



УДК 004.045

**TEST GENERATION FOR ADAPTIVE E-LEARNING SYSTEMS USING
ADVANCED METHOD BASED ON THE CONCEPT-THESIS MODEL**
**ГЕНЕРАЦІЯ ТЕСТІВ ДЛЯ АДАПТИВНИХ СИСТЕМ ЕЛЕКТРОННОГО НАВЧАННЯ
З ВИКОРИСТАННЯМ ВДОСКОНАЛЕНОГО МЕТОДУ НА ОСНОВІ ПОНЯТІЙНО-
ТЕЗИСНОЇ МОДЕЛІ**

Zaichko O.P. / Заїчко О.П.

магістрант

Kruhlyi D.V. / Круглий Д.В.

магістрант

Tytenko S.V. / Титенко С.В.

с.т.с., ас.проф. / к.т.н., доц.

ORCID: 0000-0002-7548-9053

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute",
Kyiv, Peremohy Avenue 37, 03056

Національний технічний університет України "Київський політехнічний інститут імені
Ігоря Сікорського", Київ, проспект Перемоги, 37, 03056

Анотація. У роботі розглядається метод автоматизованої побудови тестових завдань для систем електронного навчання, який базується на понятійно-тезисній моделі (ПТМ). Наводиться перелік аспектів способу, які можливо вдосконалити для збільшення його ефективності. Пропонується введення нових абстракцій в метод для збільшення гнучкості в створенні контрольних областей навчального матеріалу. Пропонується вдосконалити пошук альтернативних варіантів відповідей шляхом введення додаткової області навчального матеріалу. В результаті роботи була розроблена програмна система, яка використовує модернізований метод автоматизованої генерації завдань.

Ключові слова: електронне навчання, інформаційно-навчальні системи, онтологія, тестові завдання, понятійно-тезисна модель, автоматична генерація

Вступ

У сучасних освітніх технологіях особливий акцент зроблено на індивідуалізації процесу навчання, збільшення самостійної роботи студента та його розвиток контролюється дистанційними формами навчання. Таким чином величезну цінність має процес створення та доставки індивідуальних та тестових завдань. У зв'язку з цим у поточній статті йдеться про генерацію питань для перевірки набутих рівнів знань студентів. Значний обсяг досліджень у сфері перевірки знань зосереджений на якості та достовірності тестів. [1] Однак сучасна швидкість оновлення професійних галузей визначає важливість автоматизованого складання тестових завдань, які значно прискорюють підготовку навчальних курсів та середовищ у сучасних галузях навчання.

Запропоновано вдосконалений метод автоматизованої побудови засобів тестування на основі моделі формалізації дидактичного тексту. Основою даного методу є формалізація навчальних матеріалів на базі понятійно-тезисної моделі із подальшим використанням алгоритмів генерації та перевірки тестових завдань. Дана модель розроблена для формалізації дидактичного тексту і мультимедіа, що є вмістом інформаційно-навчальної програмної системи. Понятійно-тезисна модель (ПТМ) служить як засіб моделювання смислу контенту і відповідає за предметну формалізацію навчальних матеріалів. ПТМ



надає засоби опису понятійної складової контенту і забезпечує основу для програмного інструментарію редагування і використання бази даних та знань [2].

Основою підготовки тестів в ПТМ є семантичний розбір навчального тексту. Дана технологія дещо схожа із «семантичним конспектом», що запропонував Г. А. Атанов [3], є альтернативою до процесу формування бази знань (БЗ) у таких підходах, як семантичні мережі та концептуальні графи [4] тощо. Основною перевагою даного процесу є природність в рамках технології ПТМ. Крім того в даній моделі робиться акцент на педагогічному підході, тоді як традиційні технології штучного інтелекту акцентують увагу на питаннях, що є характерними для цієї сфери.

Перша ітерація методу описана в минулих дослідженнях [5] має простір для вдосконалення. Подальша робота над методом може покращити ефективність і гнучкість генерації завдань. У роботі пропонується поділити весь навчальний контент на різні гілки контенту. Комбінація гілок дозволяє нам адаптивно створити іншу абстракцію – курс, що відповідатиме специфічним освітнім потребам тієї чи іншої аудиторії користувачів. Також модернізується пошук альтернативних відповідей у завданнях множинного вибору. Це досягається шляхом задіювання додаткової області контенту.

Основний текст

Гілка контенту та адаптивний курс. Першим вдосконаленням алгоритму є збільшення гнучкості тестування. Це досягається введенням можливості генерувати тест як по гілці контенту, так і по курсу, який містить набір гілок:

$$TestBaseEntity \in \{BranchBased, CourseBased\} \quad (1)$$

Множину гілок і курсів позначимо так:

$$Branch = \{branch_i\} \quad (2)$$

$$Course = \{course_i\} \quad (3)$$

Кожен курс може містити у собі будь-яку кількість гілок, це задається відношенням:

$$CourseBranch: Course \rightarrow 2^{branch} \quad (4)$$

Гілка є вузлом в дереві навчального контенту V :

$$branch \in V \quad (5)$$

Множина усіх елементів-нащадків даного елемента $e \in V$ являє собою послідовне об'єднання множин дочірніх елементів вглиб по ієрархічній структурі. Для визначення елементів-нащадків застосовуватимемо оператор [6]

$$Desc(e), e \in V \quad (6)$$

Позначимо гілку або курс як основу тесту $branch_{test}$ і $course_{test}$ відповідно.

Вибір гілки або курсу задає ПТ-елементи, які ляжуть в основу тесту.



Множина елементів контенту з множини всього контенту V формує контрольну область тесту з урахуванням вхідної інформації про тип області контенту $TBE = TestBaseEntity$ гілку $branch_{test}$ або курс $course_{test}$:

$$SuffControl = \begin{cases} \{branch_{test}\} \cup Desc(branch_{test}), & TBE = BranchBased \\ \bigcup_{v \in CourseBranch(course_{test})} Desc(v), & TBE = CourseBased \end{cases} \quad (7)$$

Додаткова область навчального матеріалу. Нова множина елементів контенту $SuffAdditionalA \subseteq V$, яка формує додаткову область контенту для інших варіантів відповіді задана нижче:

$$SuffAdditionalA \cap SuffControlA = \emptyset \quad (8)$$

Вираз [5], який описує ПТ-елементи для пошуку контрольної пари в контрольній області контенту для деякого шаблону tt_i :

$$SCControlA = \{c: VC(c) \cap SuffControlA \neq \emptyset\} \quad (9)$$

$$STControlA = \{t: VT(t) \in SuffControlA\} \quad (10)$$

Нововведена множина сутностей, які підходять для пошуку альтернативних варіантів відповіді, для деякого шаблону tt_i :

$$SCAdditionalA = \{c: VC(c) \cap SuffAdditionalA = \emptyset\} \quad (11)$$

$$STAdditionalA = \{t: VT(t) \in SuffAdditionalA\} \quad (12)$$

Для кожного завдання $task_i$ відбувається вибір шаблону tt_{task_i} [5]:

$$TTempl = \{tt_1, tt_2, \dots, tt_n\} - templates \quad (13)$$

$$tt_{task_i} \in TTempl \quad (14)$$

Множина допустимих ПТ-елементів для утворення контрольної ПТ -пари за шаблоном tt_{task_i} у випадку, коли $TQEntity(tt_{task_i}) = Thesis$ або

$TQEntity(tt_{task_i}) = Concept$, задана наступним виразом [5]:

$STQEntities(tt_{task_i})$

$$= \begin{cases} \left\{ \begin{array}{l} t: t \in T \wedge t \in STControlA \wedge \\ (TClass(t) \in TQClasses(tt_{task_i}) \wedge \\ \vee TQClasses(tt_{task_i}) = 0) \wedge \\ TClass(t) \notin TQNotClasses(tt_{task_i}) \end{array} \right\}, TQEntity(tt_{task_i}) = Thesis \\ \left\{ \begin{array}{l} c: c \in C \wedge c \in SCControlA \wedge \\ (CClass(t) \in TQClasses(tt_{task_i}) \wedge \\ \vee TQClasses(tt_{task_i}) = 0) \wedge \\ TClass(t) \notin TQNotClasses(tt_{task_i}) \end{array} \right\}, TQEntity(tt_{task_i}) = Concept \end{cases} \quad (15)$$

Випадковий елемент цієї множини позначимо $STQEntity$.

Для пошуку альтернативних варіантів відповіді окрім контрольної області, також буде використовуватись додаткова область контенту. Таким чином, коли $TQEntity(tt_{task_i}) = Thesis$ або $TQEntity(tt_{task_i}) = Concept$, множина ПТ-



елементів для утворення альтернатив задається виразом (15), який був модифікований з урахуванням додаткової області контенту:

$$STDEntities(tt_{task_i}) = \left\{ \left\{ \begin{array}{l} (t: t \in T \wedge t \in (STControlA \cup STAdditionalA) \wedge \\ (TClass(t) \in TQClasses(tt_{task_i})) \wedge \\ \forall TQClasses(tt_{task_i})=0 \\ TClass(t) \notin TQNotClasses(tt_{task_i}) \end{array} \right\}, TQEntity(tt_{task_i}) = Thesis \right. \\ \left. \left\{ \left\{ \begin{array}{l} (c: c \in C \wedge c \in (SControlA \cup SAdditionalA) \wedge \\ (CClass(t) \in TQClasses(tt_{task_i})) \wedge \\ \forall TQClasses(tt_{task_i})=0 \\ TClass(t) \notin TQNotClasses(tt_{task_i}) \end{array} \right\}, TQEntity(tt_{task_i}) = Concept \right\} \right\} \quad (16)$$

Випадковий елемент цієї множини позначимо *STDEntity*.

Відповідно змінився вираз оператора для пошуку альтернатив [5], в якому зараз врахована нова множина ПТ-елементів, яка була наведена вище (16):

$$WTAEntities(tt_{task_i}) = \{e: e \in E \wedge (TAClasses(tt_{task_i}) = \emptyset \vee \\ EClass(e) \in TAClasses(tt_{task_i}) \wedge \neg corresponds(e, STDEntity))\} \quad (17)$$

У шаблонах для завдань множинного вибору міститься інформація про необхідну кількість правильних і альтернативних варіантів. Поточний контент, по якому іде тестування, аналізується на предмет достатності даних для побудови завдання. У випадку, коли альтернативних варіантів у поточному вузлі ієрархії контенту недостатньо, буде зроблена спроба пошуку альтернатив, починаючи з батьківського вузла вглуб по ієрархії контенту. Загальна кількість варіантів відповідей описана відношенням:

$$TAOption(tt_i) = m, m \in Z \quad (18)$$

Кількість правильних варіантів відповідей:

$$TCOption(tt_i) = k, k \in Z \quad (19)$$

Кількість потрібних альтернативних варіантів відповідей для успішного створення завдання задана в наступному виразі:

$$distractors_required(tt_i) = TAOption(tt_i) - TCOption(tt_i) \quad (20)$$

Отже, умова недостатності альтернативних відповідей виглядає так:

$$|WTAEntities(tt_{task_i})| < distractors_required(tt_{task_i}) \quad (21)$$

Якщо вираз вище істинний, це сигналізує про наявну недостатність. У цьому разі здійснюється підйом по ієрархії контенту для залучення більшої кількості понять і тез.

Підйом по ієрархії контенту для залучення додаткових ПТ-елементів.

Ієрархія гілок контенту представляє з себе орієнтований граф-дерево. Кожна гілка містить тези і поняття. Приклад простої ієрархії гілок зображено нижче (рисунок 1).

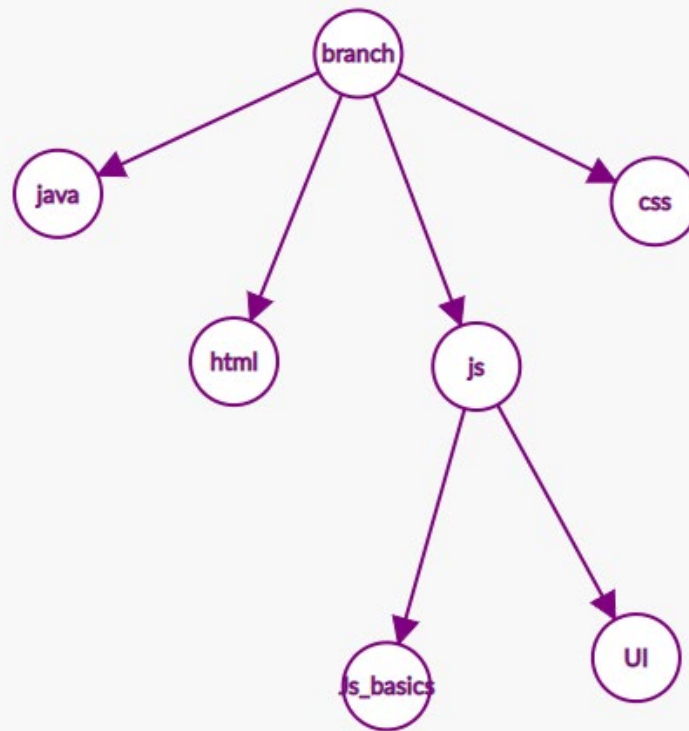


Рисунок 1 – Приклад графу ієрархії гілок контенту

Якщо була виявлена недостатність:

Для кожної гілки $branch_{child} \in (SuffAdditionalA \cup SuffControlA)$,

включаються гілки вище по ієрархії. Відповідно до ієрархічної структури кожен елемент дерева має один батьківський елемент, що задається відображенням [6]:

$$F: V \rightarrow V \quad (22)$$

Множину гілок контенту, які потрібно включити до додаткової області задано виразом:

$$BranchToInclude = \{branch \mid (branch_{child}, branch) \in F \wedge branch \notin (SuffAdditionalA \cup SuffControlA)\} \quad (23)$$

$$SuffAdditionalA_{n+1} = SuffAdditionalA_n \cup BranchToInclude \quad (24)$$

В результаті ми отримуємо розширений банк тез і понять для пошуку альтернативних варіантів відповіді. Після цього повторюється спроба створення тестового завдання. Якщо вдалося побудувати необхідну кількість завдань, то генерацію тесту можна вважати успішною.

Висновки

В цій роботі була розглянута модифікація методу автоматичної генерації тестових завдань на основі понятійно-тезисній моделі [5] шляхом введення нової області контенту для альтернативних варіантів відповіді та можливості генерації тестів по гілкам і курсам. Запропоновано застосування додаткової області контенту та вирішення проблеми недостатності альтернативних



відповідей шляхом підйому по ієрархії навчального контенту і залучення нових ПТ-елементів до множини потенційних дистракторів. Це вдосконалення збільшило кількість контрольних областей матеріалу по яким можлива успішна генерація тесту. Забезпечено можливість генерації завдань по адаптивним курсам і гілкам контенту, що збільшило гнучкість під час задання контрольної області контенту. Був розроблений програмний продукт, що використовує покращений метод генерації завдань. Він доступний на освітній платформі Semantic Portal [7].

Література:

1. Аванесов В.С. Склад тестових завдань. Навчальна книга для викладачів університетів, шкільних вчителів, аспірантів та студентів педагогічних університетів. 2 вид., Вип. Тощо. М.: Прим. 1998 р. - 217 с
2. Титенко С. В. Формалізація даних часопису на основі понятійно-тезисної моделі / С. В. Титенко, Астахов А. Г. // XVII Міжнародна наукова конференція імені Т.А. Таран «Інтелектуальний аналіз інформації» ИАИ 2017, Київ, 17–19 травня 2017 р.: зб. пр.– К. : Просвіта, 2017. – С. 243-247. ISBN 978–617–7010–13–4
3. Атанов Г. А. Навчання іскусственный інтелект, або основи сучасної дидактики вищої школи / Атанов Г. А., Пустиннікова І. Н. - Донецьк: Вид-во ДООУ, 2002.
4. Сова Дж. Ф. Концептуальні структури: обробка інформації в розумі та машині. Читання МА: Еддісон. - Веслі, 1984
5. Nerodenko V. Generation of tests of various complexity levels in e-learning system based on educational text formalization model / Nerodenko V., Tytenko S. // Modern Aspects of Software Development: Proceedings of VI International Scientific and Practical Virtual Conference of Software Development Specialists, June, 24 2019 р. – Kyiv: Igor Sikorsky KPI, 2019. – pp. 121-133.
6. Титенко, С. В. Модель навчального Web-контенту Tree-Net як основа для інтеграції керування знаннями і безперервним навчанням / С. В. Титенко, О. О. Гагарін // Системні дослідження та інформаційні технології. – 2009. – № 1. – С. 74–86.
7. Semantic Portal [Електронний ресурс] – Режим доступу до ресурсу: <http://semantic-portal.net>.

***Abstract.** The paper considers the method of automated construction of test tasks for e-learning systems, which is based on the concept-thesis model (CTM). Listed aspects of the method that can be improved to increase its efficiency. It is proposed to introduce new abstractions in the method to increase flexibility in creating control areas of educational material. It is proposed to improve the search for alternative answers by introducing an additional area of training material. As a result, a software system was developed that uses an upgraded method of automated task generation.*

***Key words:** e-learning, information-learning systems, ontology, test tasks, concept-thesis model, automatic generation.*

Стаття відправлена: 11.10.2021 г.

© Заїчко О.П., Круглий Д.В., Титенко С.В.