



АЛГОРИТМІЧНА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ЕЛЕКТРОННОГО ПІДРУЧНИКА “ОСНОВИ ТЕОРІЇ ЕЛЕКТРОННИХ КІЛ”

Юрій Бобало, Богдан Мандзій, Петро Стахів, Наталя Шаховська

Національний університет “Львівська політехніка”
вул. Степана Бандери, 12, м. Львів, 79013, Україна
e-mail: natalya233@gmail.com

Резюме: У роботі розглянуто досвід розроблення електронного підручника “Основи теорії електронних кіл”. Подано концепцію, принципи розроблення електронного підручника. Докладно описано процедуру перевірки та оцінювання отриманих знань. Побудовано алгоритм автоматичної перевірки на тестові завдання з формульним записом.

Ключові слова: електронний підручник, дистанційне навчання, віртуальне навчальне середовище.

ALGORITHMIC AND SOFTWARE IMPLEMENTATION OF E-TEXTBOOK “FUNDAMENTALS OF THEORY OF ELECTRONIC CIRCUITS”

Yuriy Bobalo, Bogdan Mandziy, Petro Stakhiv, Natalya Shakhovska

National University “Lviv Polytechnic”
12, Stepana Bandery Street, Lviv, 79013, Ukraine
e-mail: natalya233@gmail.com

Abstract: The paper presents the experience of developing electronic textbook “Fundamentals of the theory of electronic circuits”. The concept and development principles of electronic textbook are discussed. The procedure of knowledge control and evaluation is described in details.. The algorithm of automatic control the test tasks with possibilities of entering mathematical expressions is developed.

Keywords: electronic textbook, distance learning, virtual learning environment.

ВСТУП

Електронний підручник – комп’ютерний, педагогічний програмний засіб, призначений, в першу чергу, для подання нової інформації, який доповнює друковані видання, слугує для індивідуального й індивідуалізованого навчання і дозволяє в обмеженій формі тестувати отримані знання та вміння учня.

Розглянемо деякі елементи електронного підручника:

1. Електронний підручник повинен містити тільки мінімум текстової інформації у зв’язку з тим, що тривале читання тексту з екрану призводить до значного стомлення і, як наслідок, до зниження сприйняття та засвоєння знань. Електронний варіант підручника дозволяє

виділити окремі слова чи фрази кольором і фоном, що покращує наочність, дозволяє акцентувати увагу на головному.

2. Такі підручники повинні містити велику кількість ілюстративного матеріалу. Для обмеження обсягу підручника (розміру файлів) і, відповідно, можливості його використання на широкому спектрі комп’ютерів, доцільно використання упакованих форматів графічних файлів (GIF, JPEG та ін).

3. Використання відеофрагментів дає змогу передати в динаміці процеси і явища. Незважаючи на великі розміри файлів, застосовувати їх доцільно, оскільки сприйняття і зацікавленість студентів підвищуються і, як наслідок, поліпшується якість знань.

4. У традиційному навчанні переважають вербальні засоби при подання нового матеріалу. У зв'язку з цим застосування аудіо фрагментів в електронному підручнику дозволяє не тільки наблизити його до звичних способів подачі інформації, а й покращити сприйняття нового матеріалу, при цьому активізує не тільки зорові, але і слухові центри головного мозку.

5. Електронний підручник повинен містити гіперпосилання за елементами підручника.

6. Можливо, наприклад, за допомогою технології OLE, запуск інших комп'ютерних програм для показу прикладів, тестування та інших цілей.

Завдання, запропоновані студентам для самонавчання, можуть бути різного типу. Кожний тип завдання припускає певну діяльність студента при його виконанні та формуванні відповіді. Можна виділити наступні типи завдань:

- меню (вибір однієї вірної відповіді з декількох);
- обчислення (введення цілого, дійсного числа або декількох чисел);
- слово (введення одного або декількох слів);
- фраза (введення речення);
- формула (введення формули);
- відповідність (вибір вірної відповіді з меню для кожного із заданих речень, рисунків або фото);
- послідовність (введення послідовності дій або вибір її з меню).

Іншим методичним аспектом контролю є формування набору контрольних завдань. Залежно від вигляду і мети контролю можна виділити такі підходи до комплектування груп завдань [1]:

- послідовність питань і завдань різної складності і значущості, що оформлена у вигляді розгалуженого або багаторівневого алгоритму і може бути використана для поточного дослідження і самоконтролю рівня знань;
- спеціальний набір завдань різної складності, сформований для перевірки певного або комплексного рівня підготовки (знань, умінь, навичок), який пред'являється в заданій послідовності: довільно, в порядку самостійного вибору студента або в порядку зростання їх рівня складності;
- група завдань, що формуються за допомогою випадкової вибірки з банку

завдань з урахуванням динаміки і наповнення дистанційного курсу дисципліни та складності завдань.

Перераховані підходи можуть бути доповнені, а також скомбіновані один з одним.

2. АНАЛІЗ ЗАСОБІВ РОЗРОБЛЕННЯ ЕЛЕКТРОННИХ ПІДРУЧНИКІВ ТА ПОСТАНОВКА ЗАДАЧІ

Засоби розроблення електронних навчальних комплексів з'явилися в середині 70-х рр. ХХ ст. Були розраховані на роботу в багатокористувацькому режимі на універсальних і міні-ЕОМ. Використання в той час алфавітно-цифрових дисплеїв істотно обмежувало можливості графічної підтримки навчальних дисциплін. Навчання можливо було лише в спеціальних дисплейних класах навчального закладу. Однак у цей же час з'являється поняття гіпертексту, що дало змогу знизити затрати часу на розроблення розгалужених програм навчання. Різке зростання інструментальних засобів розроблення електронних навчальних комплексів спостерігається з кінця 80-х, початку 90-х рр. у зв'язку з широким розповсюдженням персональних комп'ютерів.

Однією з перших систем для електронного навчання на теренах СНД була адаптивна діалогова інформаційна система АДОНИС. Призначена для організації автоматизованого навчання і контролю знань з різних дисциплін, а також інформаційного забезпечення навчального процесу. В системі АДОНИС не використовувалися засоби гіпертексту, але структурні елементи її були організовані вдало. АДОНИС мала вбудований графічний редактор для розроблення ілюстративного матеріалу. Ця система не була адаптована до ОС WINDOWS, і в даний час не використовується. Однак, ідеї АДОНИС, знайшли своє втілення в інших системах, наприклад "TrainingWare", "eLearning Server 3000 v2.0", "eLearning Office 3000" та "HyperMethod 3.5".

TrainingWare призначена для організації централізованої системи підготовки та контролю знань персоналу, автоматизації вхідного тестування, регулярних тренінгів та атестацій, створення єдиної системи обліку компетенції персоналу, сертифікації клієнтів і партнерів компанії. Орієнтована на великі та середні компанії, чисельністю від 300 осіб, що мають розгалужену структуру і відчувають необхідність у безперервному управлінні знаннями співробітників.

eLearning Server 3000 v2.0 дає змогу створювати власні Навчальні центри в Інтернет / Інтранет та організувати повний цикл дистанційного навчання – управління розкладом, сертифікацією знань учнів, ведення електронних відомостей, електронної залікової книжки, електронної бібліотеки. eLearning Office 3000 призначений, насамперед, для викладачів вищих і середніх навчальних закладів, а також для ІТ-фахівців, що займаються проблемами дистанційного навчання.

HyperMethod 3.5 – конструктор мультимедіа застосувань, який призначений для швидкого та ефективного створення інформаційних систем, презентаційних дисків, електронних підручників, довідників, енциклопедій і будь-яких інших мультимедіа додатків. Проте він не дуже вдалий для навчання технічним спеціальностям, де необхідно проводити тестування знань з врахуванням символічних записів.

Moodle – безкоштовна, відкрита система дистанційного навчання. Система реалізує філософію “педагогіки соціального конструктивізму” та орієнтована насамперед на організацію взаємодії між викладачем та учнями, хоча підходить і для організації традиційних дистанційних курсів, а також підтримки очного навчання. Moodle перекладена на десятки мов, в числі є і переклад на українську. Національний університет “Львівська політехніка” широко використовує цю систему для розміщення навчальних матеріалів. Також потрібно зазначити, що цей проект є відкритим [5].

ATutor є системою керування навчальним матеріалом (Learning Content Management System, LCMS). Програма є простою у встановленні, налаштуванні та підтримці для системних адміністраторів; викладачі можуть досить легко створювати та переносити навчальні матеріали та запускати свої онлайн-курси. А оскільки система є модульною, то вона відкрита для модернізації і розширення функціональних можливостей. Використовується в Іспанії, Болгарії, Сербії [6].

Claroline є спільною електронною навчальною платформою (Learning Management System) випущеною під ліцензією Open Source. Платформа використовується більш ніж в 80 країнах та перекладена більше ніж на 30 мов [7].

eCollege – програмне забезпечення як послуга (SaaS), постачальник електронного навчання програмного забезпечення і послуг для середніх та вищих навчальних закладів [8].

Метою роботи є проектування архітектури та

розроблення електронного підручника з дисципліни “Основи теорії кіл”.

3. АЛГОРИТМІЗАЦІЯ КОНТРОЛЮ РІВНЯ ЗАСВОЄННЯ ЗНАТЬ НА ПРИКЛАДІ НАВЧАЛЬНІ ДИСЦИПЛІНИ “ОСНОВИ ТЕОРІЇ ЕЛЕКТРОННИХ КІЛ”

Концепція підручника “Основи теорії електронних кіл”, яку розробили спільно викладачі Львівської та Київської політехнік, за редакцією професора Ю. Я. Бобала базується на врахуванні таких факторів: комплексності навчального процесу, яка полягає в інтегрованості різних форм занять – лекцій, практичних, лабораторних, самостійних; відповідності обсягу завдань кількості виділених кредитів; важливості самоконтролю знань і ефективності підготовки до виконання лабораторних робіт.

Особливістю програмного засобу, що використовується для самонавчання в дисципліні “Основи теорії електронних кіл”, є необхідність відображення грецьких символів у стандартному вікні Windows, формування відповіді у вигляді формул, таблиць, наявність завдань, що передбачають кілька варіантів відповіді тощо.

Процес самонавчання передбачає наявність двох складових:

1. Навчального матеріалу M , який містить теоретичні викладки. Організований у вигляді дерева, у якому глибина відповідає рівню вкладеності тем M_i та підтем M_{ij} . Характеризується кількістю тем (підтем) $|M_{ij}|$ та часом, відведеним викладачем на самонавчання t_i , кількістю запитань певних рівнів складності Q_{ik} та відведеними балами R_{ik} .

2. Тестових завдань T , які містять питання Q_{t_i} , варіанти відповідей V_{ij} , рівні складності Sk_i та типи завдання Tq_i :

$$E = \langle M, T \rangle, M = \langle M_i, \{M_{ij}\}, t_i, \{Q_{ik}, R_{ik}\} \rangle,$$

$$T = \langle Q_{t_i}, \{V_{ij}\}, Sk_i, Tq_i \rangle$$

До кожної теми викладач вказує кількість питань певної складності та відводить час на виконання завдань. Студенту виводиться випадковим чином сформований набір тестових запитань. Результат тестування є визначним для вибору наступного елементу навчального матеріалу. Визначення наступної теми для

вивчення здійснюється з допомогою I-АБО графу. Формування послідовності тем для вивчення здійснюється на основі обходу I-АБО графу в ширину. АБО-вершина відвідується, коли студент набрав 0-39 балів – обов'язково, а коли 40-70 – пропонується пройти 2 варіант тесту для уточнення його рівня знань.

Алгоритм формування та структурування навчального матеріалу складається з таких кроків:

1. Студент обирає розділ для вивчення серед вершин з найбільшим пріоритетом. Складові розділу вивчаються у залежності до дуг, які виходять з нього (І – необхідно вивчити у послідовності спадання оцінок, АБО – один з найбільшою оцінкою або усі складові за бажанням).
2. Після вивчення елементу M_{ij} , що відповідає одній з вершин, відбувається тестування. Отримана оцінка визначатиме послідовність подальшого вибору тем:
 - незадовільна – вивчення наступного за пріоритетом параграфа розділу;
 - задовільна – пріоритет розділу залишається незмінним. Студентові пропонується вивчення параграфів, що залишилися нерозглянутими, або перехід до вивчення наступних розділів;
 - добра – пріоритет розділу змінюється за формулою, використовуючи операцію багатозначного АБО та перехід до вивчення наступних розділів з максимальним значення між найнижчою оцінкою серед вершин та різницею поточної оцінки й найнижчої. Перехід до наступних розділів;
 - відмінна – присвоєння вивченому розділу найнижчої оцінки та перехід до вивчення наступних розділів
3. Визначення загального часу вивчення t .
4. Зупинка виконання процедури обходу графу здійснюється за наступних умов:
 - Навчання перерване студентом. У такому випадку його оцінка визначається на основі уже отриманих оцінок:

$$Rate = \frac{\sum Q_{ik} R_i}{\sum Q_{ik} R_{ik}},$$

де R_i – оцінка, отримана за результатами тестування по розділу;

– Вичерпано загальний час навчання.

Розроблено такі алгоритмічні блоки:

1. Розбір арифметичного виразу та його автоматичний розрахунок – для реалізації питань-задач на основі згенерованих параметрів;
2. Формування алфавіту спеціальних символів, їх опрацювання, перемальовування вікна для відображення символів та введення формульних виразів з клавіатури;
3. Визначення типу відповіді, запитання та відображення їх у вигляді тексту, формули, малюнку, таблиці чи змішано;
4. Перевірка правильності формування тестового завдання – аналіз на наявність помилок з боку викладача – вказання неправильної кількості варіантів відповідей, неіснуючого типу запитання тощо;
5. Формування випадковим чином банку питань для студента з врахуванням заданої викладачем кількості питань за рівнями складності;
6. Перевірка правильності відповіді студента;
7. Визначення сумарної оцінки студента;
8. Налаштування таймеру тестування;
9. Визначення суміжних тем, які необхідно вивчити студентом для досягнення кращого результату.

Опишемо розбір арифметичного виразу та його автоматичний розрахунок.

Лексичний аналіз – процес аналітичного розбору вхідної послідовності символів з метою отримання на виході послідовності лексем – найменших синтаксичних одиниць, які мають значення [2 – 3]. Синтаксичний аналіз (парсинг) – процес зіставлення лінійної послідовності лексем (слів, токенів) мови з її формальної граматикою. Результатом його є деяка структура даних – наприклад синтаксичне дерево розбору. Як правило, застосовується спільно з лексичним аналізом. Синтаксичний аналізатор (парсер) – програма або частина програми, що виконує синтаксичний аналіз.

Розрізняють два типи алгоритмів синтаксичного аналізу:

- спадний синтаксичний аналіз – завдання побудови дерева розбору для вхідної стрічки, починаючи з кореня і створюючи вузли дерева розбору в прямому порядку обходу або, що те ж саме, спадний синтаксичний аналіз можна розглядати як пошук лівого породження вхідної стрічки.
- висхідний синтаксичний аналіз буде деревом розбору для вхідної стрічки,

починаючи з листів (знизу) і йдучи у напрямку до кореня (вгору).

Програма спадного синтаксичного аналізу, або аналізу методом рекурсивного спуску (recursive-descent parsing) складається з набору процедур, по одній для кожного нетерміналу. Робота програми починається з виклику процедури для початкового символу і успішно закінчується в разі сканування всього вхідного рядка.

Загальний вигляд висхідного синтаксичного аналізу, відомий як синтаксичний аналіз типу “перенесення/згортання” (shift-reduce) – процес “згортання” стрічки до стартового символу граматики. На кожному кроці згортання (Reduction) визначений підрядок, що відповідає тілу продукції, замінюється нетерміналом із заголовка цієї продукції. Ключові рішення, що приймаються в процесі висхідного синтаксичного аналізу, – коли виконувати згортку і яку продукцію застосовувати.

Для опису вхідних даних для тестового завдання, відповідь на яке формується формулою, використовуємо нотацію Бекуса-Наура:

```

<змінна> ::= 'v' <цифра>
<число> ::= <цифра> {<цифра>}
<ціле число> ::= [<знак>] <число>
<знак> ::= + | -
<кома> ::= ,
<розділювач> ::= ;
<цифра> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<дробове число> ::= <число> [ <кома> <ціле число> ]
<операція> ::= <знак> | * | /
<функція> ::= sin | cos | exp | ln | sqrt | sqr | tan | atan
<дужка вх> ::= (
<дужка вих> ::= )
<лексема> ::= <функція> <дужка вх>
<дробове число> <дужка вих> |
<функція> <дужка вх> <змінна> <дужка вих> |
[ <дужка вх> ] <дробове число> <операція>
<дробове число> [ <дужка вих> ] |
[ <дужка вх> ] <змінна> <операція> <змінна>
[ <дужка вих> ] |
[ <дужка вх> ] <змінна> <операція> <дробове
число> [ <дужка вих> ] |
[ <дужка вх> ] <дробове число> <операція>
<змінна> [ <дужка вих> ]
<складена лексема> ::= [ <дужка вх> ] <лексема>
<операція> <лексема> [ <дужка вих> ] | [ <дужка вх> ]
<функція> <дужка вх> <лексема> <дужка вих>
[ <дужка вих> ]
<вираз> ::= { <складена лексема> }
<параметр> ::= <дробове число> ‘-‘ <дробове
число> <розділювач> { <дробове число> ‘-‘

```

<дробове число> <розділювач> }

Для прикладу, вираз $\frac{\sin(x-y)z}{w}$, де

параметри змінюються в межах

X=20..70, Y=0..20, Z=10..90, W=1..100,

описується як :

$\sin(\sqrt{3}-\sqrt{4}) * \sqrt{1}/\sqrt{2}$

10-90;1-100;20-70;0-20;

Автоматичне визначення значення виразу здійснюється за таким алгоритмом:

1. Зчитування ВИРАЗу і ПАРАМЕТРу.
2. Формування динамічного масиву за кількістю елементів ПАРАМЕТРу.
3. Заповнення елементів утвореного масиву випадковими значеннями, межі генерації яких вказані для кожного елемента.
4. Методом рекурсивного спуску заміняємо по чергово ЗМІННу у ВИРАЗі на значення з утвореного масиву.
5. Побудова синтаксичного дерева розбору

Для відображення грецьких символів у стандартному Windows-вікні використано регулярні вирази і перевантаження основного класу TLabel. Регулярний вираз- це рядок, що описує множину рядків відповідно до набору спеціальних синтаксичних правил. Регулярні вирази використовуються в багатьох текстових редакторах та допоміжних інструментах для пошуку та заміни тексту на основі заданих шаблонів. Багато мов програмування підтримують регулярні вирази для роботи з рядками. В тестовій системі використано таблицю квантифікаторів (табл. 1). Саме за допомогою квантифікаторів визначаємо правильну кількість повторень.

Таблиця 1. Квантифікатори

Позначення	Число повторів
*	Нуль або більше
+	Одну або більше
?	Нуль або один раз
{n}	Точно n відповідностей
{n,}	n або більше відповідностей
{n,m}	Мінімум n та максимум m відповідностей

Далі визначено принцип представлення спеціальних символів у стандартному Windows-вікні:

1) Спеціальні символи та грецькі букви позначати таким чином (позначення починається з косої риски):

\Downarrow - \Downarrow, \Leftarrow - \LeftArrow, \Leftrightarrow - \Leftrightarrow, \Leftrightarrow - \Leftrightarrow, \Leftarrow - \LeftArrow, \Uparrow - \Uparrow, \downarrow - \downarrow, \leftarrow - \leftarrow, \uparrow - \uparrow, \rightarrow - \rightarrow, \cup - \cup, \cap - \cap, \subset - \subset, \subseteq - \subseteq, \supset - \supset, \vee - \vee, \wedge - \wedge, \angle - \angle, \approx - \approx, \cong - \cong, \equiv - \equiv, \bullet - \bullet, \cdot - \cdot, \perp - \perp, \in - \in, ∞ - \infty, \notin - \notin, \geq - \geq, \div - \div, \oplus - \oplus, \oslash - \oslash, \otimes - \otimes, \pm - \pm, \times - \times, Δ - \Delta, Γ - \Gamma, Λ - \Lambda, Ω - \Omega, Φ - \Phi, Π - \Pi, Ψ - \Psi, Σ - \Sigma, Θ - \Theta, Υ - \Upsilon, Ξ - \Xi, α - \alpha, β - \beta, χ - \chi, δ - \delta, \diamond - \diamond, ε - \varepsilon, η - \eta, γ - \gamma, ι - \iota, κ - \kappa, λ - \lambda, \leq - \leq, μ - \mu, \neq - \neq, ν - \nu, ω - \omega, ∂ - \partial, ϕ - \phi, π - \pi, ψ - \psi, ρ - \rho, σ - \sigma, τ - \tau, θ - \theta, υ - \upsilon, ε - \varepsilon, φ - \varphi, ϖ - \varpi, ς - \varsigma, ξ - \xi, ζ - \zeta, \setminus , \backslash , $\} - \}$

2) Формат символів:

верхній індекс \wedge , нижній індекс $_$, \it – курсив, \bl – жирний, \ul – підкреслений, \st – перекреслений, \gm – закінчення формату символів.

3) Для позначення адреси мультимедійного файлу використовується символ $@$. Наприклад: $@\img\1.bmp@$

4. ПРОЕКТУВАННЯ СИСТЕМИ КОНТРОЛЮ РІВНЯ ЗАСВОЄННЯ ЗНАТЬ НА ПРИКЛАДІ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ “ОСНОВИ ТЕОРІЇ ЕЛЕКТРОННИХ КІЛ”

Діаграма прецедентів – діаграма, на якій зображено відношення між акторами та прецедентами в системі. Прецеденти є основним засобом визначення необхідної поведінки системи. Як правило, вони використовуються для описання вимог до системи, тобто, що має робити система. Основними поняттями, пов’язаними з прецедентами в розробленій діаграмі є актори такі як: Користувач, та Сервер БД. Використано прецеденти такі як: Реєстрація, Логування, Тестування, та суб’єкт. Суб’єкт – це система, що розглядається і до якої відносяться прецеденти, тобто система дослідження знань. Користувачі та будь-які інші системи, що можуть взаємодіяти із суб’єктом, представлено як акторів (рис. 1).

Будову та взаємодію компонентів відображає діаграма компонентів (рис. 2). Зв’язки між окремими компонентами зведені до мінімуму завдяки використанню узагальнених інтерфейсів

та обміну даними за допомогою подій. Таким чином досягається легкість заміни окремих компонентів.

5. РОЗРОБЛЕННЯ СИСТЕМИ КОНТРОЛЮ РІВНЯ ЗАСВОЄННЯ ЗНАТЬ НА ПРИКЛАДІ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ “ОСНОВИ ТЕОРІЇ ЕЛЕКТРОННИХ КІЛ”

Розроблена система в середовищі Delphi 7 з використанням переважаних стандартних елементів інтерфейсу. Після запуску користувач бачить перед собою вікно, подане на рис. 3.

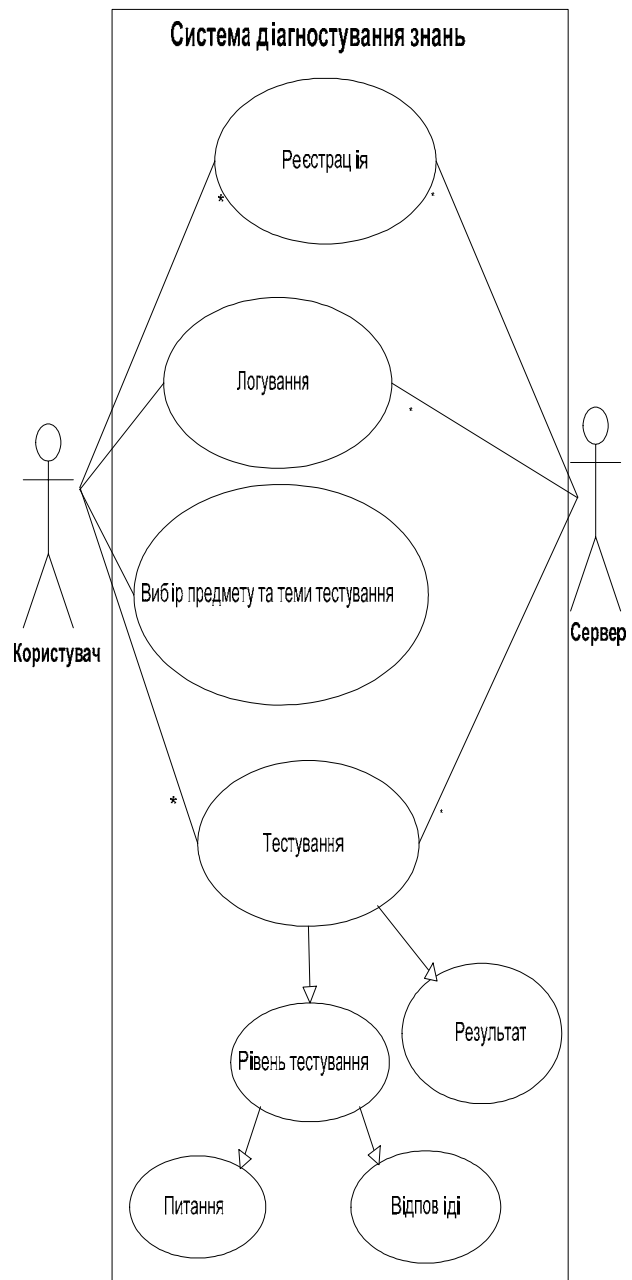


Рис. 1 – Діаграма прецедентів

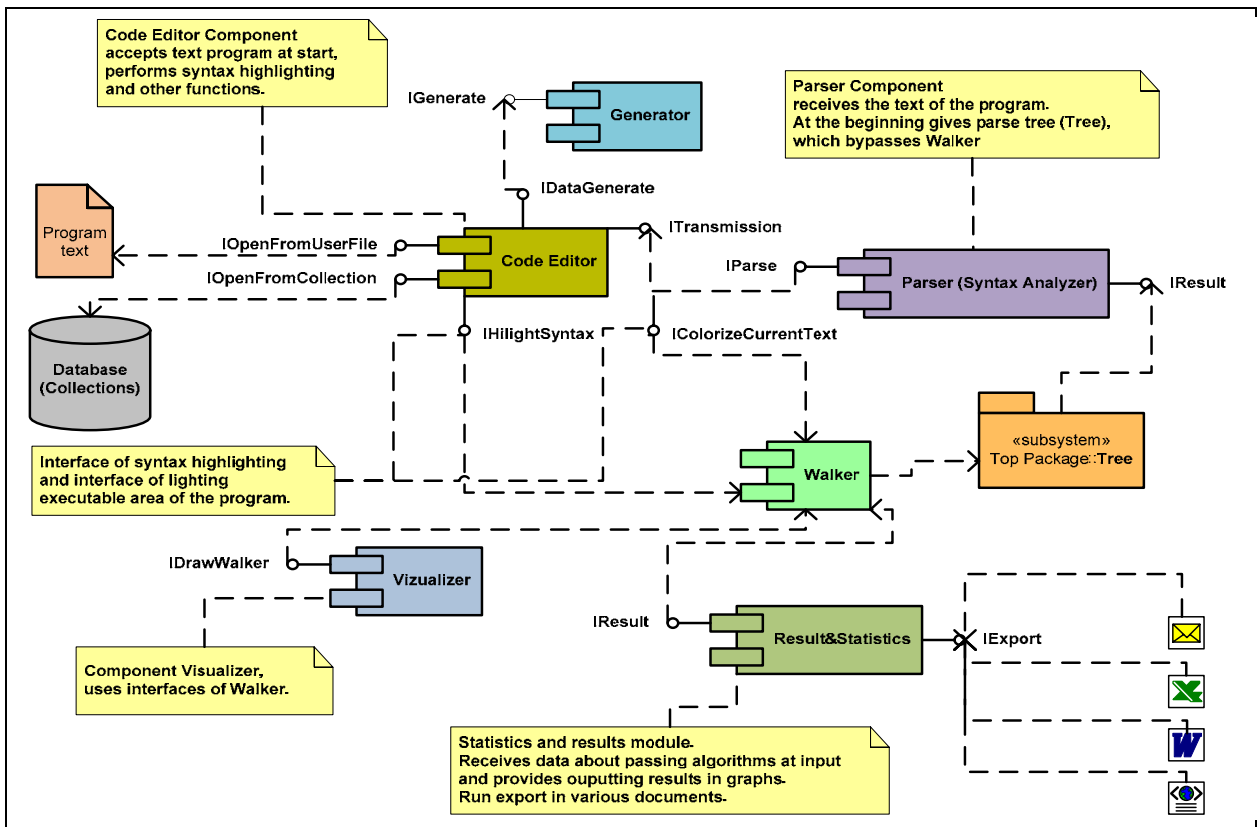


Рис. 2 – Діаграма компонентів системи

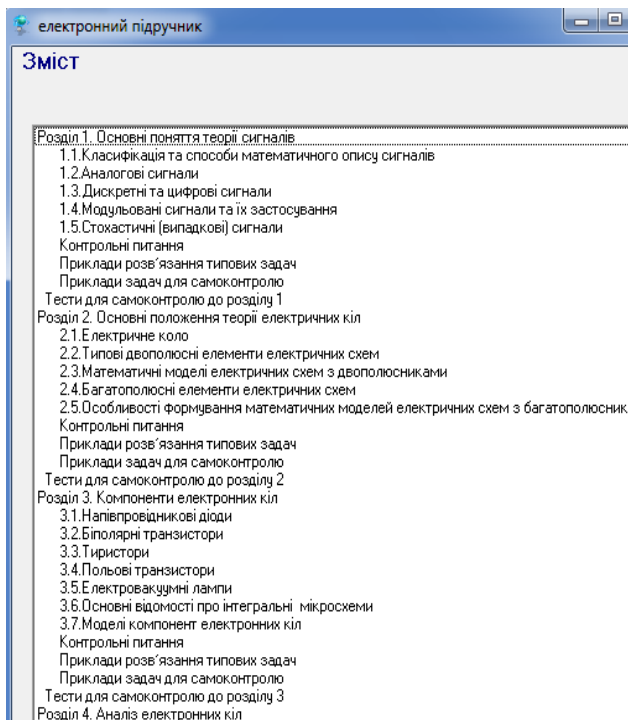


Рис. 3 – Головне вікно програми

У ньому вміщено перелік тем дисципліни основ теорії електричних кіл, контрольні питання до розділів, приклади задач для самоконтролю, а також тести та задачі для самонавчання. Вибір необхідної теми здійснюється шляхом подвійного клацання на назві. Короткий текст підтем,

приклади розв'язування задач, перелік контрольних питань відображається у браузері (рис. 4).

1. Основні поняття теорії сигналів

1.1 Класифікація та способи математичного опису сигналів

Детерміновані та випадкові сигнали - дві основні групи сигналів.

Первинне повідомлення можна перетворити в один із чотирьох можливих типів сигналів: аналоговий, дискретизований, квантований, цифровий.

Аналогова та цифрова електроніка - два напрями інформаційної електроніки.

Способи математичного опису реальних сигналів:

Рис. 4 – Вікно для відображення тексту теми

Для початку тестування подвійним клацанням обирається необхідний тест за розділами. Відкривається вікно для тестування (рис. 5).

У верхній частині вікна вказано номер питання, рівень його складності та кількість балів, яку можна отримати за правильну відповідь, у нижній розміщений бігунок часу.

Коли бігунок досягає правого краю форми, тест автоматично припиняється. Кнопки ВПЕРЕД, НАЗАД дозволяють переходити на наступне питання або повертатись до попереднього. Також користувач може бачити кількість балів, які він отримав за відповідь.

Обрання варіанту відповіді здійснюється шляхом позначення прапорця.

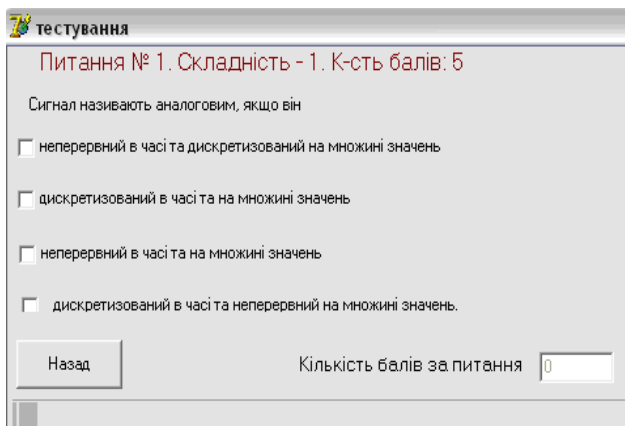


Рис. 5 – Вікно для тестування

Деякі питання вимагають введення відповіді з клавіатури (рис. 6). При розділювач цілої та дробової частини вводиться як КОМА, точність вказується у тексті запитання.

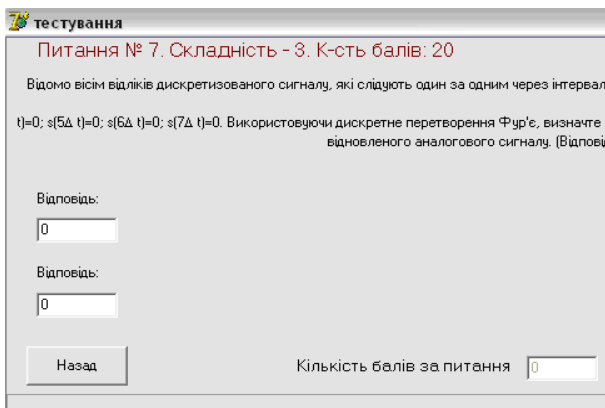


Рис. 6 – Питання, яке вимагає введення відповіді з клавіатури

Також передбачено заповнення масивів певної розмірності (рис. 7). Введення у комірки масиву здійснюється шляхом клацання мишкою на відповідній комірці та введенням необхідного значення з клавіатури.

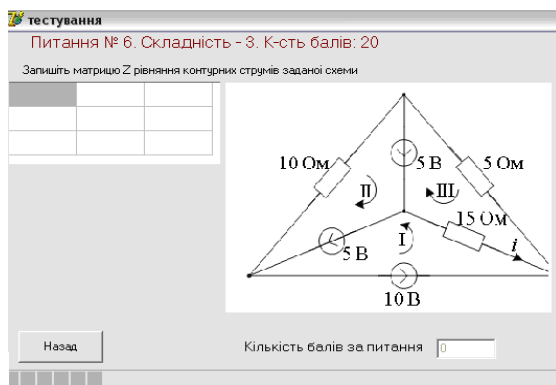


Рис. 7 – Питання, відповідь на яке подається у вигляді матриці

Питання-задачі містять схему електричного кола, його параметри, а також їх значення. Значення параметрів генеруються автоматично і здійснюється формульне розв'язання задачі. Після цього студент вносить свій варіант відповіді. Послідовність списку параметрів відповідає послідовності списку значень (рис. 8).

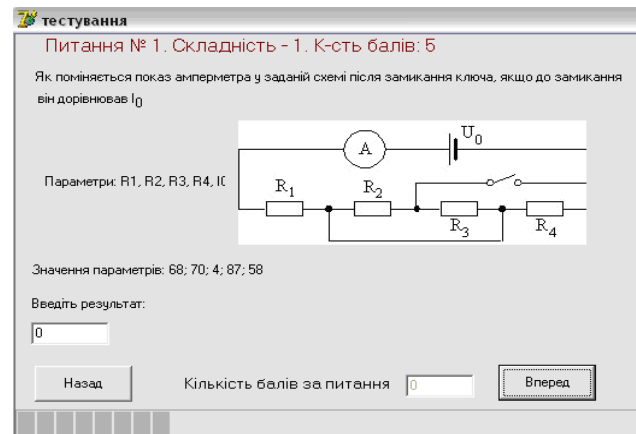


Рис. 8 – Питання-задачі

Якщо користувач відповідає на всі питання, а час ще не завершений, то він може повернутися до попередніх питань.

Результат тестування відображається у такій формі (рис. 9). Тут вказано кількість балів, яку отримав студент (максимально можна отримати 100 балів), а також номера тем чи підтем, які було б варто засвоїти краще.

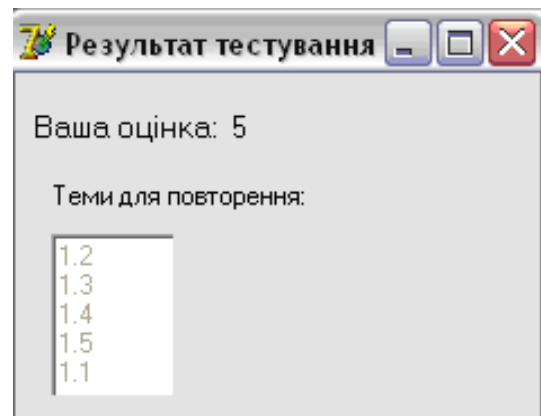


Рис. 9 – Форма з результатом

ВИСНОВКИ

Розроблений підручник демонструє новий підхід до навчально-методичного забезпечення для вивчення базових дисциплін. Авторському колективу вдалося об'єднати усі види занять з дисципліни "Основи теорії електронних кіл" і подібних дисциплін, які читають у Львівській політехніці, в єдиному підручнику з компакт-диском, який забезпечує вивчення теоретичного

матеріалу з використанням сучасних методів доступу до інформації, засвоєння практичних навиків за допомогою як традиційних підходів, так і комп'ютерних методик, а також виконання лабораторних завдань на основі імітаційного моделювання. Електронна частина також дає змогу здійснити самостійне оцінювання знань в інтерактивному режимі.

СПИСОК ЛІТЕРАТУРИ

- [1] M. Elsom-Cook, Student modelling in intelligent tutoring systems, *Artificial Intelligence Review*, (7) 3-4 (1993), pp. 227-237.
- [2] D. Gris, *Designing of Compilers for Digital Computing Engines*, Moscow: Mir, 1975, 544 p. (in Russian)
- [3] M. Gross, A. Lanten, *Theory of Formal Grammars*, Moscow: Mir, 1971, 294 p. (in Russian)
- [4] Yu.Ya. Bobalo, B.A. Mandziy, P.G. Stakhiv, L.D. Pysarenko, Yu.I. Yakymenko, *Bases of Electrical Circuits Theory*, Lviv: Lviv Polytechnics Publishing, 2008. (in Ukrainian)
- [5] Moodle, [Online], <http://moodle.org/mod/forum/discuss.php?d=133543>
- [6] ATutorLearning Management Tools, [Online], <http://atutor.ca/atutor/docs/index.php>
- [7] Claroline, [Online], <http://demo.claroline.net/index.php>
- [8] E-college, [Online], www.e-college.ru/education/speciality/manager.html



Юрій Бобало, д.т.н., професор, ректор Національного університету "Львівська політехніка".

Наукові інтереси зосереджені в галузі надійнісного проектування радіоелектронної апаратури.



Богдан Мандзій, д.т.н., професор, професор кафедри теоретичної радіотехніки та радіовимірювань Національного університету "Львівська політехніка".

Наукові інтереси: аналіз нелінійних електронних кіл, методи забезпечення надійності радіоелектронних пристроїв.



Петро Стахів, д.т.н., професор, завідувач кафедри теоретичної і загальної електротехніки Національного університету "Львівська політехніка".

Наукові дослідження ведуться в рамках наукової школи "Математичне моделювання динамічних процесів складних електричних кіл та електротехнічних систем".



Наталія Шаховська, д.т.н., доцент, декан базової вищої освіти Інституту комп'ютерних наук та інформаційних технологій Національного університету "Львівська політехніка".

Наукові інтереси: бази та сховища даних, інтеграція, розподілені системи.



ALGORITHMIC AND SOFTWARE IMPLEMENTATION OF E-TEXTBOOK “FUNDAMENTALS OF THEORY OF ELECTRONIC CIRCUITS”

Yuriy Bobalo, Bogdan Mandziy, Petro Stakhiv, Natalya Shakhovska

National University “Lviv Polytechnic”
12, Stepana Bandery Street, Lviv, 79013, Ukraine
e-mail: natalya233@gmail.com

Abstract: *The paper presents the experience of developing electronic textbook “Fundamentals of the theory of electronic circuits”. The concept and development principles of electronic textbook are discussed. The procedure of knowledge control and evaluation is described in details.. The algorithm of automatic control the test tasks with possibilities of entering mathematical expressions is developed.*

Keywords: *electronic textbook, distance learning, virtual learning environment.*

1. INTRODUCTION

Electronic textbook – computer, educational software tool intended primarily for the submission of new information that complements the printed publication, serves individual and personalized learning and provides a limited way to test the knowledge and skills of the student.

The tasks proposed to students in Electronic textbook may be different types. Each type of task involves a student’s activity during his performance and decision making. You can distinguish the following types of tasks:

- menu (select one correct answer for several);
- calculation (input field, real number or several numbers);
- word (enter one or more words);
- phrase (input sentence);
- formula (input formula);
- compliance (correct answer choice from the menu bar for each set of sentences, pictures or photos);
- sequence (input sequence of actions or choose it from the menu).

Depending on the type and purpose of control are the following approaches to completing tasks e-textbook groups:

- the sequence of questions and tasks of varying complexity and importance that is framed in a branched or multi-level algorithm and can be used for current research and self-awareness;
- a special set of tasks of varying complexity, generated to test specific or comprehensive level of training (knowledge, skills), which is

presented in a given sequence: randomly, by way of self-selection of the student or in order of increasing their level of complexity;

- group tasks, which are formed by random sampling from bank task considering the dynamics and distance course discipline and complexity.

So important is the creation of methodological and instrumental bases of simulation training systems management and control knowledge, adequate current trends in information technology and didactic principles of organizing and conducting training.

The popular systems fro e-textbook developing are “TrainingWare”, “eLearning Server 3000 v2.0”, “eLearning Office 3000”, “HyperMethod 3.5”.

2. THE MAIN PART

The concept of the textbook “Fundamentals of the theory of electronic circuits”, which jointly developed in the faculty of Lviv and Kyiv Polytechnic, edited by Professor Yu. Bobalo based on considering the following factors: complexity of the learning process, which is the integration of various forms of activities – lectures, practical laboratory, independent, under the volume of tasks allocated credits; importance of self-knowledge and effectiveness of training for laboratory work.

The feature of the software tool that used for learning in the course “Fundamentals of the theory of electronic circuits”, it is necessary to display Greek characters in the standard window Windows, the answer in the form of formulas, tables,

availability exercises, that providing some answers and more.

The process of learning involves the presence of two components:

1) Training material M – has short or full theoretical calculations. It is organized as a tree, which corresponds to the depth of nesting of topics M_i and sub-topics M_{ij} . It is characterized by the number of sub-topic $|M_{ij}|$, and time t_i allotted for teacher learning, questions of certain levels of difficulty Q_{ik} and points cushion R_{ik} .

2) Tests T – consist of the questions Qt_i , answers V_{ij} , difficulty level Sk_i and type of task Tq_i .

The teacher indicates the number of certain complexity for each theme and gives time to perform tasks. If the allotted time is finished and student does not answer all the questions, the test automatically stops, and program puts scores obtained by those questions that the student had answered.

Student can see randomly generated set of test questions:

- if a student takes 71-100 points – then it is recommended him to start learning the next section;
- if a student takes 40-70 points – then it is offered to take Option 2 to clarify his level of knowledge;
- if a student takes 0-39 – it is recommended to repeat the whole section.

In the program there are developed the following modules:

1) Analysis of arithmetic expression and its automatic calculation – for the implementation of issues-based problems generated by parameters;

2) The formation of special alphabet characters, their processing, redrawing the window to display the characters and the introduction of expressions from the keyboard;

3) Determine the type of questions and answers and display them in the form:

- text,
- formulas,
- figures,
- table,
- mixed;

4) Validation of tests – Analysis of errors by the teacher – specification of the number of incorrect answers, non-existent type of question, etc.;

5) Formation of random questions bank, that consider a given number of questions by level of difficulty for a student;

6) Check the correct answers a student;

7) Determination of the total assessment of student;

8) Setting the time of testing;

9) Definition of related topics that is necessary to examine the student to achieve better results.

To describe the input data for the test task, the answer to which is formed formula, we use the notation Backus-Naur.

Automatic detection of the expression is carried out by the following algorithm:

1. Reading the terms and parameters.

2. Formation of a dynamic array element number of parameters.

3. Filling array elements formed by random values the limits specified for the generation of each element.

4. The method of recursive descent alternatively replace variables in an expression formed by the value of the array.

5. Building syntactic parse tree.

To display Greek characters in a standard Windows-box uses regular expressions and overload the main class TLabel.

Regular expression is a string that describes or matches a set of strings, according to a set of special syntax. Regular expressions are used in many text editors and supporting tools for finding and changing text based on preset templates. Many programming languages support regular expressions for string manipulation. For example, Perl has a powerful mechanism for working with regular expressions, built directly in its syntax. In the test system will be widely used in regular expressions, so for a more detailed description of them is given a table of quantitative.

The system is designed in Delphi 7 using congested standard interface.

3. CONCLUSION

The tutorial demonstrates a new approach to the teaching of basic subjects for study. Electronic parts also allows you to make an independent assessment of knowledge online.

4. REFERENCES

- [1] M. Elsom-Cook, Student modelling in intelligent tutoring systems, *Artificial Intelligence Review*, (7) 3-4 (1993), pp. 227-237.
- [2] D. Gris, *Designing of Compilers for Digital Computing Engines*, Moscow: Mir, 1975, 544 p. (in Russian)
- [3] Yu.Ya. Bobalo, B.A. Mandziy, P.G. Stakhiv, L.D. Pysarenko, Yu.I. Yakymenko, *Fundamentals of the Theory of Electrical Circuits*, Lviv: Lviv Polytechnics Publishing, 2008. (in Ukrainian)