

РЕАЛІЗАЦІЯ МОДЕЛІ АДАПТИВНОГО ТЕСТУВАННЯ В CPN TOOLS

У статті розглянуто модель адаптивного тестування, яку реалізовано та апробовано в середовищі CPN Tools і яка дозволяє зменшити число тестових завдань для студентів із правильними профілями знань і об'єктивно визначити їх рівень підготовки.

В статті рассмотрена модель адаптивного тестування, которая реализована и апробирована в среде CPN Tools и которая позволяет сократить количество тестовых заданий для студентов с правильными профилями знаний и объективно определить их уровень подготовки.

The article explores a model of adaptive testing which was created in a CPN Tools environment and facilitates a reduction in the quantity of test tasks for students with correct knowledge profiles, and determines their level of preparation objectively.

Постановка проблеми у загальному вигляді. Однією з областей інформатизації освіти є проведення контролю засвоєння навчального матеріалу з використанням різних тестуючих програм. Особливої уваги заслуговує метод адаптивного тестування, поява якого була зумовлена намаганням підвищити ефективність педагогічних замірів, яка насамперед пов'язувалась зі зменшенням числа завдань, часу, вартості тестування, а також підвищенням точності оцінювання [1].

Аналіз останніх досліджень і публікацій. Протягом багатьох років проблеми адаптивного тестування звертали на себе увагу зарубіжних і вітчизняних учених. Про це свідчать багаточисельні дослідження фундаментального характеру і публікації таких зарубіжних авторів як J.A. Arter, R.K. Hambleton, J.L. Horn, C.D. Jensema, G.G. Kingsbury, F.M. Lord, D.J. Weiss, R.W. Wood і багато інших. Питанням тестування та суміжній проблематиці присвячені праці В.С. Аванесова, Т.М. Балихіної, Н.Ф. Єфремової, В.П. Панасюка, І.Д. Рудинського, М.Б. Челишкової й ін [2].

Формулювання цілей статті. Розробити та реалізувати алгоритм адаптивного тестування на основі однопараметричної моделі Раша в середовищі CPN Tools.

Виклад основного матеріалу дослідження. Під адаптивним тестуванням розуміють широкий клас методик тестування, які передбачають зміну послідовності подання завдань у самому процесі тестування із врахуванням відповідей студента на вже надані завдання. Під сценарієм адаптивного тестування будемо розуміти індивідуальний набір тестових завдань з різними рівнями складності, кожний тест якого обирається для кожного студента в залежності від його відповіді на попереднє питання.

Модель сценарію адаптивного тесту представимо мережею Петрі $S = \langle P, T, F, M_0 \rangle$, де:

$P = (Tin, S, StudentLog, Tests, LOGITS, level1, level 2, level 3, level 4, StudentLevel, StudentNewLevel, Testing, R, true answer, false answer, MAP, Nt, Tres, of, Tout);$

$T = (s, StudLevel, Logits of stud level(model of Rash), logit, PL, Select level1, Select level2, Select level3, Select level4, P, SelectLevel, true answer, falseanswer, P0, P1, Analysis map of knowledges, testof, TestOf, U, R);$

F – функція інцидентності (табл. 1):

Таблиця 1

<p> $F(Tin, s) = (st, m, lr, sts);$ $F(S, StudLevel) = sn;$ $F(StudentLog, Logits of Stud level(model of Rash)) = s;$ $F(StudentLog, Select Level) = s;$ $F(Tests, logit) = test;$ $F(LOGITS, PL) = logitlv;$ $F(level 1, Select level1) = logitlv;$ $F(level 2, Select level2) = logitlv;$ $F(level 3, Select level3) = logitlv;$ $F(level 4, Select level4) = logitlv;$ $F(R, Select level(i)) = r (i=1..4);$ $F(StudentLevel, Select level(i)) = sn (i=1..4);$ $F(Student NewLevel, P) = (sn, logitlv, r);$ $F(Student NewLevel, TestOf) = (sn, logitlv, r);$ $F(Testing, true answer) = (logitlv, sn);$ $F(Testing, false answer) = (logitlv, sn);$ $F(Testing, testof) = (logitlv, sn);$ $F(true answer, P1) = (sn, logitlv, r);$ $F(false answer, P0) = (sn, logitlv, r);$ $F(R, false answer) = r;$ $F(R, true answer) = r;$ $F(MAP, Analysis map of knowledges) = kz;$ $F(Nt, nt) = rl;$ $F(TRes, U) = s;$ $F(of, Analysis map of knowledges) = kz;$ </p>	<p> $F(s, S) = (st, 0, 0, \#3 st, "0", "0", \#3 st, 0, "0", "0", m);$ $F(StudLevel, StudentLog) = s;$ $F(Logits of stud level (model of Rash), StudentLevel) = sn;$ $F(logit, LOGITS) = logitlv;$ $F(PL, level i(i=1..4)) = if (Option.valueOf(Real.fromString(\#3(logitlv)))) = i then 1`logitlv else empty;$ $F(Select level(i), StudentNewLevel) = kz;$ $F(P, StudentLog) = sn;$ $F(P, MAP) = (sn, logitlv, r);$ $F(P0, StudentLog) = sn;$ $F(P0, MAP) = (sn, logitlv, r);$ $F(P1, StudentLog) = sn;$ $F(P1, MAP) = (sn, logitlv, r);$ $F(true answer, true answer) = if L_3 (Option.valueOf(Real.fromString(\#7(sn)))) = klv then 1`(\#3 kzc, logitlv, r) else 1`(\#1 kzc, logitlv, r);$ $F(false answer, false answer) = if L(Option.valueOf(Real.fromString(\#7(sn)))) = 1 then 1`(\#3 kzc, logitlv, r) else 1`(\#2 kzc, logitlv, r);$ $F(Analysis map of knowledges, Nt) = rl;$ $F(U, of) = s;$ $F(nt, Tout) = rl;$ $F(testof, TRes) = if (\#3 sn = \#2 sn) then 1`((\#1(\#1 sn), \#2(\#1 sn) + bmax, \#3(\#1 sn)), \#2 sn, \#3 sn, \#4 sn, \#5 sn, \#6 sn, \#7 sn, bmax, \#9 sn, \#10 sn, \#11 sn) else 1`(\#1 sn, \#2 sn, \#3 sn, \#4 sn, \#5 sn, \#6 sn, \#7 sn, 0, \#9 sn, \#10 sn, \#11 sn)$ </p>
--	--

W – правила спрацювання переходів між темами:

$W(\text{Logits of stud level(model of Rash)}) = [\#3 s * (\#2 s - \#3 s) > 0 \text{ andalso } \#3 s < \#2 s];$

$W(\text{SelectLevel}) = [(\#3 s = 0 \text{ orelse } \#2 s = \#3 s) \text{ andalso } \#7 s = \#3 \text{ logitlv andalso } \#1 \text{ logitlv} = \#11 s];$

$W(\text{Select level } I (i=1..4)) = [P(\text{Option.valueOf(Real.fromString } (\#4 \text{ sn)), Option.valueOf(Real.fromString } (\#3 \text{ logitlv})) >= 0.5 \text{ andalso } \#1 \text{ logitlv} = \#11 \text{ sn});$

$W(\text{true answer}) = [r=1 \text{ andalso } \#2 \text{ sn} < \text{max}];$

$W(\text{false answer}) = [r=0 \text{ andalso } \#2 \text{ sn} < \text{max}];$

$W(\text{testof}) = [\#2 \text{ sn} = \text{max}];$

$W(\text{TestOf}) = [\#2 \text{ sn} = \text{nt} \text{ orelse } \#8 \text{ sn} >= \text{bmax}];$

$W(P) = [\#8 \text{ sn} <= \text{bmax} \text{ andalso } \#2 \text{ sn} < \text{nt}];$

M_0 – початкова розмітка.

Розділимо модель адаптивного тестування на логічні етапи.

1 етап. Параметри маркера студента на вході в тестування мають вид (st, m, lr, sts) , де st – (індивідуальний номер, загальний бал, рівень підготовки), m – номер модуля, lr – сценарій пройдених тем (повинен бути пустим), sts – статус вивчення. Під час тестування параметри маркера студента трансформуються у наступні (рис. 1):

$colset \text{ testNum} = INT;$

$colset N = INT;$

$colset XI = int;$

$colset FI = STRING;$

$colset SUMFI = STRING;$

$colset SUMBJ = STRING;$

$colset P = STRING;$

$colset BAL = INT;$

$var l : STRING;$

$var b : BAL;$

$colset V = STRING;$

$colset W = STRING;$

$colset STUD = product S * N * XI * FI * SUMFI * SUMBJ * P * BAL * V * W * Module;$

$var s, sn, sd : STUD;$

де S відповідає st , N – загальна кількість виконаних тестових завдань, XI – загальна кількість правильно виконаних тестових завдань, FI – логіт рівня підготовки студента, $SUMFI$ і $SUMBJ$ – сума логітів рівнів підготовки студента та сума логітів складності завдань, P – імовірність правильного виконання студентом тестового завдання, BAL – сума набраних балів за тест, V і W – дисперсії по вибірці логітів рівнів підготовки та складності завдань, $Module$ – модуль, за темами якого студент проходить тести.

2 етап. Тестові завдання знаходяться у вузлі $Tests$ типу $Tests$ і мають такі параметри: номер модуля, номер теми, рівень складності. Рівень складності тестових завдань задається від 1 до n . Для того, щоб перевести рівні складності завдань, використовуємо перехід $logit$ з таким кодом (рис. 2):

$input (test);$

$output (logitlv);$

```

action
let
val k=Real.fromInt klv;
val b= 40.0/ k;
val lv=Real.fromInt(#3 test);
fun level(lv) = Real.realFloor (30.0+(lv-1.0)*b-50.0)/4.55;
val logitlv=(#1 test,#2 test,Real.toString(level(lv)));
in
logitlv
end

```

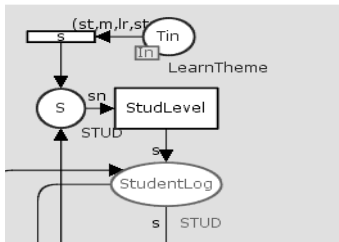


Рис. 1. Підготовка параметрів студента

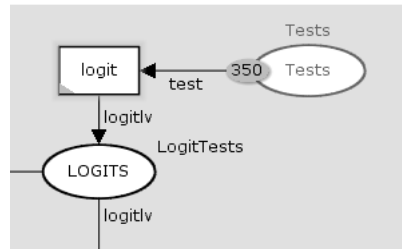


Рис. 2. Підготовка тестових завдань

- етап. Маркер студента переміщується у вузол StudentLog і в залежності від його відповідей на тестові завдання або проходить тести у блоці 1 (рис. 3) або в блоці 2 (рис. 4).
-

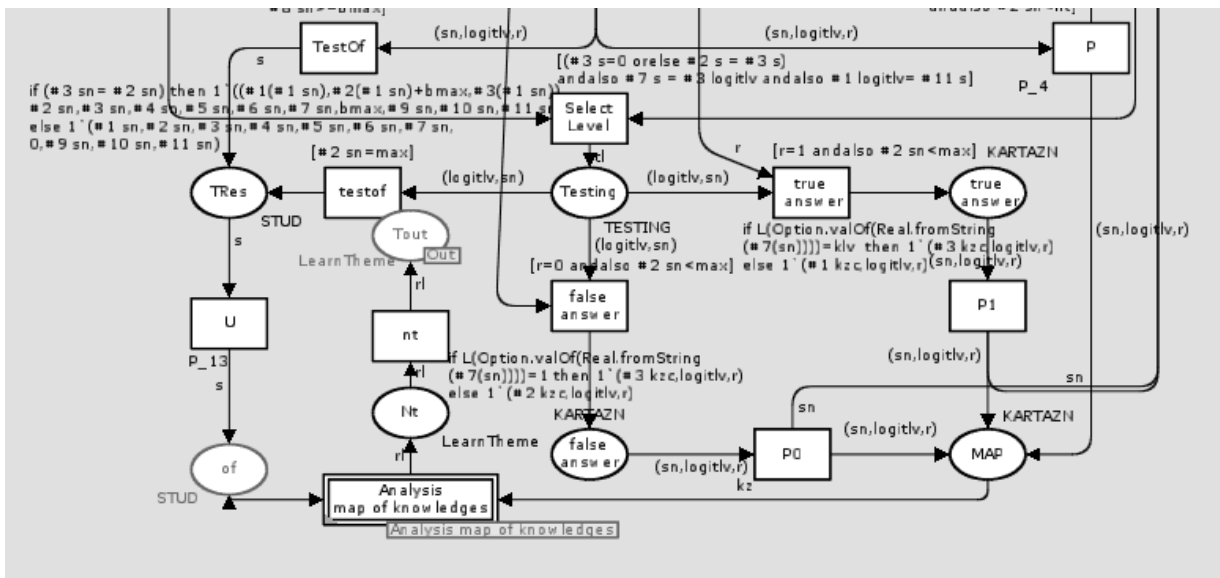


Рис. 3. Тестування у випадку всіх вірних або невірних відповідей студента

У блок 1 маркер студента потрапляє у випадку всіх вірних або всіх невірних відповідей. У вузлі *Select Level* студенту підбирається тестове завдання, відповідне його рівню підготовки та пройденому модулю, тобто рівень складності завдання дорівнює рівню підготовки студента ($\#7 s = \#3 logitlv \text{ andalso } \#1 logitlv = \#11 s$). У разі правильних відповідей рівень підготовки студента збільшується до максимального й студент залишається на цьому рівні, неправильних – зменшується до мінімального та залишається на цьому рівні. Коли кількість правильних (неправильних) відповідей підряд досягає

максимального заданого викладачем числа відповідей, необхідного для виставлення оцінки, тестування завершується.

4 етап. У випадку, коли загальна кількість тестових завдань більша за кількість правильних відповідей, маркер студента потрапляє через перехід *Logits of stud level (model of Rash)*, у якому розраховується його рівень підготовки за моделлю Раша (код наведений нижче), до вузла *StudentLevel*:

```

input (s);
output (sn);
action
let
val nzag = Real.fromInt (#2 s);
val Xi = Real.fromInt (#3 s);
val l=Option.valueOf(Real.fromString (#4(s)));
val pi = Xi / nzag;
val qi =1.0-pi;
val Fi = Math.ln (pi /qi);
val SumFi = Fi+ Option.valueOf(Real.fromString (#5(s)));
val Fisr = SumFi/nzag;
val Sumbj = Option.valueOf(Real.fromString (#6(s)));
val bjsr = Sumbj/ nzag;
val VV=Math.pow(Fi,2.0)-Math.pow(Fisr,2.0);
val V = (VV +Option.valueOf(Real.fromString (#9(s))))/(nzag-1.0);
val UU=(Math.pow(l,2.0)-Math.pow (bjsr,2.0));
val U = (UU+Option.valueOf(Real.fromString (#10(s))))/(nzag-1.0);
val X = Math.sqrt(((1.0+U)/2.89)/((1.0-U*V)/8.35));
val Y = Math.sqrt(((1.0+V)/2.89)/((1.0-U*V)/8.35));
val Fi0 = bjsr+X*Fi;
val f=Fi0;
val fi = Real.toString Fi;
val sn=(#1(s),#2(s), #3(s),fi,Real.toString(SumFi),
Real.toString(Sumbj),#7(s),#8(s),Real.toString VV,Real.toString UU,#11 s);
in
sn
end

```

Далі студенту підбирається тестове завдання в залежності від ймовірності правильної відповіді на нього, що задана в CPN Tools як функція *fun* $P(f,l)=(\text{Math.exp}(1.7*(f-l)))/(1.0+\text{Math.exp}(1.7*(f-l)))$ (де *f* – рівень підготовки студента, *l* – рівень складності завдання) і є правилом спрацювання переходів *Select level1*, *Select level2*, *Select level3*, *Select level4*: $[P(\text{Option.valueOf(Real.fromString}(\#4\ sn)),\text{Option.valueOf(Real.fromString}(\#3\ logitlv))\geq 0.5 \text{ and also } \#1\ logitlv = \#11\ sn]$.

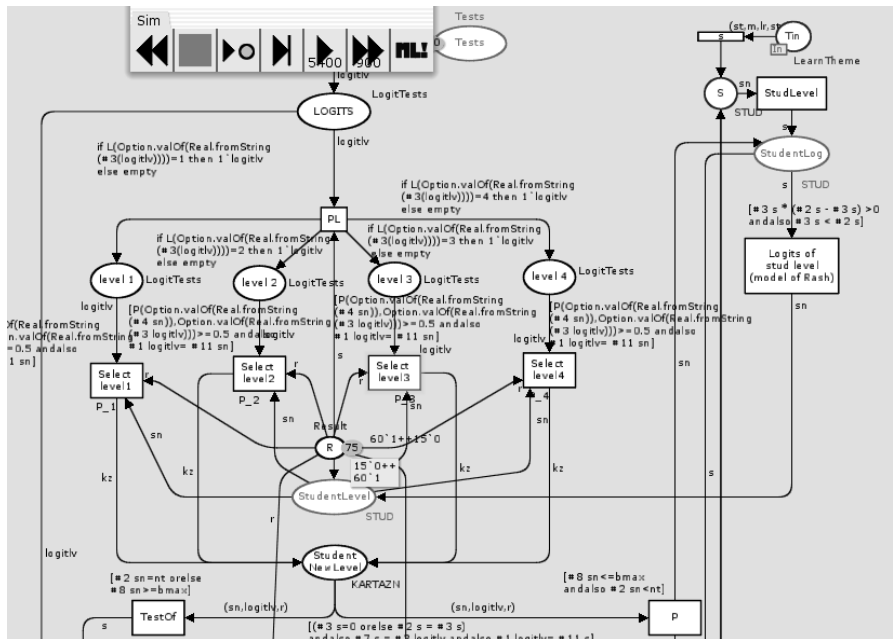


Рис. 4. Визначення рівня підготовки за моделлю Раша

Після відповіді студента на тестове завдання r з вузла $R = \{0, 1\}$, маркер потрапляє у вузол $StudentNewLevel$ типу $KARTAZN$ і має вид kz :

$colset KARTAZN = product STUD * LogitTests * Result;$
 $var kz : KARTAZN;$

де $LogitTests$ – тестове запитання, на яке отримана відповідь;
 $Result$ – 0 або 1 в залежності від вірності відповіді.

Далі процедура розрахунку нового рівня підготовки повторюється до тих пір, поки не набрано максимальну кількість балів або кількість тестових завдань стала дорівнювати максимальній довжині тесту, заданої викладачем.

Під час тестування через складний перехід *Analysis map of knowledges* з вузла MAP у вузол W потрапляє перелік тестових завдань з відповідями на них кожного студента та по кожній темі підраховується кількість правильних і неправильних відповідей у вузлі Q за допомогою переходів $T1$ і $T2$ (рис. 5).

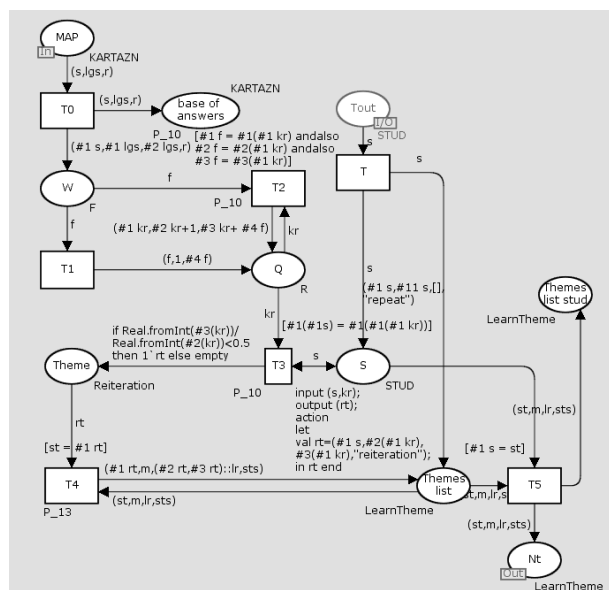


Рис. 5. Аналіз карти знань студента

Після завершення тестування спрацьовує перехід $T3$ і формується список тем, які необхідно повторити при умові, що на тестові завдання з цих тем студент дав 50% неправильних відповідей (*if Real.fromInt(#3(kr))/Real.fromInt(#2(kr)) < 0.5 then 1`rt else empty*).

У вузлі Themes list отримуємо сценарій тем для повторення. У вузлах *base of answers* і *Themes list stud* зберігаються відповідно база відповідей на тестові завдання та сценарії для повторення по кожному студенту.

Проаналізуємо результати експерименту на прикладі трьох студентів з різним рівнем підготовки: (11,0,"3"), (12,0,"4"), (16,0,"2").

Після проходження тестування з першого і другого модулів були отримані такі сценарії для повторення навчального матеріалу:

1 студент: 1`((11,34,"2"),1,[(1,3)],"repeat");1`((11,74,"2"),2,[],"repeat");

2 студент: ((12,41,"2"),1,[],"repeat"); ((12,86,"3"),2,[],"repeat");

3 студент: 1`((16,20,"1"),1,[(1,1),(1,5)],"repeat"); 1`((16,45,"1"),2,[(2,3)],"repeat").

Профілі студентів мають вид (завдання розташовані у порядку підвищення складності) (табл. 2):

Таблиця 2

	1 рівень складності	2 рівень складності	3 рівень складності	4 рівень складності
Студент № 11	1011	111101	001111	
Студент № 12	11111110	110110111	011	
Студент № 16	11000101000111000	0111	1	

Аналізуючи дані таблиці 2 і рис. 6, можна зробити висновок про те, що чим правильніший профіль студента, тим швидше визначиться його рівень підготовки і тим менше завдань йому потрібно виконати.



Рис. 6. Підбір складності тестових завдань під рівень підготовки студента

Висновки. Запропонована модель адаптивного тестування дозволяє зменшити число тестових завдань для студентів із правильними профілями знань і об'єктивно визначити їх рівень підготовки. Розроблена у статті модель реалізована програмно у дистанційній адаптивній системі DAOS (Distance Adaptive Open System).

Література

1. Звонников В.И., Чельшкова М.Б. Современные средства оценивания результатов обучения. – М.: Издательский центр «Академия», 2007. – 224 с.
- Ефремова Н.Ф. Тестовый контроль в образовании. Учебное пособие. – М.: Университетская книга; Логос, 2007. – 386 с.