

## НЯКОИ ИДЕИ ЗА ИЗПОЛЗВАНЕ НА ИНТЕРВАЛНА ТЕМПОРАЛНА ЛОГИКА И ПОЛИТИКИ В СИСТЕМА ЗА ЕЛЕКТРОННО ОБУЧЕНИЕ

Тодорка Глушкова

СОУ "Христо Смирненски" гр. Брезово, обл. Пловдив  
todorka.glushkova@gmail.com

**Абстракт:** В статията се обсъждат някои идеи за персонализация на учебния процес и адаптивност на система за електронно обучение към потребителските знания, предпочитания и цели, чрез използване на интервална темпорална логика и политики.

**Ключови думи:** eLearning, DeLC, SCORM, ITL, policy, user modeling, adaptability

Съвременното общество е заинтересовано от формиране на креативни личности със силно развити способности за мобилност, адаптация и самоусъвършенстване. Това поставя качествено нови изисквания към образованието като цяло, свързани както с преодоляване на външната или недостатъчна мотивация за учене, така и с развитие на умения за извличане на информация и учене през целия живот. Едно витално решение за средното училище е приложението на т.нар. „реформаторска педагогика“, разглеждана у нас от Бижков, Андреев, Петров и други видни педагози, [1], [3]. Създаването на системи за електронно обучение и приложението им в различна степен в класната стая и при самоподготовката на учениците е едно удачно приложение на реформаторската педагогика в динамично променящите се условия на информационния 21-ви век [3], [4]. Съществуват различни дефиниции на понятието електронно обучение, но при всички тях персонализацията и адаптивността са ключови характеристики [7]. В статията ще разгледаме някои възможности за реализиране на индивидуализация и динамична адаптивност на система за електронно обучение към персоналните характеристики, знания и изисквания на конкретния ученик, чрез използване на интервална темпорална логика и политики.

Сломан дефинира политиките като множество от правила, активиращи различни състояния и действия, в зависимост от поведението на потребителите или текущото състояние на системата [12]. Съществуват различни техники за формализиране на политиките - чрез използване на среда за графично моделиране (LasCoPolicyLanguage [10]); чрез използване на обектно-ориентираните възможности за дефиниране на политиките (Ponder Policy Framework [9]) и др. Ние ще използваме възможностите, които предоставя интервалната темпорална логика (ITL [11]), тъй като тя надгражда класическата логика от първи ред и позволява описание на динамичните процеси в хода на тяхното изпълнение. Тя е гъвкава нотация за обработка на събития, променящи се в определен времеви интервал, позволява последователно и паралелно композиране като използва добре дефинирана математическа система за доказване. ITL включва четири компонента - логика от първи ред, темпорални конструкции, състояния и интервали. Логиката от първи ред борави с променливи, константи, функции и предикати. Ако желаем да опишем динамични процеси е нужно към познатите ни логически оператори да добавим и темпоралните конструкции skip, chop и chopstar. Състоянията са конкретните придавания на стойности на наблюдаваните променливи, а интервалите са последователности от състояния.

Да разгледаме трите множества: S-множество на потребителите; O-множество на достъпните ресурси и A-множество на действията, които могат да се извършат с тези ресурси. Тогава идентификацията на потребителя в системата можем да представим като една от булевите променливи:

- **autho+(s,o,a)** – положителна идентификация на потребителя **s** с право да ползва ресурса **o** като извършва действие-**a**. *Например: autho+(Иван, Урок1, четене) или autho+(Иван, Тест1, пише)*
- **autho-(s,o,a)** – отрицателна идентификация – отказ на потребителя **s** да ползва ресурса **o** като извършва действие-**a**. *Например: autho-(Иван, Урок1, пише)*

При първоначалното стартиране на системата тези променливи имат стойност по подразбиране *false*. Математическият модел на *autho* е матрица с 3 стълба- потребител, обект, действие и n-брой редове за всички потребители в системата.

Достъпът до ресурсите ще бъде позволен, ако са удовлетворени определени „условие-действие“ правила от вида:  $f \rightarrow w$ , т.е. *f* винаги се следва от *w* във финалното състояние на наблюдавания подинтервал. Според тази дефиниция правилата за идентификация приемат следния вид: **f→autho+(s,o,a)** - правило за положителна идентификация и **f→autho-(s,o,a)** - правило за негативна идентификация. *Например: Ако първоначално достъпа е отказан, но в следващия момент е разрешен за продължителност 10 времеви единици, то: (autho-(s,o,a)∧skip; autho+(s,o,a)∧len<=10) → autho+(s,o,a)*. Ако двама потребители **s** и **m** са в група и единият получи достъп, то втория също получава:  $In(s,m) \wedge autho+(s,o,a) \rightarrow autho+(m,o,a)$ .

Правилата за достъп определят дали конкретният потребител има право на достъп до конкретния учебен ресурс или услуга. За да се реализира самия достъп, управлението се предава на правилата за изпълнение, които имат следния по-общ вид:  $f \rightarrow \text{autho}(s,o,a)$ . Възможни са две алтернативи при достъпа: отворен достъп с малка сигурност, при който ако достъпа не е забранен, то той се разрешава:  $\rightarrow \text{autho}(s,o,a) \rightarrow \text{autho}(s,o,a)$  или ограничен достъп, при който се проверява дали достъпът е разрешен и дали не е междуременно забранен. Т.е.  $(\text{autho}^+(s,o,a) \wedge \neg \text{autho}^-(s,o,a)) \rightarrow \text{autho}(s,o,a)$ .

Друг начин за достъп до порталните ресурси е делегирането на права на неотризиран потребител. Например: Учител делегира право на достъп за четене до негов урок на другите потребители:  $\text{Teacher}(s,lesson) \rightarrow \text{candeleg}(s,lesson,read)$ . Правилото за делегиран достъп, което авторът A1 дава на учителят T2 да нанесе корекции в урока Урок1 ще бъде:

$(\text{autho}(A1, \text{Урок1}, \text{Пише}) \wedge \text{candeleg}(A1, T1, \text{Урок1}, \text{Пише})) \rightarrow \text{autho}(T1, \text{Урок1}, \text{Пише})$

Политиката P е колекция от правила от вида:  $P \equiv (w \wedge (\Delta_i) \wedge \text{fin } w')$ , където w е началното състояние, а w' – крайното, а  $\Delta_i$  е i-степенна конюнкция на междинните състояния. Например: За автор на урок(автор), учител, провеждащ този урок(teacher) и ученик(student):

$P1 \equiv ((\text{avtor}(s,lesson) \rightarrow \text{autho}^+(s,lesson,read)) \wedge (\text{avtor}(s,lesson) \rightarrow \text{autho}^+(s,lesson,write))) \wedge (\text{teacher}(s,lesson) \rightarrow \text{autho}^+(s,lesson,read)) \wedge (\text{teacher}(s,lesson) \rightarrow \text{autho}^-(s,lesson,write)) \wedge (\text{stu}(s,lesson) \rightarrow \text{autho}^+(s,lesson,read)) \wedge (\text{autho}^+(s,lesson,a) \wedge \neg \text{autho}^-(s,lesson,a)) \rightarrow \text{autho}(s,lesson,a)$

Първата стъпка към създаване на система за електронно обучение е стандартизиране на основните процеси и в частност на електронните уроци. Разработваният образователен портал при СОУ»Христо Смирненски» гр. Брезово се базира на теоретичната рамка на системата за електронно и дистанционно обучение DeLC и притежава механизми за приложение на стандарта SCORM<sup>1</sup> [5]. В статията ще споделим един модел за формализиране на избора и обучението със SCORM-базирани уроци, чрез използване на ITL и политики.

Авторите създават електронните уроци в специализирана SCORM-съвместима авторска среда [13], след което ги публикуват в образователния портал. Считаме за удачно използването на шаблони, чрез които да се формализират съдържанието, структурата и обучителния сценарий на съответния урок. Ръководството за създаване на SCORM-базирани електронни уроци Best Practices Guide for Content Developers на Carnegie Mellon University [14], предлага десет базови шаблона и пет модела, които описват разнообразни учебни ситуации и отговарят на различни педагогически изисквания и цели. Ако параметризираме тези шаблони [6], ще получим множество от различни варианти на всеки един от тях, което ще удовлетворява по-пълно разнообразието от изисквания и цели на учителите и учениците. Така по всяка тема могат да се разработят варианти от уроци с различна степен на трудност и обхват.

Формализирането на достъпа на учениците до тези уроци, ще реализираме като първоначално ги регистрираме в специален **Lesson**-регистър, в който ще се съхраняват някои техни спецификации като:

- **Info**- заглавие на урока, учебен предмет, автор и пр. характеристики, които се поддържат от SCORM.
- **Subdomain** -матрица с понятията, които ще се изучават в урока и степента на тяхното изучаване. Subdomain(concept, m), където m=1,2,3 като: **1**- отговаря на ниска степен на усвояване на понятието (задължителен минимум, според ДООИ<sup>2</sup>); **2**- за добро ниво и **3** – за високо ниво.
- **Num\_Class** (класът, за който е предназначен урока) -цяло число от 1 до 12.
- **Form\_of\_teaching** (форма на обучение) –цяло число: 1- присъствена; 2- неприсъствена
- **Less\_Status** (статус на урока)- цяло число от 1 до 4: **4**-свободен за ползване от всички потребители в този и останалите портали в DeLC-образователната мрежа; **3**-свободен за ползване само от учениците и учителите в портала; **2**-разрешен за ползване само за определени потребители; **1**-недостъпен за други потребители, освен за автора му.
- **Didactic\_aims** (педагогически цели, според таксономията на Блум [8]) – цяло число от 1 до 5: **1**- усвояване на нови знания (нива „знание“ и „разбиране“ от Таксономията); **2**- преговор на стари знания (нива „разбиране“, „приложение“ и „анализ“); **3**-упражнение и усъвършенстване на знанията (нива „приложение“, „анализ“ и „синтез“); **4**-обобщение и систематизиране (нива „анализ“, „синтез“ и „оценка“); **5**-проверка на знанията (ниво „оценка“).

Следователно всеки електронен урок в образователния портал представлява вектор с посочените по-горе измерения:

**Lesson**(info(ID, Title, Domain, Author,... ), Subdomain(concept,m), Num\_class, Form\_of\_teaching, Less\_status, Didactic\_aims)

<sup>1</sup> SCORM-Sharable content object referent model

<sup>2</sup> Държавните образователни изисквания и Законът за общообразователния минимум

Например, за урока „Минало несвършено време на глагола“ от област „Български език“ за 5-ти клас, с автор Сарафов, в който се изучават понятията от матрицата *subdomain*, и е предназначен за ученици на самостоятелно обучение; свободен е за ползване за всички потребители в образователната мрежа и е урок за нови знания получаваме:

**Lesson1**(info(ID, „Минало несвършено време на глагола“,Български език, Сарафов,...), *subdomain\**, 5, 2, 4, 1), където *subdomain\** се представя със следната таблица:

ПОНЯТИЯ	СТЕПЕН НА ИЗУЧАВАНЕ
глагол	3
лице на глагола	2
число на глагола	3
време на глагола	2
момент на комуникация	3
момент на действие	3
основен ориентационен момент	2
допълнителен ориентационен момент	2

Когато ученикът изисква стартирането на урок по избрана тема, системата проверява наличието на подходящ урок от този регистър. Уроците, които отговарят на първоначалните потребителски изисквания, обикновено са повече от един и затова системата трябва да осигури подходящ механизъм за избор на най-подходящия. След диалог с потребителя, персоналният агент формализира неговите лични цели, предпочитания, очаквания и желаниа и предава този вектор на системата за избор, където се извършва сравняване с векторите на предоставените в регистъра уроци и се извличат тези, които имат най-високо ниво на сходство. Резултатът ще е множество от уроци, от които системата трябва да избере най-подходящия. Този избор може да се реализира чрез някакъв интелигентен алгоритъм (напр.CBR-подхода).

Предпочитанията и личните цели на един ученик също можем да формализираме чрез политика, която да формализира последователността на действията в конкретния учебен сценарий. След идентификация на ученика в обучителната среда, на база профилната и стереотипна информация, както и евентуален явен диалог, системата получава необходимите начални стойности на наблюдаваните променливи. След определяне на началното състояние управлението на политиката може да се предаде на специална система за управление на политиките (*policy-engine*), която е част от инфраструктурата на *run-time* на образователната среда в портала за електронно обучение.

Първоначално, въз основа на диалога с потребителя, съответната политика на предпочитанията се регистрира в *policy-engine* и след това се стартира механизъмът за избор на урок, който прави заявка към регистъра на уроците. След като се избере конкретен урок, той се подава на SCORM-Learning Management System (LMS) за изпълнение. Сценарият, по който ще се стартират дейностите в учебния процес са описани и формализирани от Sequencing & Navigation- модела на стандарта SCORM[15] и съответния шаблон, чрез който е създаден урока. *Policy-engine* може непрекъснато да променя политиките, според информацията, която постъпва от поведението на обучаемия. Учебният сценарий може да включва и задължителни за изпълнение действия (напр. тестове). Ако ученикът не се справи с тях, учебният процес попада в критично състояние и това води до повторен избор и стартиране на по-предпочитан урок. В този случай процесът на обучение временно се прекъсва и отново да се стартира с новия урок.

Предпочитанията, които определят избора на урок, се дефинират като специална политика на предпочитанията. Тя се изразява чрез правила от вида условие-действие. Условието представят множество от поведения, които предизвикват определени действия. Формалната семантика на модела се базира на IFL, като правилата имат следния вид:

**when *b* [increase | decrease] preference in lesson [*little* | *medium* | *high*],**

където ***b*** поведение, а **lesson** е конкретния урок.

Тук намаляването или увеличаването на предпочитанията се моделира като елементарно действие, което придава нова стойност на предпочитанието към определен урок. Степента на предпочитание може да се представи като цяло число. По-голямото число представлява по-висока степен на предпочитание. Да приемем, че първоначално няма предпочитания и всички стойности са 0. Дефинираме смисъла на малко, средно и високо ниво като 1, 2 и 3. За реализацията на този модел е подходящо да се използват персонални агенти, които да следят учениковото поведение и взаимодействат с *policy-engine*.

Ще разгледаме един пример за евентуално обучение с два урока върху един и същи учебен материал като първият е по-труден и разглежда понятията на по-високо ниво от втория. Ученикът първоначално не е решил какво е неговото предпочитание към тях. В системата е дефинирана политика, която определя, че уроците, гарантиращи повече от 70% успеваемост на финалния тест са за предпочитани пред онези, които

гарантират между 50% и 70%, а тези, които гарантират под 50% са непредпочитани. Можем да изразим политиката със следните правила:

```
score (Lesson1,Lesson2) :
when (1: test_result >= 70%) increase preference in Lesson high
when (2: 50%<=test_result <70%) decrease preference in Lesson medium
when (3:test_result<50) decrease preference in Lesson little
```

От тези правила policy-engine определя информацията, нужна за изпълнението. LMS чрез SCORM RTE (Run-Time Environment) и механизма на целевите променливи следи стойностите на успеваемостта на ученика при решаването на теста. След всеки опит policy-engine проверява предположението, дефинирано от правилата и определя дали то се изпълнява. Например, нека ученикът си е поставил за цел да усвои учебния материал на високо ниво (3) и резултатите от три последователни опита при решаването на финалните тестове за двата урока са:

Урок	Урок 1			Урок 2		
Опит	1	2	3	1	2	3
Резултат	55%	49%	60%	64%	66%	72%

Първоначално системата стартира по-трудния урок Урок1. След първия опит за решаване на финалния тест policy-engine активизира второто правило, тъй като резултатът е между 50% и 70%. Това определя предпочитанието към Урок1 на 2. Следващата стойност е <50% и това намалява предпочитанието от 2 до 1, тъй като се стартира правило три. Последното взаимодействие стартира отново правило две и увеличава предпочитанията от 1 до 2. Този резултат е незадоволителен за личните цели, които си е поставил потребителя и в резултат на това към policy-engine се подава искане за смяна на урока като неподходящ. Процесът на обучение временно спира и продължава с изпълнението на учебните действия на Урок2, където предпочитанията още при първия опит се покачват на 2, при втория се запазват, а при третия се повишават до 3, което е напълно задоволителен резултат за целите, които ученикът си е поставил.

Адаптивността на системата за електронно обучение спрямо знанията и предпочитанията на ученика можем да разглеждаме на три нива – елементарно, статично и динамично. Елементарното ниво се гарантира от профилната информация за конкретния ученик при идентификацията му в системата и допускането му до съответните ресурси в учебната среда за съответния клас и форма на обучение. Статичното адаптивно ниво се базира на модела за избор на най-подходящ урок от регистъра на уроците. Динамичното адаптивно ниво се реализира чрез политиките на предпочитания и policy-engine, която следи поведението на ученика и нивото на предпочитанието му към съответния урок като в самия ход на обучението може да замени текущия урок с друг – по-подходящ. Следователно, приложението на интервалната темпорална логика и политиките може да гарантира високо ниво на персонализация на учебния процес и достигане на по-високите нива на адаптивност на системата за електронно обучение.

### Използвана литература

- [1] Андреев, М. Процесът на обучение. Дидактика. София, 1996.
- [2] Асенова, П., А. Ескенази, MULTIVERSUM-европейски проект с българско участие – един нов подход към електронното обучение в училищата, “Новите технологии в образованието и професионалното обучение”, стр. 299-306, София, 2003.
- [3] Бижков, Г. Реформаторска педагогика. София, Просвета, 1994.
- [4] Глушкова Т., Концептуален модел за разработка на система за електронно обучение в средното училище, Международен научен семинар „Intelligent eLearning Services and Architectures: Problems & Perspectives”, 2007,
- [5] Glushkova T., M. Trendafilova, N. Uzunova, Application of SCORM standard for e-learning in the secondary school, In: Proc. Of the International Conference “Informatics in the Scientific Knowledge”, 18-22 June 2006, Varna, pp.205-216, 2006. ISSN 1313-4345, ISBN – 10:954-715-303-X, ISBN-13:978-954-715-303-5
- [6] Glushkova T., E-learning environment for support of secondary school education, Cybernetics and information technologies, volume 7, No 3, p.89-106, 2007, ISSN: 1311-9702
- [7] Glushkova T., User modeling of distributed e-learning systems for the secondary schools, In: Proc. Of the International conference DIDMATHTECH, 2007, p.117-123, Komarno, Slovakia, ISBN-978-80-89234-23-3.
- [8] Bloom B. S. (1956). Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain. New York: David McKay Co Inc.
- [9] Hoagland J. Specifying and implementing security policies using lasco, the language for security constraints on objects, Ph.D. dissertation, University of California, 2000.

- [10] Damianou N., N. Dulay, E. Lupu, and M. Sloman, “The ponder specification language”, in Workshop on Policies for Distributed System and Networks (Policy 2001), January 2001
- [11] B. Moskowski, Reasoning about Digital Circuits. PhD Thesis, Department of Computer Science, Stanford University, 1983. (Available as technical report STAN–CS–83–970.)
- [12] Sloman M. Policy driven management for distributed systems, Journal of Network and System Management, vol.2, pp.333-360, 1994, Available: <http://citeseer.ist.psu.edu/sloman94policy.html>, ISSN 1064-7570 (Print) 1573-7705 (online), Springer New York
- [13] Stoyanov, S., D. Mitev, I .Minov, T. Glushkova, eLearning Development Environment for Software Engineering Selbo 2, In: Proc. of the 19<sup>th</sup> International Conference on Database and Expert Systems Application, Turin, Italy, pp. 100-104, 2008, ISBN: 978-3-540-85653-5.
- [14] Best Practices Guide for Content Developers, LSAL, Carnegie Mellon University – [www.lsal.cmu.edu/lsal/expertise/projects/developersguide/](http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide/), посетен на 12.01.2010
- [15] SCORM2004, <http://www.adlnet.gov>, посетен на 12.01.2010.