

## ПІДСИСТЕМА АНАЛІЗУ ТА ОЦІНКИ ВІДПОВІДЕЙ КОМП'ЮТЕРНОЇ НАВЧАЛЬНОЇ СИСТЕМИ “PILOT”

© Чабан К., 2001

**Описано методологію автоматичної оцінки знань студентів за 5- і 100 бальними шкалами. Синтезовано, мінімізовано та досліджено математичні моделі алгоритмів розв'язання задач із письмовими відповідями і текстовим опитуванням.**

### Вступ

Одним з основних напрямків підвищення ефективності навчального процесу є його комп'ютеризація. Комп'ютер застосовується як для підтримки традиційного навчання, так і для навчання за допомогою комп'ютерних навчальних систем (КНС). Використання КНС в навчальному процесі дає змогу організувати безперервний і всебічний контроль за засвоєнням навчального матеріалу. Забезпечується масовість контролю при збереженні його індивідуальності, можливість отримання достовірної оцінки знань кожного учня протягом періоду навчання з мінімальними затратами часу на проведення контролю. Із використанням КНС у вищій школі оптимізується навчальна діяльність, інтенсифікується процес засвоєння студентом матеріалу, але актуальною залишається проблема створення інформаційних засобів ефективного контролю та об'єктивного оцінювання знань студентів. Як свідчить аналіз літератури, сьогодні немає цілісної методики комп'ютерного оцінювання знань, реальні знання студентів не дістають об'єктивного відображення. В цій роботі розглядається підсистема аналізу та оцінки відповідей КНС “Pilot” [1].

### 1. Аналіз відповіді та методика виставлення оцінки

КНС “Pilot” складається з двох основних частин: оболонки – керуючої програми та бібліотеки завдань (задачі, запитання, тексти тощо). Оболонка навчальної системи є її ядром. Вона являє собою завершений користувацький і графічний інтерфейс, забезпечує діалог з користувачем, відповідно до вибраного режиму роботи формує завдання, приймає та аналізує відповідь, виставляє оцінку, формує рейтинг. Система “Pilot” виставляє оцінки за 100- і 5-бальними шкалами. Вихідною є 100-бальна шкала. Методика обчислення оцінок за цією шкалою залежить від типу задачі.

Для задач, що вимагають цифрової відповіді, оцінка розраховується за формулою:

$$\text{Оцінка}_{100} = 100 * (1 - |A - B| / B),$$

де А – відповідь студента, В – правильна відповідь.

Отже, оцінка залежить від похибки обчислень. Перехід із 100-бальної в 5-бальну відбувається автоматично, згідно з рекомендованою шкалою переведення [5].

Складніше, якщо необхідно ввести словесну відповідь. Система “Pilot” дає можливість в кожному конкретному випадку, для будь-якої задачі, задати алгоритм її оцінки. Стандартний ж алгоритм такий: рядок відповіді студента посимвольно порівнюється з рядком правильної відповіді і підраховується кількість символів, що збігаються, а далі оцінка розраховується за формулою:

$$\text{Оцінка}_{100} = 100 * N / N_0,$$

де  $N$  – кількість символів, що збігаються,  $N_0$  – кількість символів у правильній відповіді.

Відповідаючи на запитання і тести, студент з пропонуванних відповідей вибирає ту, що на його погляд, найбільш правильна. Звичайно, якщо номер вибраної студентом відповіді і номер правильної відповіді збігаються – оцінка максимальна, інакше – мінімальна. У системі “Pilot” використано найдосконаліший стандартний алгоритм. Складаючи питання, автор володіє всіма можливими відповідями на них і розміщує їх в такій послідовності, щоб на першому місці була найповніша відповідь, на другому – теж правильна, але менш повна і так далі. На екран список можливих відповідей виводиться кожного разу в довільній, випадковій послідовності, що позбавляє змісту просте запам’ятовування або записування номера правильної відповіді. Студентові треба запам’ятовувати зміст відповіді (що, власне кажучи, і є метою навчання). Оцінка ж розраховується так :

$$\text{Оцінка}_{100} = 100 * (N_0 - N - 1) / N_0,$$

де  $N$  – номер відповіді студента у вихідному, ранжованому автором списку відповідей,  $N_0$  – кількість варіантів можливих відповідей.

Отже, вираховуються оцінки за окремі задачі, тести і питання.

Середні оцінки визначаються з врахуванням складності окремих задач, кожна з яких має свій коефіцієнт складності  $V$ . Коефіцієнт завдання середньої складності дорівнює одиниці. Простіші завдання можуть мати коефіцієнт, наприклад, 0.7, але, зрозуміло, більше від нуля, більш складні – 1.8 або навіть більше. Середня оцінка обчислюється за формулою:

$$\text{Середня оцінка}_{100} = \sum V \cdot \text{оцінка}_{100} / \sum V,$$

де підсумовування виконується за кількістю запропонованих студентові задач.

## 2. Синтез секвенцій

Секвенція, якою описується знаходження значення відхилення відповіді студента від правильної відповіді, має такий вигляд :

$$S_1 = B; P(k, ko); P(kb, c_7); P(i, c_7); P(h, V(c_{12}, c_{13})); P(j, M(a, V(c_{12}, h))); U,$$

де  $B$  – опис проміжних змінних, задання масиву, присвоєння змінній  $a$  максимального значення бала, підрахунок кількості символів у правильній відповіді і відповіді студента;  $P(k, ko)$  – присвоєння проміжній  $k$  кількості символів у правильній відповіді;  $P(kb, c_7)$  – присвоєння змінній для підрахунку однакових символів у правильній відповіді і відповіді студента початкового значення – нуля.  $P(h, V(c_{12}, c_{13}))$  – присвоєння значення  $h = 1.0 - 1 * 10^{-6}$ ;  $P(j, M(a, V(c_{12}, h)))$  – обчислення значення оцінки в балах, як  $j = a * (1 - h)$ , де

$$B = P(a, c_1), P(ko, s), P(kr, t),$$

$P(a, c_1), P(ko, s), P(kr, t), P(k, ko), P(kb, c_7), P(i, c_7), P(h, V(c_{12}, c_{13})), V(c_{12}, c_{13})$  – двомісні оператори,  $U$  – формула оцінки розв’язку задач типу  $Z[2]$ .

Якщо в секвенції  $S_1$  оператор  $P(h, V(c_{12}, c_{13}))$  замінити оператором  $P(h, A(D(V(ko, kb), ko)))$ , яким описується знаходження абсолютного значення похибки в символній відповіді, то отримаємо секвенцію  $S_2$ .

$$S_2 = B; P(k, ko); P(kb, c_7); P(i, c_7); P(h, A(D(V(ko, kb), ko))); P(j, M(a, V(c_{12}, h))); U$$

Символьна відповідь утворена із фіксованої кількості знаків. Кожен символ введеної студентом відповіді порівнюється з кожним символом правильної відповіді. Підраховується як загальна кількість символів відповіді, так і кількість символів, що збігаються із символами правильної відповіді. Це описується секвенціями  $S_3, S_4$ .

$$S_3 = B; P(k, ko); P(kb, c_7); P(i, c_7); \mathcal{C}_i; P(kb, S(kb, c_{12}));$$

$$S_4 = B; P(k, ko); P(kb, c_7); P(i, c_7); \mathcal{C}_i,$$

де  $S(kb, c_{12})$  – двомісний оператор;  $i$  – змінна кількості знаків символьної відповіді.

Якщо кількість символів у правильній відповіді більша за кількість символів у відповіді студента, то оператор  $P(kb, c_7)$  в секвенціях  $S_1, S_2$  замінюється оператором  $P(k, kr)$ . В результаті отримуємо секвенції  $S_5, S_6$ .

$$S_5 = B; P(k, ko); P(k, kr); P(h, V(c_{12}, c_{13})); P(j, M(a, V(c_{12}, h))); U ;$$

$$S_6 = B; P(k, ko); P(k, kr); P(h, A(D(V(ko, kb), ko))); P(j, M(a, V(c_{12}, h))); U .$$

Аналогічно  $S_5, S_6$  із секвенцій  $S_3, S_4$  утворюються секвенції  $S_7, S_8$ , які матимуть такий вигляд:

$$S_7 = B; P(k, ko); P(k, kr); \mathcal{C}_i; P(kb, S(kb, c_{12}));$$

$$S_8 = B; P(k, ko); P(k, kr); \mathcal{C}_i .$$

При відсутності відповіді студента матимемо секвенцію  $S_9$

$$S_9 = B; P_1(c_{14}),$$

де  $P_1(c_{14})$  – повідомлення системи, що екзамен зданий.

### 3. Синтез елімінувань

Секвенції  $S_1, S_2$  елімінуємо за умовою  $u_{17}$  нерівності  $c_7$  кількості символів у правильній відповіді, отримаємо таку формулу:

$$\overline{S_1, S_2, u_{17} - ?}$$

За умовою збігу символів відповіді студента і символів правильної відповіді ( $u_{18}$ ) елімінуємо секвенції  $S_3, S_4$ .

$$\overline{S_3, S_4, u_{18} - ?}$$

Секвенції  $S_1, S_2, S_3, S_4$  елімінуємо за умовою  $u_{16}$  – збігу поточної кількості символів у відповіді студента із фіксованою кількістю символів правильної відповіді, матимемо формулу (1).

$$\overline{\overline{S_1, S_2, u_{17} - ?} \mid \overline{S_3, S_4, u_{18} - ?}, u_{16} - ?} \quad (1)$$

За умовою  $u_{17}$  елімінуємо секвенції  $S_5, S_6$ , одержимо формулу:

$$\overline{S_5, S_6, u_{17} - ?}$$

За умовою  $u_{18}$  елімінуємо секвенції  $S_7, S_8$ , отримаємо такий вираз:

$$\overline{S_7, S_8, u_{18} - ?}$$

Останні два елімінування елімінуємо за умовою  $u_{16}$ , матимемо формулу (2):

$$\overline{\overline{\overline{S_5, S_6, u_{17} - ?} \mid \overline{S_7, S_8, u_{18} - ?}}, u_{16} - ?} \quad (2)$$

Вирази (1), (2) елімінуємо за умовою  $u_{15}$  – порівняння кількості символів правильної відповіді і відповіді студента, отримаємо формулу (3).

$$\overline{\overline{\overline{u_{15} - ?} \mid \overline{\overline{\overline{S_1, S_2, u_{17} - ?} \mid \overline{S_3, S_4, u_{18} - ?}}, u_{16} - ?} \mid \overline{S_5, S_6, u_{17} - ?} \mid \overline{S_7, S_8, u_{18} - ?}}, u_{16} - ?} \quad (3)$$

#### 4. Оптимізація структури алгоритму

Згідно з властивостями операцій секвенційної алгебри алгоритмів [3] в прямому елімінуванні виносимо за знак операцій елімінування такі оператори  $B, P(k, ko); P(kb, c_7), P(i, c_7), P(j, M(a, V(c_{12}, h)))$ ,  $U$ , отримаємо формулу (4).

Аналогічно оптимізуються всі решта елімінувань. В результаті отримаємо оптимізовану за кількістю операторів, операцій секвентувань та циклічних секвентувань структуру алгоритму, яка наведена формулою (5).

$$\begin{array}{l}
 B; \\
 P(k, ko); \\
 P(kb, c_7); \\
 P(i, c_7); \\
 \overline{P(h, V(c_{12}, c_{13})), P(h, A(D(V(ko, kb), ko))), u_{17} - ?}; \\
 P(j, M(a, V(c_{12}, h))); \\
 U
 \end{array} \quad (4)$$

$$\begin{array}{l}
 B; \\
 \overline{P(k, ko), P_1(c_{14}), u_{14} - ?}; \\
 ; \\
 \overline{P(kb, c_7), P(k, kr), u_{15} - ?}; \\
 ; \\
 \mathcal{C}_i \\
 \overline{P(h, V(c_{12}, c_{13})), P(h, A(D(V(ko, kb), ko))), u_{17} - ?, P(kb, S(kb, c_{12})), c_i, u_{18} - ?, u_{16} - ?}; \\
 ; \\
 P(j, M(a, V(c_{12}, h))); \\
 ; \\
 U
 \end{array} \quad (5)$$

## 5. Модель структури алгоритму

**Теорема 1.** Структура алгоритму (5) має модель.

**Доведення.** Будують модель структури алгоритму, замінюючи змінні оператори (терми) предметними операторами (термами) із заданням секвентних областей значень термальних операторів, значень їхніх змінних та замінюючи змінні умови на предметні умови [4].

Змінні оператори  $P(a, c_1)$ ,  $P(ko, s)$ ,  $P(kr, t)$ , якими утворено В-структури алгоритму, замінюємо предметним оператором присвоєння ( $:=$ ). Отримаємо такий вираз:

$$\overbrace{a := c_1, ko := s, kr := t}$$

Нехай  $a, c_1 = 100$ ,  $ko, s, t, kr \in S_1 = \overbrace{0, 1, \dots, 80}$

$P(k, ko)$  є оператором присвоєння, для якого  $k \in S_2 = \overbrace{1, 2, \dots, 80}$

Оператор  $P_1(c_{14})$  замінюємо предметним оператором друкування повідомлення ( $c_{14} =$  “Екзамен зданий”). Змінну умову  $u_{14}$  замінюємо предметною умовою перевірки значення  $ko(ko \neq 0 - ?)$ .

Для оператора присвоєння ( $kb := c_7$ ) задаємо  $kb, c_7 \neq 0$ .

Оператор  $P(k, kr)$  є оператором присвоєння, тобто  $k := kr \in S_1$ .

Умову  $u_{15}$  замінюємо такою предметною умовою  $kr < ko$ .

Змінній циклу  $i$  задаємо таку секвентну область значень:

$$i \in S_3 = \overbrace{0; 1; \dots; 79}$$

В операторі  $P(h, V(c_{12}, c_{13}))$ , змінний оператор  $V(c_{12}, c_{13})$  замінюємо відніманням, в результаті отримаємо  $h := c_{12} - c_{17}$ . Для  $c_{12}$  задаємо константу 1, а для  $c_{13}$  –  $1 \cdot 10^{-6}$ . Аналогічно в операторі  $P(h, A(D(V(ko, kb), ko)))$  термальний оператор  $V(ko, kb) \in ko - kb$  а  $D(ko - kb, ko)$  замінимо таким предметним оператором  $(ko - kb) / ko$ .

$$ko \in S_2, kb \in S_2, kb \leq ko, (ko - kb) \in S_3, (ko - kb) / ko \in S_4 = \overbrace{0, \frac{1}{80}, \frac{2}{80}, \dots, \frac{79}{80}, 1}$$

Змінний оператор  $A$  замінюємо термальним оператором знаходження абсолютної величини, його значеннями є секвентна область  $S_4$ . Тоді оператор  $P(h, A(D(V(ko, kb), ko)))$

є таким предметним оператором  $h := \left| \frac{ko - kb}{ko} \right|$ ,  $h \in S_4$ .

Умову  $u_{17}$  замінюємо перевіркою кількості символів у правильній відповіді КНС на нерівність 0 ( $ko \neq 0$ ).

В  $P(kb, S(kb, c_{12}))$  змінний оператор  $S(kb, c_{12})$ , замінюємо додаванням  $(kb, c_{12})$ . Нехай  $c_{12} = 1$ ,  $kb \in S_1$ . Тоді  $kb := k + 1$ .

Умову  $u_{18}$  замінюємо перевіркою збігу кількості символів у відповіді студента із кількістю символів правильної відповіді ( $i < k$ ).

Для змінного оператора  $P(j, M(a, V(c_{12}, h)))$   $V(c_{12}, h) \in (1 - h)$ , а  $M(a, (1 - h))$  замінюємо предметним оператором множення  $(a * (1 - h))$ , де  $a = 100$ .

$$(a * (1 - h)) \in S_5 = \overbrace{0, 01, \dots, 49.9, 50, \dots, 99.9, 100}, j := a * (1 - h), j \in S_5$$

Модель структури алгоритму описаної формулою (5) побудована і наведена формулою (6). Теорема доведена.

$$\begin{array}{l}
 \overbrace{a := c_1, ko := S, kr = t} \\
 \left. \begin{array}{l}
 k := ko, P_1(c_{14}), (ko \neq 0) - ? \\
 ; \\
 \overbrace{kb := c_7, k := kr, (kr < ko) - ?} \\
 ; \\
 \mathcal{Q}_i \\
 \overbrace{h := c_{12} - c_{13}, h := |(ko - kb) / ko|, (ko \neq 0) - ?, kb := kb + c_{12}, c_i, (v_p = v_{st}) - ?, (i < k) - ?} \\
 ; \\
 j := a * (1 - h)
 \end{array} \right\} (6)
 \end{array}$$

## 6. Вірогідність моделі алгоритму

Математична модель структури алгоритму є вірогідною, якщо змінні та предметні оператори набирають значення тільки із заданих секвентних областей значень [4].

**Теорема 2.** Математична модель структури алгоритму (6) вірогідна.

**Лема 1.** Модель алгоритму вірогідна при початкових значеннях змінних предметних операторів.

**Доведення.** Константі  $c_1$  оператором  $a := c_1$  присвоюється значення  $a$ . Отже оператор присвоєння набуває значення із заданої області, тому він є вірогідним. Для оператора  $ko := s$  маємо початкове значення  $S = 0, ko = 0$ . Оператор вірогідний. Аналогічно для  $kr := t$  отримуємо  $t = kr = 0$ . Умова  $(ko \neq 0)$  не виконується, тому має місце оператор  $P_1(c_{14})$  – друк повідомлення. Якщо ж умова  $(ko \neq 0)$  виконується, тобто у цьому разі  $ko \neq 0$  належить секвентній області значень  $S_4$ . Далі перевіряється виконання умови  $(kr < ko)$ . У разі виконана умови  $kb := c_7 \neq 0$ , що є допустимим значенням, тому оператор вірогідний. Початковим значенням змінної циклу  $i \in 0$ . У разі виконання умови  $(i < k)$  і виконання  $(ko \neq 0)$  виконується оператор  $h := c_{12} - c_{13} = 1 - 1 * 10^{-6}$ , його значення допустиме. Після нього виконується  $j := a * (1 - h) = 100 * (1 - (1 - 1 * 10^{-6}))$ , що є допустимим значенням.

Коли ж умова  $(ko \neq 0)$  не виконується, то має місце оператор  $h := |(ko - kb) / ko|$ . Значення  $ko$  належить секвентній області  $S_2$ ,  $kb = 0$ , тому  $h = |(ko - 0) / ko| = 1$ , що є допустимим значенням, оскільки належить секвентній області значень  $S_4$ . Якщо не виконана умова  $(i < k)$ , і виконується умова  $(v_p = v_{st})$ , змінна циклу  $i$  набуває значення  $2 \in S_3$ , та знову аналізується виконання умови  $(i < k)$ .

У разі невиконання умови  $(kr < ko)$  має місце оператор  $k := kr$ , тут  $kr = t \in S_1$ , та  $k = kr \in S_1$ . Далі іде цикл за змінною  $i$ , який ми розглянули вище, та показали вірогідність всіх операторів, охоплених циклом.

Отже, встановлено вірогідність моделі для початкових значень змінних і операторів. Лема 1 доведена.

**Лема 2.** Математична модель (6) вірогідна для будь-яких  $n$ , які не перевищують кількості значень секвентних областей.

Доводиться аналогічно лемі 1. На основі трансфінитної індукції робимо висновок про те, що математична модель алгоритму є вірогідна. Теорема доведена.

## 7. Висновки

1. Розроблена методологія оцінки знань студентів забезпечила на основі секвенційної теорії алгоритмів виконання синтезу, оптимізації, побудови моделі і моделювання алгоритмів автоматичної оцінки знань студентів із письмовими відповідями і текстовим опитуванням.

2. Побудова математичної моделі структури алгоритму і дослідження її вірогідності дало змогу ще до практичної реалізації та апробації методології автоматичної оцінки знань студентів виявити допущені помилки.

3. Результатом оптимізації є зменшення кількості операторів в 1,3 раза.

1. Оганесян А.Г. Опыт компьютерного контроля знаний // Дистанционное образование. 1999. № 6. С.30–35. 2. Чабан К. Синтез інструментальних засобів розв'язки задач із цифровою відповіддю // Зб. наук. пр. "Комп'ютерні технології друкарства". 2000. № 4. С.262–268.

3. Овсяк В. Алгебра алгоритмів – секвенцій // Зб. наук. пр. “Комп’ютерні технології друкарства”. 1999. № 3. С.3–13. 4. Овсяк В. Метод подання, оптимізації, побудови математичних моделей і моделювання алгоритмів засобами алгебри алгоритмів // Вісн. ДУ “Львівська політехніка”. 2000. № 387. С.465-469. 5. Положення “Про систему підсумкового контролю, оцінювання знань та визначення рейтингу студентів”. ДУ “Львівська політехніка”, 1997.

УДК 681

A. Kernytskyy, M. Lobur, J. Wróbel\*

Lviv Polytechnic National University

\*Warsaw University of Technology

## APPROACHES TO ENGINEERING DATABASES

© Kernytskyy A., Lobur M., Wróbel J., 2001

**In the article problems arriving when developing and introducing engineering databases are outlined and the necessity of the research in the field of engineering databases is emphasized.**

### 1. Introduction

When starting speculation over possible approaches to engineering databases development and implementation next two possible ways come instantly into the mind:

- The first one is connected with development of engineering database as a part of the creation of CAD/CAM system.
- The second one concerns engineering databases as a core element of Integrated Engineering Document Management System (IEDMS). [1]

After analyzing the ways small enterprises solve the problem of automation of their design and work we came to the next conclusion: from the tree methods of creation of the enterprise information systems they choose the first one:

- Application of standard software.
- Acquirement of separate components and their following integration.
- Independent development.

Independent development of CAD/CAM and document management systems by small enterprises is ineffective because of scantiness of staff. On the other hand, a high cost of corporative information systems positioned on market does not allow enterprise to use them. Specific characteristics of small enterprises do not find their embodiment in such systems. That is why the development of engineering document management system reflecting specific activity character of small enterprise is highly necessary.

Besides, the Integrated Engineering Document Management System is an intermediate point on the way of creation of the Engineering-Manufacturing Information Systems (EMIS) (to which Polish enterprises are much closer because of the high competitiveness with western manufacturers) which is one of the main tools for the small enterprises to increase the manufacturing efficiency in the near future [2].

This way doesn't mean the rejection of the engineering database technology, because while designing and implementing the IEDMS:

- the technology and techniques of the engineering database development still reserve;
- modern trends of the engineering database development remain at the tips of developer's fingers.