  $Q$  là 2 nút bằng chứng. Do độ mạnh mỗi cạnh lượng hóa bằng bảng CPT chưa được xác định nên xác suất biên  $P(J=I)$  chưa thể tính. Luận văn đề xuất 2 bước để xác định bảng CPT:

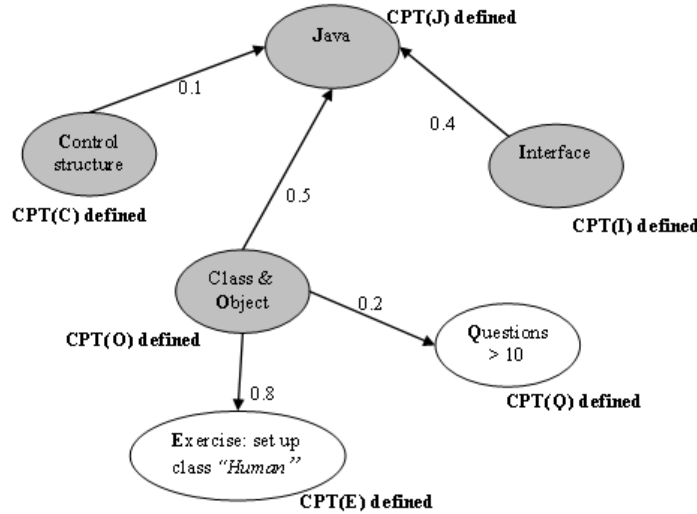
1. Mỗi cạnh được gán một trọng số diễn tả mức quan trọng của bài học, bài giảng, bài kiểm tra...
2. Bảng CPT được xây dựng từ trọng số qua công thức cộng Union.

Giả sử trọng số của bài giảng *control structure* ( $C$ ), *class & object* ( $O$ ), *interface* ( $I$ ) lần lượt là  $w_1=0.1$ ,  $w_2=0.5$ ,  $w_3=0.4$ . Bài giảng *class & object* quan trọng nhất kế tiếp là *interface* và *control structure*. Trọng số của bài tập  $E$  và bài kiểm tra  $Q$  lần lượt  $0.8$  và  $0.2$ . Bảng CPT của khóa học Java được tính qua công thức cộng Union như sau:

$$Pr(J \mid C, O, I) = w_1 * h_1 + w_2 * h_2 + w_3 * h_3$$

(Vui lòng xem mục III.1.1.3 trong luận văn – chứng minh công thức cộng Union)

$$\text{Where } h_1 = \begin{cases} 1 & \text{if } C = J \\ 0 & \text{otherwise} \end{cases} \quad h_2 = \begin{cases} 1 & \text{if } O = J \\ 0 & \text{otherwise} \end{cases} \quad h_3 = \begin{cases} 1 & \text{if } I = J \\ 0 & \text{otherwise} \end{cases}$$



**Hình 3.2.** Mô hình Bayesian được đặc tả đầy đủ bằng các bảng CPT

C	O	I	$Pr(J = 1)$	$Pr(J = 0)$ $1 - p(J = 1)$
1	1	1	<b>1.0</b> $(0.1*1 + 0.5*1 + 0.4*1)$	<b>0.0</b>
1	1	0	<b>0.6</b> $(0.1*1 + 0.5*1 + 0.4*0)$	<b>0.4</b>
1	0	1	<b>0.5</b> $(0.1*1 + 0.5*0 + 0.4*1)$	<b>0.5</b>
1	0	0	<b>0.1</b> $(0.1*1 + 0.5*0 + 0.4*0)$	<b>0.9</b>
0	1	1	<b>0.9</b> $(0.1*0 + 0.5*1 + 0.4*1)$	<b>0.1</b>
0	1	0	<b>0.5</b> $(0.1*0 + 0.5*1 + 0.4*0)$	<b>0.5</b>
0	0	1	<b>0.4</b> $(0.1*0 + 0.5*0 + 0.4*1)$	<b>0.4</b>
0	0	0	<b>0.0</b> $(0.1*0 + 0.5*0 + 0.4*0)$	<b>1.0</b>

**Bảng 3.1:** Bảng CPT của nút *Java* ( $J$ )

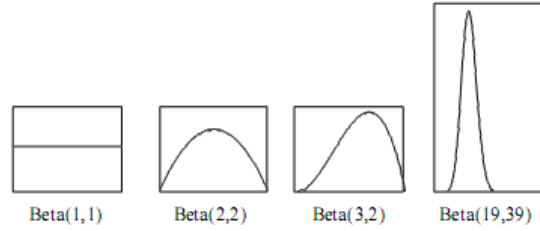
Khi mạng Bayesian được đặc tả đầy đủ bằng các bảng CPT, xác suất hậu biên  $Pr(J=I)$  được tính dễ dàng (xem luận văn mục III.1.1.2 trang 75). Mô hình tri thức (mạng Bayesian) được cải tiến qua hai phương thức:

1. Học tham số (parameter learning): Các bảng CPT (tham số mạng Bayesian), xác định qua đặc tả trọng số và công thức cộng Union, cần được cải thiện khi Zebra hoạt động sau một thời gian. Đây là vấn đề học tham số. Luận văn sử dụng giải thuật Expectation Maximization (EM) [Neapolitan 2003], [Dellaert 2002] và đề xuất giải thuật Maximum Likelihood Estimation (MLE) cho học tham số.
2. Học cấu trúc (structure learning): cấu trúc nút, cạnh, các bảng CPT của mạng có thể rất khác với mô hình dẫn xuất từ phương pháp chồng (overlay). Luận văn đề xuất một giải thuật tối ưu dùng mạng Bayesian động [Neapolitan 2003] cho học cấu trúc.

### Cải tiến mô hình tri thức bằng giải thuật EM

Phương pháp EM giả định hàm mật độ xác suất  $\beta(f)$  trên mỗi bảng CPT tuân theo phân phối beta:

$$\beta(f) = \frac{\Gamma(N)}{\Gamma(a)\Gamma(b)} f^{a-1} (1-f)^{b-1}$$



**Hình 3.3.** Phân phối beta

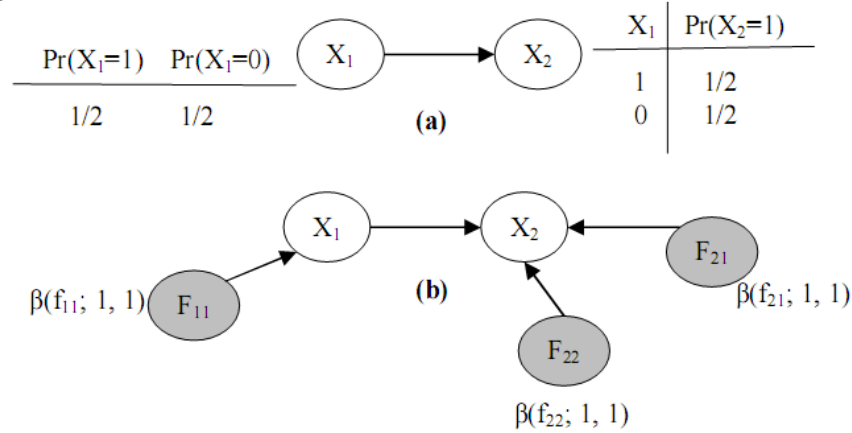
Mỗi nút trong mạng Bayesian là biến ngẫu nhiên. Xác suất tại nút  $X$ :

$$\Pr(X = 1) = E(\beta(f)) = \frac{a}{a+b}$$

Trong đó “ $a$ ” tượng trưng số lần thành công (ví dụ,  $f = 1$ ) trong  $N = a + b$  phép thử. Xác suất tại nút  $X$  sau  $M = s + t$  phép thử:

$$\Pr(X = 1 | M \text{ trials}) = E(\beta(f) | M \text{ trials}) = \frac{a + s}{a + b + s + t} = \frac{a + s}{N + M}$$

Trong đó  $s$  tượng trưng số phép thử thành công qua  $M = s + t$  phép thử. Giả sử mạng Bayesian được đặc tả như sau với 5 phép thử  $X^{(1)}, X^{(2)}, X^{(3)}, X^{(4)}, X^{(5)}$ :

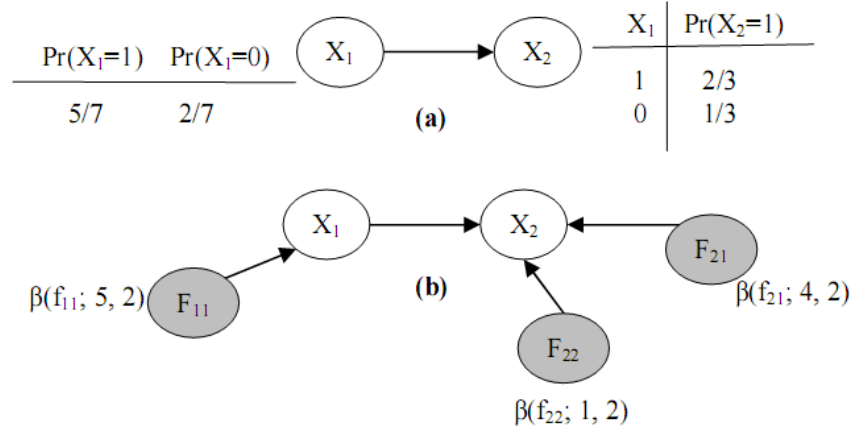


	$x_1$	$x_2$
$X^{(1)}$	$X_1^{(1)} = 1$	$X_2^{(1)} = 1$
$X^{(2)}$	$X_1^{(2)} = 1$	$X_2^{(2)} = 1$
$X^{(3)}$	$X_1^{(3)} = 1$	$X_2^{(3)} = 1$
$X^{(4)}$	$X_1^{(4)} = 1$	$X_2^{(4)} = 0$
$X^{(5)}$	$X_1^{(5)} = 0$	$X_2^{(5)} = 0$

$$\begin{aligned} s_{11} &= 1 + 1 + 1 + 1 + 0 = 4 & t_{11} &= 0 + 0 + 0 + 0 + 1 = 1 \\ s_{21} &= 1 + 1 + 1 + 0 + 0 = 3 & t_{21} &= 0 + 0 + 0 + 0 + 1 = 1 \\ s_{22} &= 0 + 0 + 0 + 0 + 0 = 0 & t_{21} &= 0 + 0 + 0 + 0 + 1 = 1 \end{aligned}$$

**Hình 3.4.** Mạng Bayesian, bảng CPT và các phép thử

Giải thuật EM sẽ cộng dồn sau mỗi phép thử để thay đổi xác suất tại nút  $X_i$ :



$$\begin{aligned}
 \beta(f_{11}|E) &= \beta(f_{11}; a_{11}+s_{11}, b_{11}+t_{11}) = \beta(f_{11}; 1+4, 1+1) = \beta(f_{11}; 5, 2) \\
 \beta(f_{21}|E) &= \beta(f_{21}; a_{21}+s_{21}, b_{21}+t_{21}) = \beta(f_{21}; 1+3, 1+1) = \beta(f_{21}; 4, 2) \\
 \beta(f_{22}|E) &= \beta(f_{22}; a_{22}+s_{22}, b_{22}+t_{22}) = \beta(f_{22}; 1+0, 1+1) = \beta(f_{22}; 1, 2) \\
 s_{11} &= 1+1+1+1+0=4 & t_{11} &= 0+0+0+0+1=1 \\
 s_{21} &= 1+1+1+0+0=3 & t_{21} &= 0+0+0+0+1=1 \\
 s_{22} &= 0+0+0+0+0=0 & t_{22} &= 0+0+0+0+1=1
 \end{aligned}$$

**Hình 3.5.** Mạng Bayesian, bảng CPT được học qua giải thuật EM

Nếu dữ liệu các phép thử bị thiếu, vui lòng xem luận văn mục III.1.3.2 trang 92. Giải thuật EM đơn giản và hiệu quả khi thực hiện tuần tự sau mỗi lần lặp. Trong trường hợp với một lượng lớn dữ liệu huấn luyện, giải thuật MLE thích hợp hơn.

### Cải tiến mô hình tri thức bằng giải thuật MLE

Ý tưởng chính của MLE là tìm điểm cực đại của hàm khả năng. Cho hàm beta  $B(x, y) = \Gamma(x) \Gamma(y) / \Gamma(x + y)$  where  $\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt$  là hàm gamma. Hàm khả năng của phân phối beta, định nghĩa như sau:

$$L(a, b) = \prod_{i=1}^n f(x_i, a, b) = \prod_{i=1}^n \frac{1}{B(a, b)} x_i^{a-1} (1-x_i)^{b-1} = \frac{1}{B^n(a, b)} \prod_{i=1}^n x_i^{a-1} \prod_{i=1}^n (1-x_i)^{b-1}$$

Hàm khả năng được hạ logarithm như sau:

$$LnL(a, b) = -n \ln(B(a, b)) + (a-1) \sum_{i=1}^n \ln x_i + (b-1) \sum_{i=1}^n \ln(1-x_i)$$

Mục tiêu của MLE là tìm hai tham số  $a$  và  $b$  sao cho hàm khả năng đạt cực đại. Sau khi lấy đạo hàm riêng bậc nhất theo  $a$  và  $b$ , luận văn chứng minh rằng  $a$  và  $b$  thỏa hệ phương trình sau (xem luận văn mục III.1.5.2 trang 113):

$$\begin{cases} e^a \sum_{k=1}^b \frac{(-1)^k}{k} \binom{b}{k} = \frac{1}{n} \sum_{i=1}^n \ln x_i \\ e^b \sum_{k=1}^a \frac{(-1)^k}{k} \binom{a}{k} = \frac{1}{n} \sum_{i=1}^n \ln(1-x_i) \end{cases} \Leftrightarrow \begin{cases} F_1(a, b) = L_1 \\ F_2(a, b) = L_2 \end{cases}$$

Where  $F_1(a, b) = e^a \sum_{k=1}^b \frac{(-1)^k}{k} \binom{b}{k}$ ,  $L_1 = \frac{1}{n} \sum_{i=1}^n \ln x_i$ ,  $F_2(a, b) = e^b \sum_{k=1}^a \frac{(-1)^k}{k} \binom{a}{k}$ ,  $L_2 = \frac{1}{n} \sum_{i=1}^n \ln(1-x_i)$

Luận văn đề xuất một giải thuật lặp để giải gần đúng  $a$  và  $b$  trong phương trình 3.2 với định đề “ $a$  và  $b$  dao động trong đoạn  $[1, n]$  với  $n$  là một số nguyên dương và không lớn hơn số phép thử trong dữ liệu huấn luyện”. Giải thuật lặp cho  $a$  và  $b$  lần lượt nhận giá trị từ 1 đến  $n$ , tại mỗi lần lặp hai độ lệch  $\Delta_1 = F_1(a, b) - L_1$  và  $\Delta_2 = F_2(a, b) - L_2$  được tính và độ lệch chuẩn tắc  $\Delta = \sqrt{\Delta_1^2 + \Delta_2^2}$  được xác định. Cặp tham số  $(a, b)$  nào có độ lệch chuẩn tắc nhỏ nhất sẽ là nghiệm  $(\hat{a}, \hat{b})$  của hệ phương trình 3.2.

```

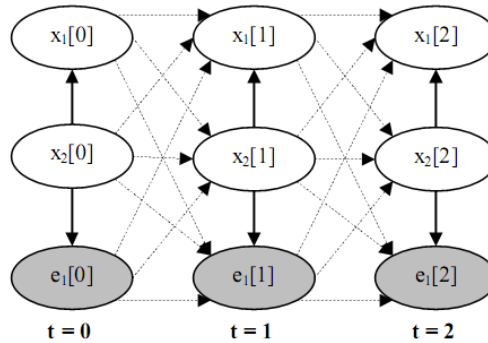
min  $\Delta = +\infty$ ;
 $\hat{a} = \hat{b} = 1$  (uniform distribution)
For  $a = 1$  to  $n$  do
  For  $b = 1$  to  $n$  do
     $\Delta_1 = F_1(a, b) - L_1$ 
     $\Delta_2 = F_2(a, b) - L_2$ 
     $\Delta = \sqrt{\Delta_1^2 + \Delta_2^2}$ 
    If  $\Delta < \min \Delta$  then
      min  $\Delta = \Delta$ 
       $\hat{a} = a$ 
       $\hat{b} = b$ 
    End If
  End for  $b$ 
End For  $a$ 
( $\hat{a}$  and  $\hat{b}$  are optimal parameters)

```

### Cải tiến mô hình tri thức bằng mạng Bayesian động

Mạng Bayesian tĩnh có ưu điểm đơn giản nhưng chưa mô tả được quá trình học tập của người dùng, chưa thể hiện được yếu tố thời gian. Trong ngữ cảnh học thích nghi, việc mô tả tiến trình học tập của người học qua thời gian là rất cần thiết. Mạng Bayesian động [Neapolitan 2003] gồm một chuỗi các  $G_t$ , với  $G_t$  là mạng Bayesian tại thời điểm thứ  $t$ :

- Mạng  $G_0$  tại thời điểm khởi đầu gồm tập nút, tập cạnh và các bảng CPT
- Mạng  $G_t$  tại thời điểm  $t$ , ngoài tập nút, tập cạnh và các bảng CPT còn có xác suất chuyển trạng thái từ  $G_{t-1}$  sang  $G_t$ , ký hiệu  $Pr(G_t | G_{t-1})$ , đây cũng là xác suất điều kiện. Nếu có  $n$  thời điểm thì có  $n - 1$  xác suất chuyển trạng thái.



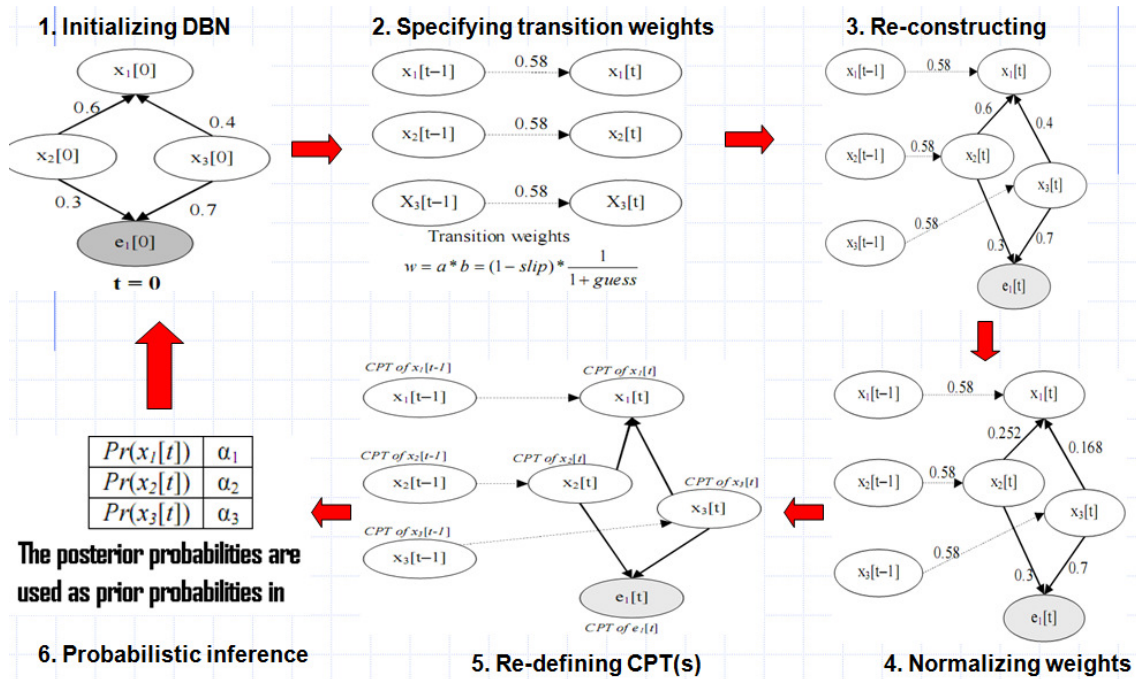
Hình 3.6. Mạng Bayesian động với thời điểm  $t = 0, 1, 2$

Mạng Bayesian động (dynamic Bayesian network – DBN) có nhược điểm kích thước quá lớn gây bùng nổ tổ hợp khi suy diễn lần lượt qua thời gian dài. Luận văn đề xuất một giải thuật mới xây dựng mạng Bayesian động với kích thước không đổi qua thời gian dựa trên thuộc tính Markov “xác suất tại thời điểm  $t + 1$  chỉ phụ thuộc vào xác suất tại thời điểm  $t$  trước đó, nói cách khác chỉ một xác suất chuyển trạng thái được đặc tả”. Tại mọi thời điểm  $t$ , giải thuật chỉ đặc tả  $G_{t-1}$  và  $G_t$  cùng với  $Pr(G_t | G_{t-1})$ . Ưu điểm của giải thuật, ngoài thu gọn tối đa kích thước mạng Bayesian, tốc độ suy diễn chậm hơn không đáng kể so với mạng Bayesian bình thường. Giải thuật gồm nhiều lần lặp, mỗi lần lặp thực hiện tại một thời điểm. Một lần lặp gồm 6 bước tích lũy theo chu trình:

1. Khởi tạo mạng Bayesian động (Initializing DBN).
2. Đặc tả trọng số chuyển trạng thái (Specifying transition weights).
3. Tái cấu trúc mạng Bayesian động (Re-constructing).
4. Chuẩn hóa trọng số (Normalizing weights).
5. Tái định nghĩa các bảng CPT cùng với xác suất chuyển trạng thái (Re-defining CPT).
6. Tính xác suất hậu biên cho lần lặp tiếp theo (Probabilistic inference).

Sau mỗi lần lặp, mạng Bayesian động được tiên hóa, cơ chế suy diễn chính xác hơn. Vui lòng xem luận văn mục III.4.1.2 với chi tiết từng bước. Tại bước 2, khái niệm *quên* (slip) và *đoán* (guess) được định nghĩa để đặc tả trọng số chuyển trạng thái.

- *slip*: Xác suất người học không làm được bài tập  $X$  nhưng trước đó đã hiểu bài  $X$  (do quên),  $Pr(not X_t | X_{t-1})$ .
- *guess*: Xác suất người học làm được bài tập  $X$  nhưng trước đó chưa hiểu bài  $X$  (do đoán),  $Pr(X_t | not X_{t-1})$ .
- *Trọng số chuyển trạng thái*:  $(1 - slip) \frac{1}{1 + guess}$



Hình 3.7. Giải thuật xây dựng mạng Bayesian động

Tóm lại, về căn bản, thành phần hay mô hình con tri thức ( $K$ ) trong TLM là sự kết hợp giữa mô hình chồng và mạng Bayesian. Cấu trúc của  $K$  được xây dựng bởi chuyên gia theo mô hình chồng, cài đặt chương trình thử nghiệm vẫn mô hình chồng (xem luận văn mục III.1.2 trang 76 và [De Bra, Calvi 1998], [De Bra, Smits, Stash 2006]) nhưng suy diễn trong  $K$  dùng mạng Bayesian. Ngoài ra luận văn còn đề xuất 3 phương pháp cải tiến  $K$ : học tham số với giải thuật EM, học tham số với giải thuật MLE, học cấu trúc trúc và mô hình hóa  $K$  theo thời gian bằng mạng Bayesian động.

### 3.2. Thành phần cách học (LS)

*Cách học* (LS) [Stash, Cristea, De Bra 2005], [Kolb 1999] được định nghĩa như là một tập hợp những đặc tính cá nhân, tâm sinh lý của người học thể hiện ở cách người học nhận thức, tương tác và đáp ứng với môi trường học tập. Lưu ý, trong luận văn, *LS* là viết tắt của *cách học*. Có nhiều lý thuyết tâm lý về LS như Dunn and Dunn, Witkin, Riding, Myers-Briggs, Kolb, Honey – Mumford, Felder – Silverman. Luận văn chọn mô hình Markov ẩn [Dugad, Desai 1996] để biểu diễn lý thuyết tâm lý Honey – Mumford và Felder – Silverman. Theo Honey – Mumford và Felder – Silverman, LS phân thành các dạng thức theo từng cặp đối lập sau:

- *Văn cảm* (verbal) / *Quan cảm* (visual): người *văn cảm* thích học qua tài liệu và sách vở, người *quan cảm* thích xem hình ảnh để học.
- *Chủ động* (activist) / *Trầm định* (reflector): người *chủ động* hiểu vấn đề qua thảo luận và áp dụng, người *trầm định* nghĩ thấu suốt mọi lẽ trước khi thực hành.
- *Lý thuyết* (theorist) / *Thực hành* (Pragmatist): người *lý thuyết* suy nghĩ vấn đề theo từng bước có logic và liên kết các sự kiện rời rạc vào mạch lý thuyết nhất quán. Người *thực hành* luôn có đầu óc thực hành, giải quyết vấn đề theo hướng thử nghiệm và kiểm định.

Mô hình Markov ẩn là cơ chế suy diễn gồm 5 thành phần  $\langle S, \Pi, A, \square, B \rangle$ :

- Tập trạng thái  $S = \{s_1, s_2, \dots, s_n\}$ . Một tiến trình ngẫu nhiên  $P$  là tập các trạng thái  $P = \{x_1, x_2, \dots, x_k\}$  với  $x_i \in S$ . Tiến trình  $P$  thỏa điều kiện Markov “xác suất điều tại trạng thái  $x_{i+1}$  chỉ phụ thuộc vào trạng thái trước đó  $x_i$ ,  $Pr(x_{i+1} | x_1, x_2, \dots, x_i) = Pr(x_{i+1} | x_i)$ ”. Trong mô hình Markov ẩn, tập  $S$  là tập ẩn
- Phân bố xác suất khởi tạo cho từng trạng thái,  $\Pi = (\pi_1, \pi_2, \dots, \pi_n)$  với  $\pi_i = Pr(s_i)$  là xác suất khởi tạo của trạng thái  $s_i$ .
- Ma trận chuyển trạng thái  $A = (a_{ij})$  với  $a_{ij}$  là xác suất chuyển trạng thái  $s_i$  sang trạng thái  $s_j$ . Ta có  $\forall s_i \in S, \sum_{s_j \in S} a_{ij} = 1$ .
- Tập các quan sát  $\Theta = \{\theta_1, \theta_2, \dots, \theta_m\}$ .
- Ma trận quan sát  $B = (b_{ij})$  với  $b_{ij}$  là xác suất xảy ra trạng thái  $s_i$  với quan sát  $\theta_j$ .

Cho trước một tiến trình ngẫu nhiên, mô hình Markov ẩn tìm ra các trạng thái ẩn qua những quan sát. Luận văn sử dụng mô hình Markov ẩn để phát hiện *LS* của người thông qua các quan sát về quá trình học tập của học. Trước tiên, lý thuyết tâm lý Honey – Mumford và Felder – Silverman được mô hình hóa bằng mô hình Markov thông qua định nghĩa 5 thành phần  $\langle S, \Pi, A, \Theta, B \rangle$ . Giả sử chọn 3 cặp dạng thức LS: *Verbal / Visual*, *Activist / Reflector*, *Theorist / Pragmatist*, chúng ta xây dựng 3 mô hình Markov  $\Delta_1 = \langle S_1, \Theta_1, A_1, B_1, \Pi_1 \rangle$ ,  $\Delta_2 = \langle S_2, \Theta_2, A_2, B_2, \Pi_2 \rangle$  và  $\Delta_3 = \langle S_3, \Theta_3, A_3, B_3, \Pi_3 \rangle$ , gồm 5 bước:

1. Định nghĩa các *trạng thái*, **mỗi trạng thái đồng nhất với một LS**. Luận văn đề xuất tìm trạng thái, nghĩa là gián tiếp tìm LS.  
 $S_1 = \{\text{verbal}, \text{visual}\}$ ,  $S_2 = \{\text{activist}, \text{reflector}\}$ ,  $S_3 = \{\text{theorist}, \text{pragmatist}\}$ .
2. Định nghĩa các *phân bố xác suất khởi tạo*: phân bố đều được khởi tạo cho mỗi  $\Pi_i$   
 $\Pi_1 = \{0.5, 0.5\}$ ;  $Pr(\text{verbal}) = Pr(\text{visual}) = 0.5$   
 $\Pi_2 = \{0.5, 0.5\}$ ;  $Pr(\text{activist}) = Pr(\text{reflector}) = 0.5$   
 $\Pi_3 = \{0.5, 0.5\}$ ;  $Pr(\text{theorist}) = Pr(\text{pragmatist}) = 0.5$
3. Định nghĩa các *ma trận chuyển trạng thái*: do người học có xu hướng giữ *cách học* riêng của họ, xác suất điều kiện cùng một trạng thái luôn cao.

	verbal	visual		activist	reflector		theorist	pragmatist
verbal	0.7	0.3	activist	0.7	0.3	theorist	0.7	0.3
visual	0.3	0.7	reflector	0.3	0.7	pragmatist	0.3	0.7

**Bảng 3.2.** Các ma trận chuyển trạng thái  $A_1, A_2, A_3$

4. Định nghĩa các *quan sát*: dựa trên mối quan hệ giữa tài liệu học tập và LS của người học. Giả sử mỗi tài liệu học tập được gán 3 thuộc tính:

- Thuộc tính định dạng có 3 giá trị: *text, picture, video*.
- Thuộc tính loại tài liệu có 4 giá trị: *theory, example, exercise, puzzle*.
- Thuộc tính tương tác biểu diễn mức độ yêu cầu tương tác của người dùng với tài liệu học tập, giá trị thuộc tính càng cao người học càng tương tác với nhau / với tài liệu trong quá trình học tập. Thuộc tính có 3 giá trị: *low, medium, high*.

Bất cứ khi nào người học chọn một tài liệu học tập, các *quan sát* sẽ phát sinh. Ví dụ, người học chọn tài liệu có thuộc tính định dạng *text*, loại *theory*, mức tương tác *low*; khi đó 3 quan sát phát sinh: *text, theory, low* (interaction); có thể suy diễn người học này có LS *theorist*. Dạng thức *verbal / visual* liên quan định dạng tài liệu, dạng thức *activist / reflector* và *theorist / pragmatist* liên quan thuộc tính loại và mức tương tác của tài liệu. Nói cách khác, **các quan sát đồng nhất với các thuộc tính tài liệu** và chúng được định nghĩa như sau:

- $\Theta_1 = \{text, picture, video\}$ .
- $\Theta_2 = \{theory, example, exercise, puzzle, low \text{ (mức tương tác)}, medium \text{ (mức tương tác)}, high \text{ (mức tương tác)}\}$ .
- $\Theta_3 = \{theory, example, exercise, puzzle, low \text{ (mức tương tác)}, medium \text{ (mức tương tác)}, high \text{ (mức tương tác)}\}$ .

5. Định nghĩa *ma trận quan sát*

Các quan sát (thuộc tính tài liệu) khác nhau ảnh hưởng lên trạng thái (LS) với cấp độ khác nhau. Luận văn đặc tả cấp độ ấy bằng những trọng số, trọng số càng lớn, mức ảnh hưởng của quan sát càng cao. Ví dụ, đối với người *văn cảm*, trọng số các quan sát: *text, picture, video* theo thứ tự giảm dần vì họ thích học bằng văn bản; ngược lại với người *quan cảm*, trọng số các quan sát: *text, picture, video* theo thứ tự tăng dần. Trọng số được chuẩn hóa trở thành xác suất điều kiện và ma trận quan sát là tập các xác suất điều kiện này. Ví dụ:  $Pr(text | verbal) = 0.6$ ,  $Pr(picture | verbal) = 0.3$ ,  $Pr(video | verbal) = 0.1$ .

Các ma trận quan sát  $B_1, B_2, B_3$  có đồng là LS, cột là quan sát, được đặc tả như sau:

	<i>text</i>	<i>picture</i>	<i>video</i>
<i>verbal</i>	0.6	0.3	0.1
<i>visual</i>	0.2	0.4	0.4

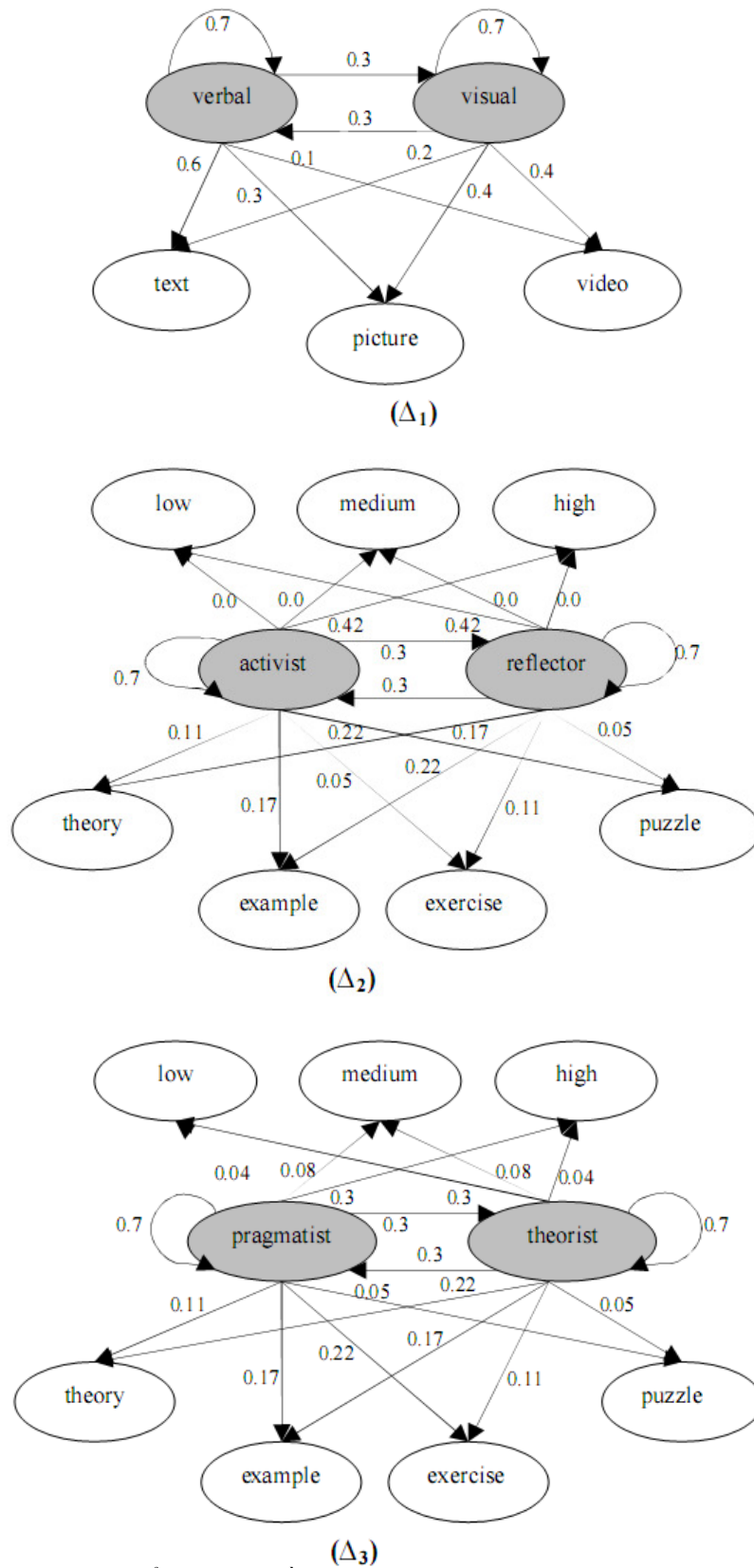
	<i>theory</i>	<i>example</i>	<i>exercise</i>	<i>puzzle</i>	<i>low</i>	<i>medium</i>	<i>high</i>
<i>activist</i>	0.11	0.17	0.05	0.22	0	0	0.42
<i>reflector</i>	0.17	0.22	0.11	0.05	0.42	0	0

	<i>theory</i>	<i>example</i>	<i>exercise</i>	<i>puzzle</i>	<i>low</i>	<i>medium</i>	<i>high</i>
<i>pragmatist</i>	0.11	0.17	0.22	0.05	0.04	0.08	0.3
<i>theorist</i>	0.22	0.17	0.11	0.05	0.3	0.08	0.04

**Bảng 3.3.** Các ma trận quan sát  $B_1, B_2, B_3$

Sau 5 bước nêu trên, ba mô hình Markov ẩn  $\Delta_1, \Delta_2, \Delta_3$  được xây dựng như sau:





**Hình 3.8.** Ba mô hình Markov ẩn  $\Delta_1$ ,  $\Delta_2$ ,  $\Delta_3$  lần lượt cho ba dạng thức LS *Verbal / Visual*, *Activist / Reflector*, *Theorist / Pragmatist*

Sau khi mô hình Markov ẩn được xây dựng, từ việc chọn tài liệu của người học và quá trình học của họ, thuật toán Viterbi được dùng để tìm trạng thái (LS). Giả sử người học chọn 3 tài liệu  $TL_1$ ,  $TL_2$ ,  $TL_3$  có các thuộc tính như sau:

	Định dạng	Loại	Mức tương tác
$TL_1$	picture	theory	<i>không gán</i>
$TL_2$	text	example	<i>không gán</i>
$TL_3$	Text	<i>không gán</i>	low

**Bảng 3.4.** Tài liệu học được chọn

Chuỗi *quan sát* được rút từ việc chọn tài liệu

Các mô hình Markov ẩn	Chuỗi quan sát
$\Delta_1$ : Dạng thức <i>Verbal / Visual</i>	picture $\rightarrow$ text $\rightarrow$ text
$\Delta_2$ : Dạng thức <i>Activist / Reflector</i>	theory $\rightarrow$ example $\rightarrow$ low
$\Delta_3$ : Dạng thức <i>Theorist / Pragmatist</i>	theory $\rightarrow$ example $\rightarrow$ low

**Bảng 3.5.** Các chuỗi quan sát

Thuật toán Viterbi áp dụng để tìm chuỗi chuyển trạng thái phù hợp nhất với chuỗi quan sát, tức là tìm LS của người học

Mô hình Markov	Chuỗi quan sát	Chuỗi chuyển trạng thái	Cách học (LS)
$\Delta_1$	picture $\rightarrow$ text $\rightarrow$ text	visual $\rightarrow$ verbal	verbal
$\Delta_2$	theory $\rightarrow$ example $\rightarrow$ low	reflector $\rightarrow$ reflector $\rightarrow$ reflector	reflector
$\Delta_3$	theory $\rightarrow$ example $\rightarrow$ low	theorist $\rightarrow$ theorist $\rightarrow$ theorist	theorist

**Bảng 3.6.** Chuỗi chuyển trạng thái

Dễ dàng nhận thấy người học này là người văn cảm, trầm định và là lý thuyết gia.

Tóm lại luận văn đề xuất 5 bước để xây dựng thành phần LS bằng mô hình Markov ẩn theo lý thuyết tâm lý Honey – Mumford và Felder – Silverman, theo đó cách học (LS) được xem như trạng thái và những tài liệu người học chọn trở thành các quan sát. Từ các quan sát, LS của người học được phát hiện dựa trên mô hình Markov ẩn. Giải thuật rất linh động với bất kỳ lý thuyết tâm lý nào ngoài Honey – Mumford và Felder – Silverman do mỗi dạng thức LS được đặc tả bằng một mô hình Markov ẩn và số lượng mô hình (cũng như dạng thức) không giới hạn, số lượng thuộc tính tài liệu (những quan sát) cũng tùy chọn.

### 3.3. Thành phần lược sử học (LH)

Thành phần lược sử học LH nằm ở đáy TLM vì nó là thành phần quan trọng nhất, làm cơ sở và đầu vào để xây dựng thành phần K và LS. LH được định nghĩa như bản mô tả hoạt động học tập của người học bao gồm: truy cập tài liệu, thời gian sử dụng máy tính, làm bài kiểm tra, làm bài tập, tương tác với giáo viên và bạn học. LH gồm 4 chức năng chính:

1. Cung cấp thông tin để xây dựng và hỗ trợ thành phần K và LS. Ví dụ, thành phần K cần những bằng chứng học tập như kết quả bài kiểm tra, tần số truy cập tài liệu để thực hiện cơ chế suy diễn đánh giá tri thức người học; thành phần LS cần những quan sát từ việc chọn tài liệu để tìm chuỗi chuyển trạng thái.
2. Hỗ trợ khuyến nghị tài liệu học tập.
3. Khai thác dữ liệu học tập để phát hiện các đặc trưng mở rộng cụ thể là sở thích người học (xem hình 2.2: Mô hình TLM mở rộng).
4. Xây dựng nhóm hay cộng đồng người học để hỗ trợ học cộng tác.

Guồng máy ME xây dựng và quản lý LH bằng các kỹ thuật khai khoáng [Han, Kamber 2006]. Bản tóm tắt này tập trung vào chức năng 2, 3, 4 của thành phần LH. Trong chức năng 1, LH thường được lưu trữ dưới dạng tập tin XML.

### Khuyến nghị tài liệu học tập qua khai thác mẫu tuần tự

Các hoạt động học tập như làm bài kiểm tra, truy cập tài liệu học được biểu diễn thành cơ sở dữ liệu tuần tự hay chuỗi học tập (sequential database, learning sequences), mẫu tuần tự được khai thác từ cơ sở dữ liệu này. Mẫu tuần tự (sequential pattern) đại diện cho con đường học tập hay tiến trình học tập cần phải theo. Ví dụ, người học đang học môn  $A$  và mẫu tuần tự là  $B \rightarrow C \rightarrow A \rightarrow E$  thì hệ thích nghi sẽ khuyến nghị họ nên học môn  $E$  tiếp theo; nếu người học mới bắt đầu, hệ thích nghi sẽ khuyến nghị học nên học tuần tự môn  $B$  đến  $C$ ,  $A$  và cuối cùng  $E$ . Vui lòng xem luận văn mục III.3.1.1 để biết thêm chi tiết về kỹ thuật khai thác mẫu tuần tự [Agrawal (Rakesh), Srikant 1995]. Luận văn đề xuất một kỹ thuật đặc biệt, bẻ gãy mẫu tuần tự (đã được khai thác) để thực hiện khuyến nghị.

Giả sử khóa học Java gồm các bài học: cấu trúc dữ liệu ( $d$ ), gói ( $p$ ), khái niệm lớp và đối tượng ( $o$ ), cấu trúc chọn ( $s$ ), máy ảo ( $v$ ), cấu trúc lặp ( $l$ ), cấu trúc điều khiển ( $c$ ), giao diện ( $f$ ). Quá trình truy cập tài liệu được biểu diễn thành cơ sở dữ liệu chuỗi học tập như sau:

Người học	Phiên học	Bài học truy cập	ID	Chuỗi học	Độ dài
1 1	Aug 5 10:20:01 Aug 5 10:26:12	$o$ $f$	1	$\langle of \rangle$	2
2 2 2	Aug 6 08:20:01 Aug 6 14:15:01 Aug 6 15:00:00	$d, p$ $o$ $s, l, c$	2	$\langle (dp)o(slc) \rangle$	6
3	Aug 7 12:30:00	$o, v, c$	3	$\langle (ovc) \rangle$	3
4 4 4	Aug 8 07:14:20 Aug 8 07:40:25 Aug 8 10:17:20	$o$ $s, c$ $f$	4	$\langle o(sc)f \rangle$	4
5	Aug 8 10:26:15	$f$	5	$\langle f \rangle$	1

**Bảng 3.7.** Cơ sở dữ liệu chuỗi học tập

Giả sử mẫu tuần tự  $\langle osc(sc) \rangle$  được khai thác từ cơ sở dữ liệu trên, ta có con đường học tập: “*khái niệm lớp & đối tượng*” – “*cấu trúc chọn*” – “*cấu trúc điều khiển*” – “*cấu trúc chọn, cấu trúc điều khiển*”. Nếu người học đang học bài “*khái niệm lớp & đối tượng*” thì hệ thống nên khuyến nghị họ nên học bài nào tiếp theo. Kỹ thuật bẻ gãy mẫu tuần tự gồm bốn bước:

1. Bẻ gãy mẫu tuần tự  $\langle osc(sc) \rangle$  thành các tập phổ biến  $o, s, c, (sc)$  và xây dựng các chỉnh hợp chập 2 từ các tập phổ biến này, lưu ý thứ tự trong mỗi chỉnh hợp phải tuân theo thứ tự mẫu tuần tự, nghĩa là không có chỉnh hợp như  $\langle so \rangle$ . Ta có 6 chỉnh hợp:  $\langle os \rangle, \langle oc \rangle, \langle o(sc) \rangle, \langle sc \rangle, \langle s(sc) \rangle, \langle c(sc) \rangle$ .
2. Từ các chỉnh hợp chập hai, xây dựng các luật với vế trái và vế phải lần lượt là phần tử thứ nhất và phần tử thứ hai của chỉnh hợp. Ta có 6 luật:  $o \rightarrow s, o \rightarrow c, o \rightarrow sc, s \rightarrow c, s \rightarrow sc, c \rightarrow sc$ .
3. Các luật được sắp xếp theo độ tin cậy của chúng. Độ tin cậy của luật  $(x \rightarrow y)$  được tính bằng độ hỗ trợ của chỉnh hợp  $\langle xy \rangle$  chia cho độ hỗ trợ của tập phổ biến  $x$ . Nếu độ tin cậy một luật nhỏ hơn một ngưỡng thì loại bỏ luật đó. Các luật này được gọi *luật tuần tự*.

Luật tuần tự	Độ tin cậy
$o \rightarrow s$	40%
$o \rightarrow c$	40%
$o \rightarrow sc$	40%
$s \rightarrow c$	0%
$s \rightarrow sc$	0%
$c \rightarrow sc$	0%

**Bảng 3.8.** Luật tuần tự

4. Nếu người học đang chọn bài học  $x$ , trước tiên hệ thống sẽ tìm luật tuần tự nào về trái chứa  $x$  và có độ tin cậy cao nhất, sau đó những bài học ở về phải luật này được giới thiệu cho người học. Ví dụ, trong trường hợp người học đang học bài “*khái niệm lớp & đối tượng*”, hệ thống tìm thấy các luật  $o \rightarrow s$ ,  $o \rightarrow c$  đều chứa  $o$  và có độ tin cậy cao nhất 40% nên các bài học  $s$  (cấu trúc chọn) và  $c$  (cấu trúc điều khiển) được khuyến nghị.

Nhìn chung kỹ thuật bê gãy mẫu tuần tự tạo thành các luật, kết quả rất giống với khai thác luật kết hợp nhưng việc bê gãy hiệu quả hơn nhiều do trong khuyến nghị có đề xuất con đường học tập thông qua mẫu tuần tự.

#### Khai thác sở thích người học qua phân lớp tài liệu

Sở thích là đặc trưng cá nhân tuy không được mô tả trong TLM chuẩn nhưng được xây dựng trong TLM mở rộng vì tính quan trọng của sở thích trong quá trình thích nghi. Thành phần LH có nhiệm vụ phát hiện sở thích cá nhân bằng khai khoáng, luận văn chỉ đề xuất một kỹ thuật khai khoáng là phân lớp tài liệu nhưng chắc chắn còn nhiều kỹ thuật khác, đó là hướng mở của luận văn. Kỹ thuật đề xuất dựa trên hai góc nhìn:

- Chuỗi truy cập (trang web, bài học, sách) được mô hình hóa như các “tài liệu” nên người dùng được gián tiếp mô tả như “tài liệu”.
- Sở thích chính là những lớp của các tài liệu này. Nói cách khác việc phát hiện sở thích người học đồng nhất với phân lớp tài liệu.

Luận văn đề xuất giải thuật phát hiện sở thích người học gồm 4 bước:

1. Các tài liệu học tập (thực sự) được biểu diễn theo mô hình vector, mỗi phần tử là tích của tần số từ khóa ( $tf$ ) và tần số tài liệu nghịch đảo ( $idf$ ). Để tiện dụng, có thể không cần quan tâm tần số tài liệu nghịch đảo.
2. Phân lớp các tài liệu bằng những kỹ thuật như cây quyết định, SVM, mạng neuron. Bất kể sử dụng kỹ thuật nào, các luật phân lớp hay bộ phân lớp phải được rút trích. Vui lòng xem luận văn mục III.3.2 trang 149 để biết thêm các kỹ thuật phân lớp.
3. Lịch sử truy cập (trang web, bài học, sách) của người học được đặc tả như tập các “tài liệu”. Các tập phổ biến cực đại được khai thác từ tập các “tài liệu” này. Mỗi tập phổ biến là một “tài liệu” ưa thích.
4. Áp dụng luật phân lớp (bước 2) vào các “tài liệu” ưa thích (bước 3) để chọn những lớp nào phù hợp nhất với chúng. Những lớp này là sở thích người học.

Ví dụ sau minh họa phương pháp phát hiện sở thích người dùng qua phân lớp tài liệu. Cho tập lớp  $C = \{\text{computer science, math}\}$  và tập từ khóa  $T = \{\text{computer, programming language, algorithm, derivative}\}$  và kho dữ liệu gồm 4 tài liệu  $D = \{\text{doc1.txt, doc2.txt, doc3.txt, doc4.txt, doc5.txt}\}$ . Tần số từ khóa được biểu diễn trong bảng sau:

	<i>computer</i>	<i>programming language</i>	<i>algorithm</i>	<i>derivative</i>	<i>class</i>
<i>doc1.txt</i>	5	3	1	1	computer
<i>doc2.txt</i>	5	5	40	5	math
<i>doc3.txt</i>	20	5	20	55	math
<i>doc4.txt</i>	20	55	5	20	computer
<i>doc5.txt</i>	15	15	4	0.3	math
<i>doc6.txt</i>	35	10	45	10	computer

**Bảng 3.9.** Tần số từ khóa

	<i>computer</i>	<i>programming language</i>	<i>algorithm</i>	<i>derivative</i>	<i>class</i>
<i>doc1.txt</i>	0.5	0.3	0.1	0.1	computer
<i>doc2.txt</i>	0.05	0.05	0.4	0.5	math
<i>doc3.txt</i>	0.2	0.05	0.2	0.55	math
<i>doc4.txt</i>	0.2	0.55	0.05	0.2	computer
<i>doc5.txt</i>	0.15	0.15	0.4	0.3	math
<i>doc6.txt</i>	0.35	0.1	0.45	0.1	computer

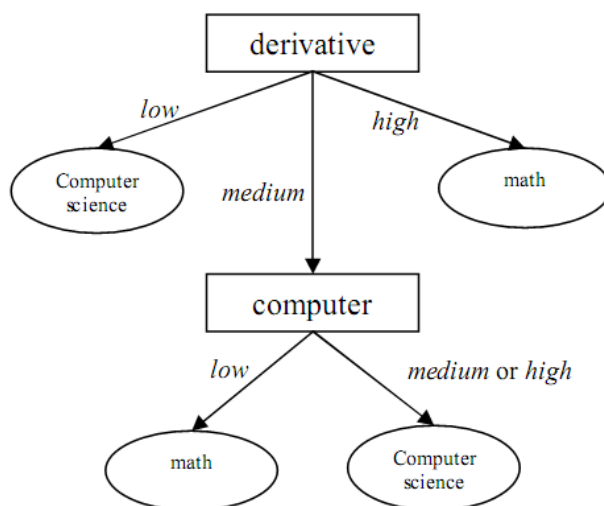
**Bảng 3.10.** Tần số từ khóa được chuẩn hóa

Giả sử phân lớp tài liệu bằng phương pháp cây quyết định, tần số từ khóa cần chuyển từ số thực sang giá trị vô hướng:  $0 \leq \text{tần số} < 0.2$ : *low*;  $0.2 \leq \text{tần số} < 0.5$ : *medium*;  $0.5 \leq \text{tần số}$ : *high*

	<i>computer</i>	<i>programming language</i>	<i>algorithm</i>	<i>derivative</i>	<b>class</b>
<i>doc1.txt</i>	high	medium	low	low	computer
<i>doc2.txt</i>	low	low	medium	high	Math
<i>doc3.txt</i>	medium	low	medium	high	Math
<i>doc4.txt</i>	medium	high	low	medium	computer
<i>doc5.txt</i>	low	low	medium	medium	Math
<i>doc6.txt</i>	medium	low	medium	low	computer

**Bảng 3.11.** Tần số từ khóa với giá trị vô hướng

Cây quyết định được tạo ra từ bảng tần số từ khóa như sau:



**Hình 3.9.** Cây quyết định

Từ cây quyết định, luật phân lớp được thiết lập

<i>Rule 1</i>	If frequency of term “derivative” is <i>low</i> then document belongs to class <i>computer science</i>
<i>Rule 2</i>	If frequency of term “derivative” is <i>medium</i> and frequency of term “computer” is <i>medium</i> or <i>high</i> then document belongs to class <i>computer science</i>
<i>Rule 3</i>	If frequency of term “derivative” is <i>medium</i> and frequency of term “computer” is <i>low</i> then document belongs to class <i>math</i> .
<i>Rule 4</i>	If frequency of term “derivative” is <i>high</i> then document belongs to class <i>math</i>

**Bảng 3.12.** Luật phân lớp

Giả sử trong một thư viện hay trên trang web, người dùng tìm kiếm những cuốn sách, tài liệu... anh ta ưa thích. Cho tập từ khóa {*computer, programming language, algorithm, derivative*}, lược sử tìm kiếm anh ta như sau:

<i>Date</i>	<i>Từ khóa tìm kiếm</i>
Aug 28 10:20:01	computer, programming language, algorithm, derivative
Aug 28 13:00:00	computer, programming language, derivative, algorithm
Aug 29 8:15:01	computer
Aug 30 8:15:06	computer

**Bảng 3.13.** Lược sử tìm kiếm

Tập phổ biến cực đại được khai thác từ lược sử tìm kiếm khi chúng ta xem lược sử tìm kiếm như dữ liệu huấn luyện

$N_o$	Tập phổ biến
1	(computer = 4, programming language = 2, algorithm = 2, derivative = 2)

**Bảng 3.14.** Tập phổ biến cực đại theo mô hình vector cùng với tần số từ khóa

Tập phổ biến cực đại được xem như “tài liệu ưa thích” và lớp của tài liệu này chính là sở thích người dùng. Chuẩn hóa “tài liệu ưa thích”:

$N_o$	Tài liệu ưa thích
1	(computer = 0.4, programming language = 0.2, algorithm = 0.2, derivative = 0.2)

**Bảng 3.15.** “Tài liệu ưa thích” với tần số từ khóa chuẩn hóa

$N_o$	Tài liệu ưa thích
1	(computer = <i>medium</i> , programming language = <i>medium</i> , algorithm = <i>medium</i> , derivative = <i>medium</i> )

**Bảng 3.16.** “Tài liệu ưa thích” với tần số từ khóa vô hướng

Áp dụng luật phân lớp thứ hai trong bảng 3.10, dễ dàng nhận thấy tài liệu ưa thích thuộc lớp “*computer science*” do tần số của cả hai từ khóa “*derivative*” và “*computer*” đều *medium*. Vậy sở thích người học là “*computer science*”.

Nhìn chung phương pháp này dựa trên góc nhìn: người dùng gián tiếp được mô hình hóa như một tài liệu và lớp của tài liệu này chính là sở thích người dùng. Phương pháp kết hợp giữa kỹ thuật phân lớp và kỹ thuật tìm kiếm.

### Xây dựng nhóm hay cộng đồng người học

Mô hình người dùng trong luận văn hướng cá nhân hóa, phù hợp cho những ứng dụng thích nghi mang tính cá nhân nhưng ngoài ra, có những yêu cầu phục vụ cộng đồng hay nhóm người dùng. Trong ngữ cảnh học thích nghi, những người học trong cùng một nhóm sẽ nhận cùng tài liệu học tập, phương pháp dạy, ... Tất nhiên hướng cá nhân hóa hay hướng cộng đồng đều có đặc trưng và nhu cầu riêng, một hệ thích nghi tốt cần quan tâm cả hai. Thích nghi nhóm (cộng đồng) có ưu điểm trong những trường hợp sau:

- Đặc trưng nhóm là đặc trưng chung của tất cả thành viên trong nhóm nên tương đối ổn định, hệ thống dễ dàng hỗ trợ thích nghi.
- Nếu một người dùng mới đăng nhập vào hệ thống, họ được phân vào một nhóm, dễ dàng nhận biết thông tin chung về họ và hệ thống sẽ cung cấp sự thích nghi ngay, không cần cơ chế suy diễn phức tạp.
- Trong học cộng tác, người dùng cần học tập và thảo luận với nhau. Việc này trở nên dễ dàng khi những người cùng chung đặc trưng, sở thích, cách học được gom thành nhóm. Xung đột rất khó xảy ra hoặc dễ dàng được giải quyết.

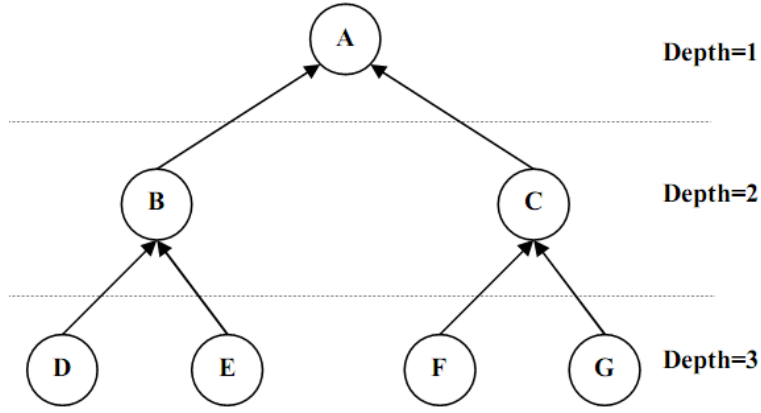
Do sự cần thiết xác định nhóm người học, luận văn đề xuất dùng kỹ thuật gom cụm (clustering) để phân nhóm, đặc biệt gom cụm cho mô hình chồng và mạng Bayesian. Các kỹ thuật gom cụm cùng các độ đo khác biệt và độ đo tương tự, vui lòng xem luận văn mục III.3.3.1 và mục III.3.3.3. Bản tóm tắt này tập trung vào kỹ thuật gom cụm cho mô hình chồng và mạng Bayesian.

Rất nhiều kỹ thuật gom cụm [Han, Kamber 2006] có thể được áp dụng nhưng vấn đề đặt ra “làm sao tính khoảng cách giữa hai mô hình chồng khi bản chất mô hình chồng thuộc dạng đồ thị”. Luận văn đề xuất kỹ thuật đặc biệt dựa vào độ sâu đồ thị.

Giả sử hai mô hình chồng  $U_1=G_1=<V_1,E_1>$  và  $U_2=G_2=<V_2,E_2>$  với  $V_1, V_2$  là các tập nút và  $E_1, E_2$  là các tập cạnh. Đồ thị  $G_1, G_2$  dưới dạng cây và mỗi cạnh biểu diễn quan hệ điều kiện tiên quyết, nếu có một cạnh từ nút  $A$  đến nút  $B$  thì người dùng phải học khái niệm  $A$  trước khi học khái niệm  $B$ .

$V_i$  được xem như một vector,  $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$ , lưu ý mỗi  $v_{ij}$  là nút được mô tả bởi một số nguyên biểu diễn tri thức người học.

Gọi  $depth(v_{ij})$  là độ sâu của nút  $j$  của đồ thị  $V_i$ . Độ sâu nút gốc là  $1$ .



**Hình 3.10.** Đồ thị và độ sâu

Độ khác biệt hay khoảng cách giữa hai đồ thị  $G_1$  và  $G_2$  được tính như sau:

$$dissim(G_1, G_2) = dis\ tan\ ce(G_1, G_2) = \sum_{j=1}^n \left| \frac{v_{1j} - v_{2j}}{depth(v_{1j})} \right|$$

Ý nghĩa công thức: “những khái niệm học ở mức cao là tổng kết của những khái niệm căn bản ở mức thấp”. Ví dụ, với ba đồ thị  $G_1$ ,  $G_2$ ,  $G_3$  có cùng cấu trúc như hình 3.10, giá trị các nút như sau:

	A	B	C	D	E	F	G
$G_1$	2	1	1	0	3	2	1
$G_2$	1	1	0	1	4	5	4
$G_3$	2	1	1	1	4	5	4

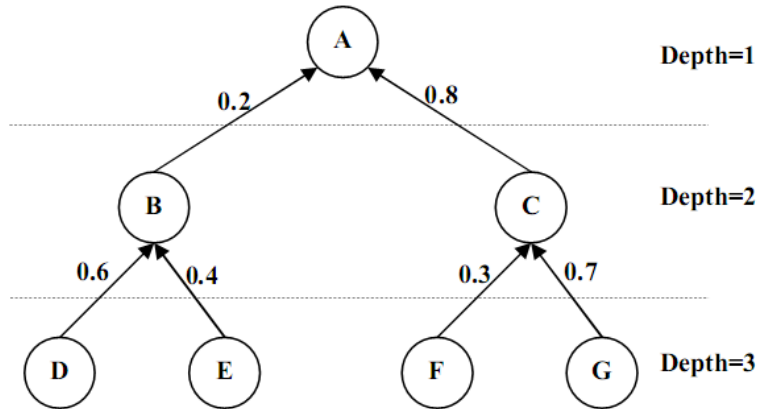
**Bảng 3.17.** Giá trị các nút của các đồ thị ( $G_1$ ,  $G_2$ ,  $G_3$ )

Độ khác biệt hay khoảng cách giữa đồ thị  $G_3$  và  $G_1$ ,  $G_2$ :

$$dissim(G_1, G_3) = \left| \frac{2-2}{1} \right| + \left| \frac{1-1}{2} \right| + \left| \frac{1-1}{2} \right| + \left| \frac{0-1}{3} \right| + \left| \frac{3-4}{3} \right| + \left| \frac{2-5}{3} \right| + \left| \frac{1-4}{3} \right| = 2.66$$

$$dissim(G_2, G_3) = \left| \frac{1-2}{1} \right| + \left| \frac{1-1}{2} \right| + \left| \frac{0-1}{2} \right| + \left| \frac{1-1}{3} \right| + \left| \frac{4-4}{3} \right| + \left| \frac{5-5}{3} \right| + \left| \frac{4-4}{3} \right| = 1.5$$

Vậy  $G_2$  giống  $G_3$  hơn  $G_1$ . Trong trường hợp mỗi cạnh có đánh trọng số, đồ thị sẽ như sau:



**Hình 3.11.** Đồ thị có đánh trọng số

Gọi  $weight(v_{ij})$  là trọng số cạnh từ nút  $j$  đến nút cha trong đồ thị  $G_i$ , công thức tính độ khác biệt (hay khoảng cách) được tính lại như sau:

$$dissim(G_1, G_2) = dis\ tan\ ce(G_1, G_2) = \sum_{j=1}^n \left| \frac{v_{1j} - v_{2j}}{depth(v_{1j})} * weight(v_{1j}) \right|$$

Lưu ý, trọng số nút gốc là 1. Ví dụ, độ khác biệt giữa 3 đồ thị  $G_1, G_2, G_3$  trong bảng 3.17 được tính lại như sau:

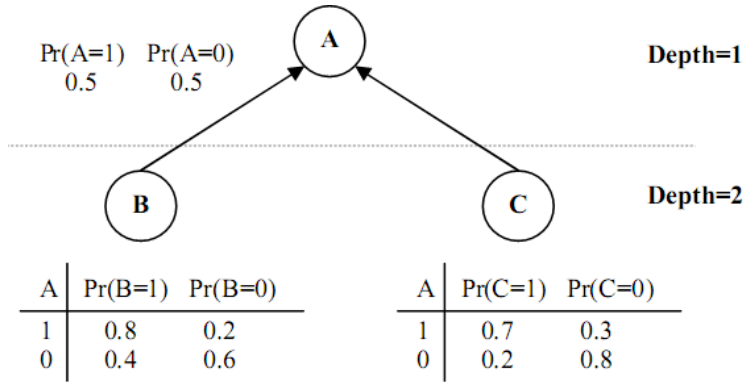
$$dissim(G_1, G_3) =$$

$$\left| \frac{2-2}{1} \right| + \left| \frac{1-1}{2} * 0.2 \right| + \left| \frac{1-1}{2} * 0.8 \right| + \left| \frac{0-1}{3} * 0.6 \right| + \left| \frac{3-4}{3} * 0.4 \right| + \left| \frac{2-5}{3} * 0.3 \right| + \left| \frac{1-4}{3} * 0.7 \right| = 1.33$$

$$dissim(G_2, G_3) =$$

$$\left| \frac{1-2}{1} \right| + \left| \frac{1-1}{2} * 0.2 \right| + \left| \frac{0-1}{2} * 0.8 \right| + \left| \frac{1-1}{3} * 0.6 \right| + \left| \frac{4-4}{3} * 0.4 \right| + \left| \frac{5-5}{3} * 0.3 \right| + \left| \frac{4-4}{3} * 0.7 \right| = 1.4$$

Bây giờ  $G_1$  giống  $G_3$  hơn  $G_2$ . Trong trường hợp mạng Bayesian, công thức tính khoảng cách phức tạp hơn. Mạng Bayesian vẫn là đồ thị, độ sâu vẫn định nghĩa như cũ, nhưng trọng số cạnh được thay bằng bảng xác suất điều kiện (CPT), giá trị tại mỗi nút được thay bằng xác suất hậu biên (marginal posterior probability).



**Hình 3.12.** Mạng Bayesian, CPT và độ sâu

Gọi  $Pr(v_{ij})$  là xác suất hậu biên của nút  $j$  trong mạng Bayesian  $G_i$ , công thức tính độ khác biệt hay khoảng cách giữa hai mạng Bayesian  $G_1$  và  $G_2$  như sau:

$$dissim(G_1, G_2) = dis\ tan\ ce(G_1, G_2) = \sum_{j=1}^n \left| \frac{\Pr(v_{1j}) - \Pr(v_{2j})}{depth(v_{1j})} \right|$$

Vui lòng xem luận văn mục III.1.1.1 về suy diễn trong mạng Bayesian và xác suất hậu biên.



## 4. Đánh giá môn hình học thích nghi

Như đề cập, hệ thống mô hình hóa là trọng tâm của hệ thích nghi, có rất nhiều lý thuyết và phương pháp bao gồm cách tiếp cận trong luận văn nhằm xây dựng hệ thích nghi và hệ mô hình hóa. Vì mỗi phương pháp đều có ưu nhược điểm nên rất hữu ích khi chúng ta đề xuất những cách đánh giá mỗi phương pháp để thẩm định những mô hình tối ưu trong những tình huống nhất định do mỗi mô hình, mỗi phương pháp phù hợp với điều kiện và ngữ cảnh nhất định, ví dụ, học qua trang web rất khác với các khóa học với sự hỗ trợ của hệ thống mạng. Luận văn tập trung vào đánh giá hệ thống học thích nghi với sự hỗ trợ của hệ mô hình hóa trong ngữ cảnh e-learning hay học từ xa và không có sự tách bạch giữa hệ thích nghi và hệ mô hình hóa trong tiến trình đánh giá. Chúng ta sẽ đánh giá kết hợp hệ thích nghi và hệ mô hình hóa trong một mô hình tích hợp, được gọi là mô hình học thích nghi. Theo đó, luận văn đề xuất phần đánh giá mô hình học thích nghi gồm hai mục tiêu:

- Trước tiên, những tiêu chuẩn đánh giá được đề xuất.
- Sau đó, những kịch bản đánh giá dựa trên những tiêu chuẩn trên được đề xuất nhằm thực hiện tác vụ đánh giá trong những tình huống xác định.

Luận văn đề xuất ba tiêu chuẩn đánh giá:

- Tiêu chuẩn  $\alpha$  được gọi là tiêu chuẩn hệ thống nhằm đánh giá hệ học thích nghi hoạt động như thế nào nếu có hoặc không có sự hỗ trợ của hệ mô hình hóa người học. Ví dụ, khi trình chủ mô hình hóa xây dựng mô hình người dùng bằng mạng Bayesian, tiêu chuẩn  $\alpha$  đo lường hiệu năng hệ thống với sự hỗ trợ hoặc không hỗ trợ của mạng Bayesian. Nhìn chung, tiêu chuẩn  $\alpha$  trả lời hai câu hỏi sau:
  - o Hiệu năng tiến trình thích nghi như thế nào với sự hỗ trợ hoặc không hỗ trợ của hệ thống mô hình hóa.
  - o Tri thức người dùng có được ước lượng chính xác hay không với sự hỗ trợ của mô hình người dùng (user model) mà cụ thể là mạng Bayesian.
- Tiêu chuẩn  $\beta$  được gọi là tiêu chuẩn học thuật nhằm đánh giá mức độ hỗ trợ người học của hệ mô hình hóa. Tiêu chuẩn này đo lường kết quả học tập của người học. Tiêu chuẩn  $\beta$  càng cao, kết quả học tập càng tốt.
- Tiêu chuẩn  $\gamma$  được gọi là tiêu chuẩn thích nghi hay tiêu chuẩn đáp ứng nhu cầu, đo lường chất lượng của chức năng học thích nghi với sự hỗ trợ của hệ mô hình hóa. Sau khi những phản hồi hay nhận xét của người học đối với hệ thích nghi được thu thập và phân tích, tiêu chuẩn  $\gamma$  được xác định dựa trên những phản hồi này để đo lường mức độ hài lòng của người học đối với hệ thích nghi. Tiêu chuẩn  $\gamma$  càng cao, chất lượng chức năng thích nghi càng tốt.

Tiêu chuẩn  $\alpha$  được tính bằng hai phương pháp: kiểm định giả thuyết hay mô hình hồi quy. Với phương pháp kiểm định giả thuyết, người học được phân thành hai nhóm với sự hỗ trợ hoặc không hỗ trợ của hệ mô hình hóa (cụ thể, hệ Zebra) và phương sai (variance) trên mỗi nhóm được xác định. Giả thuyết vô hiệu được phát biểu rằng hai phương sai trên hai nhóm bằng nhau. Bác bỏ giả thuyết vô hiệu tương ứng với tiêu chuẩn  $\alpha$  đạt giá trị *đúng* (true) xác định sự ưu việt của hệ mô hình hóa. Vui lòng xem mục IV.2.1.1 trong luận văn để biết thêm chi tiết về cách tính tiêu chuẩn  $\alpha$  bằng phương pháp kiểm định giả thuyết. Theo phương pháp mô hình hồi quy, một cách suy diễn tốt nếu giá trị dự đoán, cụ thể là tri thức toàn thể mà người học đạt được sẽ tương quan với tất cả tri thức từng phần của người học trên những giai đoạn học tập theo một phương trình hay một hàm. Tiến trình xác định tiêu chuẩn  $\alpha$  theo phương pháp hồi quy gồm bốn bước:

1. Hàm hồi quy tri thức người học được xác định cho từng nhóm người học với sự hỗ trợ hoặc không hỗ trợ của hệ mô hình hóa.
2. Ước lượng tri thức người học dựa trên hàm hồi quy.
3. Độ lỗi dự đoán được tính theo độ lệch giữa tri thức ước lượng và tri thức thực tế của người học.
4. Tiêu chuẩn  $\alpha$  là nghịch đảo tổng bình phương tất cả độ lỗi dự đoán.

Vui lòng xem mục IV.2.1.1 trong luận văn để biết thêm chi tiết về cách tính tiêu chuẩn  $\alpha$  bằng phương pháp hồi quy. Giả sử hai nhóm người học với sự hỗ trợ và không hỗ trợ của hệ mô hình hóa, ví dụ hệ Zebra, tiêu chuẩn  $\beta$  cho từng nhóm là xác suất phân bố tích lũy của các người học trong nhóm đó. Lưu ý rằng tri thức người học được giả định tuân theo phân bố chuẩn và do đó xác suất phân bố tích lũy được tính theo trung bình mẫu và phương sai mẫu của vector tri thức trên từng nhóm người học.

Giả sử một bảng câu hỏi gồm  $n$  câu hỏi được xây dựng bởi chuyên gia, người học trả lời và đánh giá trên các câu hỏi và giá trị đánh giá biểu hiện sự hài lòng và không hài lòng của người học. Theo cách đơn giản nhất, tiêu chuẩn  $\gamma$  được định nghĩa là tỉ lệ số lượng người dùng hài lòng trên tổng số người dùng. Nâng cao hơn, giá trị đánh giá dao động trong một khoảng, ví dụ  $[0...5]$  với 0 là ít hài lòng nhất và 5 là hài lòng nhất, khi đó tất cả giá trị đánh giá tạo thành ma trận đánh giá. Tiến trình xác định tiêu chuẩn  $\gamma$  theo phương thức nâng cao gồm ba bước:

1. Đầu tiên, ma trận đánh giá được “co” lại bằng cách chiếu nó lên các vector riêng.
2. Sau đó, giá trị trung bình  $\mu_i$  của mỗi cột được tính.
3. Cuối cùng, vector trung bình của ma trận đánh giá được tạo thành từ tất cả các giá trị trung bình  $\mu_i$ .

Nhìn chung, tiêu chuẩn  $\gamma$  càng cao, chất lượng thích nghi càng tốt. Vui lòng xem mục IV.2.1.1 để biết thêm chi tiết về cách tính tiêu chuẩn  $\gamma$ .

Kịch bản đánh giá là ví dụ minh họa cách tính và áp dụng những tiêu chuẩn đánh giá nêu trên vào việc đánh giá mô hình học thích nghi. Học điện tử (e-learning) không thể thay thế phương pháp giảng dạy trực tiếp truyền thống và nó nên hoạt động song song và hỗ trợ phương pháp giảng dạy truyền thống. Do đó kịch bản đánh giá sẽ so sánh giữa phương pháp giảng dạy trực tiếp và phương pháp dạy từ xa. Kịch bản này sẽ chia thành ba hồi chính đồng thời giáo viên và sinh viên đóng vai trò như những diễn viên.

1. *Hồi học tập*: Giáo viên giảng dạy và sinh viên học theo cách học trực tiếp và học qua trang web. Giả sử sinh viên được chia thành ba nhóm: nhóm *A* học trực tiếp, nhóm *B* học qua trang web và nhóm *C* học qua trang web với sự hỗ trợ của mô hình người dùng, cụ thể là mạng Bayesian.
2. *Hồi thông tin phản hồi*: sinh viên phản hồi những nhận xét đến giáo viên và giáo viên thu thập và phân tích những phản hồi này.
3. *Hồi đánh giá* được thực hiện bởi giáo viên; theo đó, các tiêu chuẩn  $\alpha$ ,  $\beta$  and  $\gamma$  được xác định qua dữ liệu thu thập từ hai hồi trên. Chất lượng mô hình học thích nghi được xác định qua những tiêu chuẩn này.

Hồi học tập gồm 5 cảnh:

1. Giáo viên xây dựng chương trình giảng dạy và thiết lập trang web học thích nghi với sự hỗ trợ và không hỗ trợ của hệ mô hình hóa.
2. Giáo viên dạy sinh viên trong các nhóm *A*, *B*, *C* bằng cách giảng dạy trực tiếp.
3. Sinh viên trong nhóm *B* và *C* vào trang web và tự học thêm. Giáo viên theo dõi hoạt động học tập và cung cấp những thông tin cần thiết.
4. Sinh viên trong các nhóm *A*, *B*, *C* làm bài kiểm tra và bài tập qua trang web.
5. Giáo viên đánh giá sinh viên dựa trên kết quả học tập.

Vai trò giáo viên trong hồi học tập bao gồm:

- Giảng dạy trực tiếp theo cách truyền thống
- Xây dựng mô hình tri thức và khởi tạo các tài nguyên web cho mô hình tri thức như những bài giảng siêu văn bản (html), bài kiểm tra, bài tập.
- Xây dựng mô hình người học, ví dụ, tạo mạng Bayesian và các trọng số tri thức.
- Cài đặt hệ thống mô hình hóa người học và trang web học thích nghi.
- Theo dõi quá trình học tập của sinh viên.
- Gởi kết quả học tập đến sinh viên.

Vai trò sinh viên trong hồi học tập bao gồm:

- Sinh viên trong nhóm *A*, *B*, *C* đến lớp và học trực tiếp với giáo viên.
- Sinh viên trong nhóm *B* và *C* tự học thêm qua trang web học thích nghi. Lưu ý những tài liệu học tập trên trang web thích ứng cho từng sinh viên theo tri thức và đặc trưng cá nhân của họ.
- Sinh viên trong các nhóm *A*, *B* và *C* làm bài tập và bài kiểm tra qua trang web.

Hồi thông tin phản hồi gồm 3 cảnh:

1. Giáo viên tạo bảng câu hỏi để khảo sát những nhận xét của sinh viên về trang web học thích nghi và chương trình giảng dạy với các mức độ rất hài lòng, hài lòng và không hài lòng.
2. Sinh viên trả lời và đánh giá trực tuyến trên các câu hỏi
3. Giáo viên thu thập và phân tích các thông tin phản hồi của sinh viên.

Hồi đánh giá gồm 2 cảnh:

1. Giáo viên tính toán ba tiêu chuẩn đánh giá dựa trên những phản hồi và kết quả học tập của sinh viên.
2. Giáo viên xác định chất lượng phương thức giảng dạy trực tiếp và phương thức giảng dạy qua trang web với sự hỗ trợ hoặc không hỗ trợ của hệ mô hình hóa.

Nhìn chung, luận văn đề xuất ba tiêu chuẩn đánh giá bao gồm tiêu chuẩn hệ thống  $\alpha$ , tiêu chuẩn học thuật  $\beta$  và tiêu chuẩn thích nghi  $\gamma$ . Hai trong số ba tiêu chuẩn trên bao gồm  $\alpha$  and  $\beta$  đánh giá tri thức người học xác định rằng mục tiêu chính của giáo dục là nâng cao tri thức người học và việc đánh giá mô hình học thích nghi tập trung vào mục tiêu này mặc dù tiến trình thích nghi và cá nhân hóa là những chủ đề quan trọng trong học thích nghi. Chúng ta có thể nhận thấy rằng mục tiêu chính của giáo dục không đi xa khỏi việc nâng cao tri thức nhân loại. Kịch bản đánh giá, một ví dụ minh họa cách xác định ba tiêu chuẩn trên, chỉ ra rằng học tập là quá trình diễn ra trong suốt đời đối với mỗi người và vì thế những lớp học và khóa học chỉ là những đoạn phim ngắn trong tiến trình học tập suốt đời đó. Cả sinh viên và giáo viên đều là những diễn viên và vai trò của họ có thể thay đổi cho nhau, ví dụ, giảng dạy là cách tốt nhất để học và sinh viên là người thầy tốt nhất cho những người thầy.

## 5. Kết luận và hướng phát triển

Tóm lại, luận văn hướng đến xây dựng một hệ mô hình hóa người dùng trong ngữ cảnh học thích nghi. Trước tiên mô hình người dùng được gọi *TLM* gồm 3 thành con kết thành tam giác: tri thức (*K*), cách học (*LS*), lược sử học (*LH*).

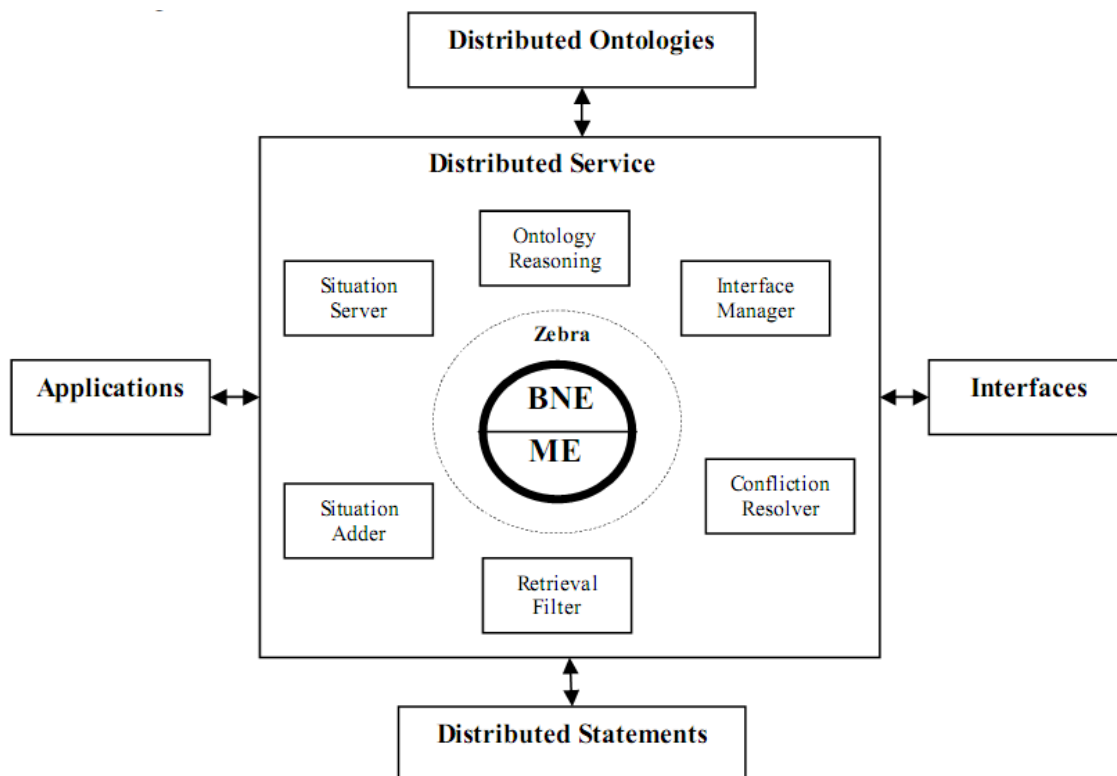
- Thành phần *K*: kết hợp mô hình chông và mạng Bayesian cùng những đặc tả trọng số. Thành phần *K* được tiến hóa thông qua các kỹ thuật: học tham số với EM, học tham số với MLE, học cấu trúc qua mạng Bayesian động.
- Thành phần *LS*: áp dụng mô hình Markov ẩn vào các lý thuyết tâm lý để phát hiện *LS* của người học.
- Thành phần *LH*: quan trọng nhất, lưu trữ hoạt động học tập của người học, định dạng thông thường là XML. Thành phần *LH* dùng xây dựng thành phần *K* và *LS* cùng với khai thác các đặc trưng mở rộng như nhóm người học, sở thích người học.

Hệ mô hình hóa Zebra quản trị *TLM* và thực hiện cơ chế suy diễn thông qua hai guồng máy:

- Guồng máy khai khoáng *ME*: quản trị toàn *TLM* nhưng tập trung vào thành phần *LH*, chủ yếu sử dụng các kỹ thuật khai khoáng.
- Guồng máy mạng xác tín *BNE*: quản trị và thực hiện cơ chế suy diễn trên thành phần *K* và *LS* qua mạng xác tín như mạng Bayesian và mô hình Markov.

Hệ Zebra trong tương lai sẽ hỗ trợ hai hướng:

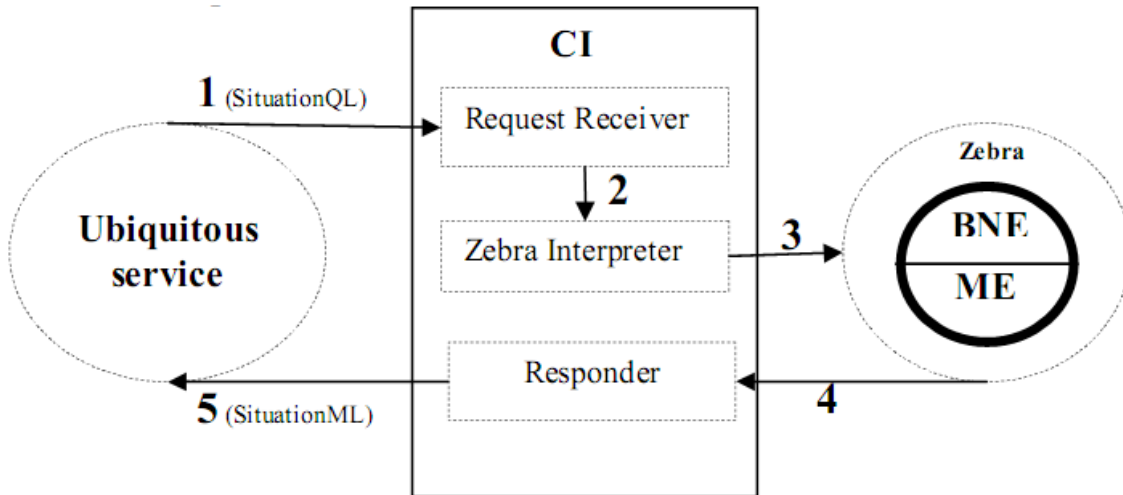
- Kiểm tra thích nghi (Computerized Adaptive Test – CAT) [Baker 2001]: mỗi người học sẽ nhận những bài kiểm tra khác nhau và mô hình kiểm định cải thiện với những điều kiện dừng tối ưu để đánh giá tốt nhất tri thức người học.
- Hỗ trợ môi trường ubiquitous [Heckmann 2005]: hệ mô hình hóa tương tác với người dùng mọi lúc mọi nơi và hoàn toàn trong suốt, nghĩa là người dùng không cần quan tâm đến cơ sở hạ tầng lẫn kiến trúc của hệ mô hình hóa, họ chỉ tận dụng những lợi ích từ dịch vụ ubiquitous. Zebra sẽ tích hợp vào môi trường ubiquitous theo kiến trúc giả định như sau:



**Hình 4.1.** Tích hợp Zebra vào môi trường Ubiquitous  
(Vui lòng xem luận văn mục IV.2.5 để biết thêm chi tiết)

Sự tương tác giữa Zebra và dịch vụ ubiquitous thông qua giao tiếp *CI* (xem phần 2. Kiến trúc) qua 5 bước:

1. Dịch vụ Ubiquitous gửi yêu cầu truy vấn hay cập nhật *TLM* cho *CI* dưới dạng *SituationQL*
2. *CI* diễn dịch *SituationQL* thành cấu trúc dữ liệu nội tại, sao cho hai guồng máy *ME* và *BNE* hiểu
3. Cấu trúc dữ liệu nội tại đó gửi cho Zebra, cụ thể *ME* và *BNE*
4. Zebra thực hiện thao tác suy diễn và cập nhật và gửi kết quả về cho *CI*
5. *CI* diễn dịch kết quả thành định dạng *SituationML* và gửi lại cho dịch vụ Ubiquitous



**Hình 4.2.** Tương tác giữa Zebra vào dịch vụ ubiquitous  
(Vui lòng xem luận văn mục IV.2.5)

## Tài liệu tham khảo

1. [Agrawal (Rakesh), Srikant 1995]. R. Agrawal and R. Srikant. Mining sequential patterns. In Proc. 1995 Int. Conf. Data Engineering (ICDE95), pp.314, Taipei, Taiwan, Mar.1995.
2. [Baker 2001]. Frank B. Baker. The basics of item response theory. Published by the ERIC Clearinghouse on Assessment Evaluation 2001.
3. [De Bra, Calvi 1998]. Paul De Bra, Licia Calvi. AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, vol. 4, pp. 115-139, Taylor Graham Publishers, 1998.
4. [De Bra, Smits, Stash 2006]. Paul De Bra, David Smits, Natalia Stash. The Design of AHA!. Proceedings of the ACM Hypertext Conference, Odense, Denmark, August 23-25, 2006 pp. 133.
5. [Dellaert 2002]. Frank Dellaert. The Expectation Maximization Algorithm. College of Computing, Georgia Institute of Technology. Technical Report number GIT-GVU-02-20, February 2002.
6. [Dugad, Desai 1996]. R. Dugad, U. B. Desai. A tutorial on Hidden Markov models. Signal Processing and Artificial Neural Networks Laboratory, Dept of Electrical Engineering, Indian Institute of Technology, Bombay Technical Report No.: SPANN-96.1, 1996.
7. [Fink 2004]. Josef Fink, Gutachter: Prof. Dr. Alfred Kobsa, Prof. Dr. Rainer Unland. User Modeling Servers - Requirements, Design, and Evaluation. Universität Duisburg-Essen, Standort Essen, Fachbereich 6 Mathematik. Amsterdam, Netherlands: IOS Press (2004).
8. [Fröschl 2005]. Christoph Fröschl. User Modeling and User Profiling in Adaptive E-learning Systems. Masters Thesis at Graz University of Technology 2005. An approach for a service-based personalization solution for the research project AdeLE (Adaptive e-Learning with Eye-Tracking).
9. [Han, Kamber 2006]. Jiawei Han and Micheline Kamber. Data Mining: Concepts and Techniques. Second Edition. © 2006 by Elsevier Inc.

10. [Hauger, Köck 2007]. David Hauger and Mirjam Köck. State of the Art of Adaptivity in E-Learning Platforms. Proceedings of the 15th Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS 2007). ISBN: 978-3-86010-907-6 (2007).
11. [Heckmann 2005]. Dominikus Heckmann. Ubiquitous User Modeling. PhD Thesis at Universität des Saarlandes. Volume 297 Dissertationen zur Künstlichen Intelligenz.
12. [Kobsa 2006]. Alfred Kobsa. Generic User Modeling Systems. User Modeling and User-Adapted Interaction 2006 (UMUAI-2006).
13. [Kolb 1999]. D. A. Kolb. The Kolb Learning Style Inventory. Version 3. Boston: Hay Group, 1999.
14. [Neapolitan 2003]. Richard E. Neapolitan. Learning Bayesian Networks. Northeastern Illinois University Chicago, Illinois 2003.
15. [Stash, Cristea, De Bra 2005]. Natalia Stash, Alexandra Cristea, Paul De Bra. Explicit Intelligence in Adaptive Hypermedia: Generic Adaptation Languages for Learning Preferences and Styles. In Proceedings of HT2005 CIAH Workshop, Salzburg, Austria, 2005.
16. [Wolf 2003]. Christian Wolf. iWeaver: Towards "Learning Style"-based e-Learning in Computer Science Education. Australasian Computing Education Conference (ACE2003), Adelaide, Australia. Conferences in Research and Practice in Information Technology, Vol.20.
17. [Wolf 2007]. Christian Wolf. Construction of an Adaptive E-learning Environment to Address Learning Styles and an Investigation of the Effect of Media Choice. PhD by Project: RMIT University - Design and Social Context Portfolio, School of Education (Research) - Melbourne - Australia (January 2007).