

Arnold E. Kiv
Serhiy O. Semerikov
Vladimir N. Soloviev
Andrii M. Striuk
(Eds.)



Computer Science & Software Engineering

**Proceedings of the 1st Student Workshop,
CS&SE@SW 2018**

Kryvyi Rih, Ukraine
November 30, 2018

Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (Eds.): Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, online

This volume represents the proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), held in Kryvyi Rih, Ukraine, in November 30, 2018. It comprises 20 contributed papers that were carefully peer-reviewed and selected from 25 submissions. The accepted papers present the ideas and early results of master's and PhD projects.

Program Committee

Albert A. Azaryan, Kryvyi Rih National University, Kryvyi Rih, Ukraine
Dmitry I. Ignatov, Higher School of Economics, Moscow, Russia
Arnold E. Kiv, Ben-Gurion University of the Negev, Beer Sheva, Israel
Yaroslav V. Shramko, Kryvyi Rih State Pedagogical University, Kryvyi Rih, Ukraine
Vladimir N. Soloviev, Kryvyi Rih State Pedagogical University, Kryvyi Rih, Ukraine
Oleksandr V. Spivakovskiy, Kherson State University, Kherson, Ukraine
Aleksandr D. Uchitel, National Metallurgical Academy of Ukraine, Dnipro, Ukraine

Local Organization Committee Members

Ihor A. Kotov, Kryvyi Rih National University, Kryvyi Rih, Ukraine
Pavlo V. Merzlykin, Kryvyi Rih State Pedagogical University, Kryvyi Rih, Ukraine
Iryna S. Mintii, Kryvyi Rih State Pedagogical University, Kryvyi Rih, Ukraine
Natalia V. Moiseienko, Kryvyi Rih State Pedagogical University, Kryvyi Rih, Ukraine
Serhiy O. Semerikov, Kryvyi Rih State Pedagogical University, Kryvyi Rih, Ukraine
Svitlana V. Shokaliuk, Kryvyi Rih State Pedagogical University, Kryvyi Rih, Ukraine
Andrii M. Striuk, Kryvyi Rih National University, Kryvyi Rih, Ukraine
Viktor Yu. Zubkevich, Kryvyi Rih National University, Kryvyi Rih, Ukraine

Additional Reviewers

Oleksandr Yu. Burov, Institute of Information Technologies and Learning Tools of NAES of Ukraine, Kyiv, Ukraine
Maya V. Popel, Institute of Information Technologies and Learning Tools of NAES of Ukraine, Kyiv, Ukraine
Natalya V. Rashevskaya, Kryvyi Rih National University, Kryvyi Rih, Ukraine
Mariya P. Shyshkina, Institute of Information Technologies and Learning Tools of NAES of Ukraine, Kyiv, Ukraine
Oleg M. Spirin, Institute of Education Content Modernization, Kyiv, Ukraine
Nataliia P. Volkova, Alfred Nobel University, Dnipro, Ukraine
Snizhana O. Zelinska, Kryvyi Rih State Pedagogical University, Kryvyi Rih, Ukraine

Перший студентський семінар із комп'ютерних наук та інженерії програмного забезпечення

Арнольд Юхимович Ків¹, Сергій Олексійович Семеріков²[0000-0003-0789-0272],
Володимир Миколайович Соловійов²[0000-0002-4945-202X],
Андрій Миколайович Стрюк³[0000-0001-9240-1976]

¹ Негевський університет імені Давида Бен-Гуріона, Беер-Шева, Ізраїль
kiv@bgu.ac.il

² Криворізький державний педагогічний університет,
пр. Гагаріна, 54, м. Кривий Ріг, 50086, Україна
{semerikov, vnsoloviev2016}@gmail.com

³ Криворізький національний університет,
вул. Віталія Матусевича, 11, м. Кривий Ріг, 50027, Україна
andrey.n.stryuk@gmail.com

Анотація. Стаття є вступною до збірника вибраних праць Першого студентського семінару із комп'ютерних наук та інженерії програмного забезпечення (First Student Workshop on Computer Science & Software Engineering – CS&SE@SW 2018), який відбувся у м. Кривий Ріг (Україна) 30 листопада 2018 р. Стаття містить короткий опис відібраних праць, деякі спостереження про семінар і висновки про його майбутнє.

Ключові слова: комп'ютерні науки, програмна інженерія, студентський семінар.

First Student Workshop on Computer Science & Software Engineering

Arnold E. Kiv¹, Serhiy O. Semerikov²[0000-0003-0789-0272],
Vladimir N. Soloviev²[0000-0002-4945-202X] and Andrii M. Striuk³[0000-0001-9240-1976]

¹ Ben-Gurion University of the Negev, Beer Sheba, Israel
kiv@bgu.ac.il

² Kryvyi Rih State Pedagogical University, 54, Gagarin Ave., Kryvyi Rih, 50086, Ukraine
{semerikov, vnsoloviev2016}@gmail.com

³ Kryvyi Rih National University, 11, Vitalii Matusevych St., Kryvyi Rih, 50027, Ukraine
andrey.n.stryuk@gmail.com

Abstract. This is an introductory text to a collection of selected papers from the First Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), which was held in Kryvyi Rih, Ukraine, on the November 30, 2018. It consists of short summaries of selected papers and some observations about the event and its future.

Keywords: computer science, software engineering, student workshop.

1 Тематика та формат семінару

У 2018 році науковими напрямками семінару CS&SE@SW 2018 були:

1. Комп'ютерні науки
 - a. Теоретична інформатика
 - (1) Структури даних та алгоритми
 - (2) Теорія обчислень
 - (3) Теорія інформації та кодування
 - (4) Теорія мов програмування
 - (5) Формальні методи
 - b. Комп'ютерні системи
 - (1) Комп'ютерна архітектура та комп'ютерна інженерія
 - (2) Аналіз продуктивності комп'ютера
 - (3) Паралельні та розподілені системи
 - (4) Комп'ютерні мережі
 - (5) Формальні методи
 - (6) Бази даних
 - c. Комп'ютерні застосування
 - (1) Комп'ютерна графіка та візуалізація
 - (2) Людина-машинна взаємодія
 - (3) Наукові обчислення
 - (4) Штучний інтелект
2. Інженерія програмного забезпечення:
 - a. Вимоги до програмного забезпечення
 - b. Проектування програмного забезпечення
 - c. Конструювання програмного забезпечення
 - d. Тестування програмного забезпечення
 - e. Обслуговування програмного забезпечення
 - f. Управління налаштуванням програмного забезпечення
 - g. Управління інженерією програмного забезпечення
 - h. Процес розробки програмного забезпечення
 - i. Моделі та методи інженерії програмного забезпечення
 - j. Якість програмного забезпечення
 - k. Професійна практика інженерії програмного забезпечення
 - l. Економіка інженерії програмного забезпечення
 - m. Обчислювальні основи інженерії програмного забезпечення

- p. Математичні основи інженерії програмного забезпечення
- o. Інженерні основи інженерії програмного забезпечення

До участі в семінарі було подано 25 доповідей, 19 із яких були прийняті програмним комітетом: 7 за напрямом «Комп'ютерні науки» та 12 – «Інженерія програмного забезпечення». Короткий огляд прийнятих доповідей буде подано у наступних секціях.

Для підвищення інтерактивності семінару всі доповіді транслювались через Facebook з можливістю онлайн-запитань та реплік віддалених учасників. Регламент семінару передбачав презентацію доповіді одним із авторів відібраної статті (15 хвилин) та дискусійну сесію (17 хвилин).

2 Пленарна доповідь

Ураховуючи значний інтерес основних учасників семінару – студентів магістратури та аспірантури – до питань співвідношення комп'ютерних наук та інженерії програмного забезпечення, пленарна доповідь «Інженерія програмного забезпечення: перші 50 років становлення та розвитку» [1] представлена у збірнику в повному обсязі. Ключовий доповідач – завідувач кафедри моделювання та програмного забезпечення Криворізького національного університету Андрій Миколайович Стрюк – зосередився на виникненні інженерії програмного забезпечення. На основі аналізу матеріалів перших конференцій з інженерії програмного забезпечення, проведених під егідою НАТО у 1968-1969 рр. доповідач показав, як криза програмного забезпечення спонукала науковців та практиків об'єднати зусилля для формування інженерного підходу до програмування. У статті окреслено відмінності професійної підготовки фахівців з інженерії програмного забезпечення, виокремлено фундаментальні складові підготовки майбутніх інженерів-програмістів, розглянуто еволюцію підходів до проектування, впровадження, тестування та документування програмного забезпечення. Аналіз історичних етапів розвитку інженерії програмного забезпечення показав, що незважаючи на загальне визнання важливості застосування при розробці програмного забезпечення апарату комп'ютерних наук, воно створювалась емпіричним способом без його використання. Фактором, що змусив програмістів-практиків звернутися до математичних основ інженерії програмного забезпечення, є зростання складності програмного забезпечення та нездатність емпіричних підходів до його розробки та управління впоратися з нею. У професійній підготовці інженерів-програмістів виділено проблему швидкого застарівання технологічного змісту навчання, розв'язання якої полягає у його фундаменталізації через виокремлення базових основ галузі. Визначено, що опанування основ комп'ютерних наук (інформатики) є фундаментом професійної підготовки з інженерії програмного забезпечення.

3 Комп'ютерні науки

У доповіді А. О. Белінського та В. М. Соловйова «Складні мережі та індикатори кризових і критичних явищ на криптовалютному ринку» [2] продемонстровано можливість побудови індикаторів-передвісників кризових і критичних явищ на швидкозмінному криптовалютному ринку із застосуванням методів теорії складних мереж. Авторами показано можливості побудови динамічних мір мережної складності та способи їх застосування для виявлення передкризових періодів. На прикладі часових рядів біткойну як ключової криптовалюти показано застосування побудованих індикаторів для актуального прогнозу її курсу.

Доповідь Б. В. Гребенюка «Модифікація аналітичного гамма-алгоритму пласкої укладки графа» [3] присвячена одному із методів візуалізації складних мереж. Автор вказує, що у вигляді графа можна зобразити будь-які структури, що мають зв'язки між елементами. Але часто подібні структури збільшуються до таких розмірів, що важко визначити, чи можливо представити їх на площині без перетину зв'язків. Існує багато алгоритмів, що вирішують це питання. Одним із таких є гамма-алгоритм. У статті визначені його проблеми та запропоновані методи їх вирішення, а також досліджені шляхи їх досягнення.

У доповіді М. Г. Долотій та П. В. Мерзликіна «Використання генератора випадкових чисел з апаратним джерелом ентропії для задач симетричної криптографії» [4] перевірено можливості використання розробленого авторами генератора випадкових чисел, який використовує в якості джерела ентропії шуми звукової карти, в алгоритмах симетричної криптографії (за набором NIST Statistical Test Suite). Показано, що послідовності, що генеруються, мають рівномірний спектр та не мають прихованих періодичностей. Зроблено висновок про те, що розроблена бібліотека генерації випадкових чисел може бути використана в проектах, які мають потребу в високоякісних послідовностях випадкових чисел.

У доповіді А. А. Зеленського «Актуальність дослідження програм семантичного аналізу текстів та огляд методів їх реалізації» [5] виконано огляд методів обробки природномовних текстових відповідей з метою екстракції та репрезентації семантики, які ґрунтуються на ефективному поєднанні лінгвістичних технологій аналізу та методів аналізу. Виокремлено методи з використанням векторної моделі та латентно-семантичного аналізу як найбільш ефективні у таких напрямках обробки природної мови, як моделювання концептуальних знань людини та інформаційний пошук. Показано, що при реалізації останнього латентно-семантичний аналіз показує набагато кращі результати порівняно зі звичайними векторними методами.

У доповіді Д. І. Куропятника «Актуальність проблеми параметричної ідентифікації математичної моделі» [6] виконано огляд можливостей підвищення ефективності математичної моделі за рахунок ідентифікації параметрів об'єкта. Встановлено, що ключовим фактором для параметризації є врахування властивостей значень моделі в конкретний момент часу, що дозволяє глибше проаналізувати залежності даних та кореляцію між ними. Наведено різні

функціонали якості (середньоквадратичний критерій, мінімаксна функція, функція-нев'язка), застосування яких під час параметризації надає можливість різноманітно аналізувати модель, тестуючи її на різних алгоритмах, об'ємах даних та умовах гарантованої збіжності методів функціоналу.

У доповіді Л. В. Легкої та С. В. Шокалюк «Квантове програмування – перспективний напрямок розвитку ІТ» [7] обґрунтовано перспективи використання квантових комп'ютерів. Зокрема, наведені тлумачення базових понять квантової механіки, відмічені основні недоліки та певні переваги використання квантових комп'ютерів для надшвидкого розв'язання суспільно значущих задач, наведено узагальнені відомості про успішні спроби у розробках квантових комп'ютерів, зокрема, компаніями IBM, Intel, Google, Microsoft, та відкритості хмарного доступу до них. Виконано короткий огляд засобів квантового програмування Q# і QISKit, вказано перспективи їх використання у початковому циклі навчання комп'ютерних наук.

У доповіді Л. О. Фадєєвої та П. В. Мерзликіна «Дослідження криптографічного алгоритму на основі тюрмітів на відповідність лавинному критерію» [8] досліджується запропонований криптографічний алгоритм на відповідність лавинному критерію. Представлено модифікацію алгоритму, що дозволяє задовольнити цей критерій без втрати інших властивостей алгоритму. В майбутньому планується розширити алгоритм для тюрмітів з різними правилами руху та створити модифікацію алгоритму для роботи в реальному часі, наприклад, для шифрування графіку.

У дискусійній частині секції комп'ютерних наук А. В. Карявка презентував персональний граф знань Neuronet.

4 Інженерія програмного забезпечення

У доповіді В. К. Гуменка та І. А. Котова «Інтелектуальний веб-інтерфейс доступу до даних особливої важливості» [9] виконано аналіз проблем забезпечення безпеки конфіденційних даних, розглянуті існуючі способи захисту інформації, визначено актуальність та завдання дослідження, виокремлено принципи побудови систем захисту інформації, розроблено структурні та функціональні моделі, спроектовано інтелектуальний блок доступу до даних особливої важливості. Досліджено рівень надійності та стійкості питань системи до набору даних, зроблені висновки, що розроблений веб-інтерфейс дозволяє підвищити рівень захищеності інформації.

У доповіді Д. О. Ганістрата та І. А. Котова «Автоматизована система оперативного розподілу ресурсів ліквідації комплексних надзвичайних ситуацій» [10] проведено критичний аналіз сучасного стану проблеми ліквідації надзвичайних станів в промисловості та природному оточенні, принципів побудови автоматизованих систем та розподілених систем керування базами даних, розроблені математичні і структурні моделі, спроектована автоматизована система управління ресурсами ліквідації надзвичайних ситуацій. Досліджено забезпечення безпеки інформаційних потоків даних програмного комплексу,

технічна ефективність прийнятих рішень. Зроблені висновки, що розроблений комплекс автоматизації дозволяє реалізовувати ефективне управління ресурсами ліквідації надзвичайних ситуацій.

У доповіді В. М. Пірогова, А. М. Горло та І. С. Мінтій «Програмна розробка алгоритму адаптації дизайну сайту для людей з порушенням кольоросприйняття» [11] розглянуто реалізацію наступних алгоритмів: конвертації із колірної моделі rgb до lms моделі (для моделювання різних видів порушень кольоросприйняття існує спеціальний алгоритм саме в колірній моделі lms), моделювання різних видів дальтонізму в lms моделі (перетворення нормальних значень в колірній моделі lms до значень з різними видами порушень кольоросприйняття), конвертації даних із колірної моделі lms до rgb моделі, конвертації з колірної моделі rgb до hsl моделі, фільтрації кольорів у hsl моделі і конвертації з hsl моделі до колірної моделі rgb.

У доповіді Є. Ю. Тищенко та А. М. Стрюка «Актуальність розробки моделі адаптивного навчання» [12] показано, що завдяки адаптивному тестуванню підвищується точність, якість, достовірність навчання та зацікавленість студента, що у цілому сприяє підвищенню рівня його мотивації. У процесі проходження тесту будується модель особи, яка навчається, для подальшого використання при виборі наступних завдань тестування залежно від рівня знань студента та його індивідуальних особливостей. При обчисленні оцінки адаптивна система тестування враховує ймовірність, що студент може вгадати відповідь, кількість спроб пройти тест і середній результат, досягнений за час всіх спроб.

У доповіді Д. О. Курганова та А. А. Азаряна «Розробка програмного забезпечення для мінімізації витрат часу та підвищення продуктивності праці у сфері комунікаційних послуг» [13] визначено, що сучасні засоби автоматизації підвищують ефективність та коректність надання консультативних послуг. Мінімізація витрат часу та підвищення продуктивності праці в сфері комунікаційних послуг можливі за наступних умов: підвищення швидкості опрацювання питань клієнтів за рахунок вибору оптимальної методики пошуку відповіді; зменшення кількості помилок за рахунок використання шаблонів відповідей. У роботі запропоновані заходи, що надають можливість знизити трафік у системі самообслуговування абонентів за допомогою інтерактивних голосових меню.

У доповіді І. В. Олексієнка та В. М. Франчука «Web-орієнтована система «Електронний розклад» [14] представлено впроваджену у навчальний процес веб-орієнтовану систему для введення та перегляду розкладу занять навчального підрозділу закладу вищої освіти.

У доповіді О. О. Соменко «Створення української локалізації системи комп'ютерної математики Sage» [15] виявлено основні проблеми, які виникають у процесі адаптації програмного забезпечення до використання україномовними користувачами, надано характеристику системи Sage та визначено особливості процесу розробки і внесення змін у програму, розглянуто принципи роботи програми Poedit як інструменту для створення української локалізації Sage Notebook.

У доповіді О. М. Гараніна та Н. В. Моїсеєнко «Адаптивний штучний інтелект у RPG-грі на ігровому рушії Unity» [16] на прикладі розробленої гри представлена система адаптації ігрових ситуацій у залежності від дій гравця та прив'язка цього механізму до ігрового процесу.

У доповіді О. О. Кацка та Н. В. Моїсеєнко «Розробка ігор на ігровому рушії Unity для дослідження елементів когнітивного мислення в ігровому процесі» [17] проаналізовані актуальні методи оцінювання когнітивних функцій людини, розроблено алгоритми та програмна реалізація оцінювання когнітивних здібностей за допомогою гри.

У доповіді В. С. Кузнєцова та Н. В. Моїсеєнко «Розробка графічної оболонки для програми розрахунків фізичних характеристик твердих тіл» [18] проаналізоване існуюче програмне забезпечення для розрахунків фізичних характеристик твердих тіл, розроблений алгоритм та програмна реалізація графічної оболонки для програми розрахунків фізичних характеристик твердих тіл.

У доповіді М. Є. Петрової, М. М. Мінтія, С. О. Семерікова та Н. П. Волкової «Розробка адаптивного навчального програмного забезпечення з теми «Дробові числа» для учнів 5 класу» [19] проаналізовано стан проблеми розробки та використання адаптивного програмного забезпечення для навчання математики, обґрунтовано вибір засобів розробки адаптивного навчального програмного забезпечення для учнів середньої школи, розроблено та експериментально перевірено адаптивне навчальне програмне забезпечення з теми «Дробові числа» для учнів 5 класу закладів загальної середньої освіти. Результати експериментального навчання з використанням розробленого програмного забезпечення показали, що виконана розробка не ураховує зміну індивідуальних особливостей учня (у тому числі психологічних) у процесі навчання. Це визначає наступні шляхи його розвитку, спрямованого на підвищення рівня адаптивності: індивідуалізація навчальних впливів; формалізація та реалізація моделі учня; урахування процесуальних характеристик навчання у моделі учня.

У доповіді О. В. Сироватського, С. О. Семерікова, Є. О. Модло, Ю. В. Єчкало та С. О. Зелінської «Проектування програмних засобів доповненої реальності навчального призначення» [20] виконано історико-технологічний аналіз досвіду застосування засобів доповненої реальності для розробки інтерактивних навчальних матеріалів, схарактеризовано програмне забезпечення для проектування засобів доповненої реальності навчального призначення та визначено технологічні вимоги для факультативу «Розробка програмних засобів віртуальної та доповненої реальності», розроблено окремі складові навчально-методичного комплексу із проектування систем віртуальної та доповненої реальності для майбутніх учителів інформатики.

У дискусійній частині секції інженерії програмного забезпечення Б. К. Мігілев презентував систему КНУ-Розклад.

5 Висновки

Огляд доповідей учасників семінару надає можливість зробити висновок про те, що були охоплені далеко не всі заплановані наукові напрями. Головну увагу було приділено питанням теоретичної інформатики, комп'ютерної графіки та візуалізації, людино-машинної взаємодії, наукових обчислень, штучного інтелекту, проектування програмного забезпечення, конструювання програмного забезпечення, тестування програмного забезпечення, процесу розробки програмного забезпечення.

Наступного року організатори семінару очікують розширення географії учасників семінару та доповідей також із комп'ютерних систем і різних аспектів інженерії програмного забезпечення.

References

1. Striuk, A.M.: Software engineering: first 50 years of formation and development. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 11–36, online (2018, in press)
2. Bielinskyi, A.O., Soloviev, V.N.: Complex network precursors of crashes and critical events in the cryptocurrency market. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 37–45, online (2018, in press)
3. Hrebeniuk, B.V.: Modification of the analytical gamma-algorithm for the flat layout of the graph. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 46–54, online (2018, in press)
4. Dolotii, M.H., Merzlykin, P.V.: Using the random number generator with a hardware entropy source for symmetric cryptography problems. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 55–62, online (2018, in press)
5. Zelenskyi, A.A.: Relevance of research of programs for semantic analysis of texts and review of methods of their realization. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 63–69, online (2018, in press)
6. Kuropiatnyk, D.I.: Actuality of the problem of parametric identification of a mathematical model. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 70–75, online (2018, in press)
7. Lehka, L.V., Shokaliuk, S.V.: Quantum programming is a promising direction of IT development. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.)

- Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 76–82, online (2018, in press)
8. Fadiieva, L.O., Merzlykin, P.V.: The avalanche criterion satisfaction research of the turmite-based cryptographic algorithm. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 83–86, online (2018, in press)
 9. Humenok, V.K., Kotov, I.A.: Intelligent web-interface for access to data of particular importance. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 87–93, online (2018, in press)
 10. Hanistrat, D.O., Kotov, I.A.: Automated system of operational resource allocation for complex emergency situations. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 94–102, online (2018, in press)
 11. Pirohov, V.M., Horlo, A.M., Mintii, I.S.: Software development of the algorithm of adaptating of the website design for people with color-blindness. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 103–108, online (2018, in press)
 12. Tyshchenko, Ye.Yu., Striuk, A.M.: The relevance of developing a model of adaptive learning. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 109–115, online (2018, in press)
 13. Kurhanov, D.O., Azaryan, A.A.: Software development to minimize time costs and increase productivity in the area of communication services. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 116–127, online (2018, in press)
 14. Oleksiienko, I.V., Franchuk, V.M.: Web-oriented electronic schedule. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 128–131, online (2018, in press)
 15. Somenko, O.O.: Creation of Ukrainian localization of computer mathematics system Sage. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 132–142, online (2018, in press)
 16. Haranin, O.M., Moiseienko, N.V.: Adaptive artificial intelligence in RPG-game on the Unity game engine. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 143–150, online (2018, in press)
 17. Katsko, O.O., Moiseienko, N.V.: Development computer games on the Unity game engine for research of elements of the cognitive thinking in the playing process. In: Kiv, A.E.,

- Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 151–155, online (2018, in press)
18. Kuznietsov, V.S., Moiseienko, N.V.: Development graphic shell for the program calculations of physical properties of solids. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 156–161, online (2018, in press)
 19. Petrova, M.Ye., Mintii, M.M., Semerikov, S.O., Volkova, N.P.: Development of adaptive educational software on the topic of “Fractional Numbers” for students in grade 5. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 162–192, online (2018, in press)
 20. Syrovatskyi, O.V., Semerikov, S.O., Modlo, Ye.O., Yechkalo, Yu.V., Zelinska, S.O.: Augmented reality software design for educational purposes. In: Kiv, A.E., Semerikov, S.O., Soloviev, V.N., Striuk, A.M. (eds.) Proceedings of the 1st Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018, CEUR-WS.org, pp. 193–225, online (2018, in press)

Інженерія програмного забезпечення: перші 50 років становлення та розвитку

Андрій Миколайович Стрюк^[0000-0001-9240-1976]

Криворізький національний університет,
вул. Віталія Матусевича, 11, м. Кривий Ріг, 50027, Україна
andrey.n.stryuk@gmail.com

Анотація. У статті проаналізовано основні етапи розвитку інженерії програмного забезпечення (ІПЗ). На основі аналізу матеріалів перших конференцій з ІПЗ (1968-1969 рр.) визначено, як криза програмного забезпечення спонукала науковців та практиків об'єднати зусилля для формування інженерного підходу до програмування. Окреслено відмінності професійної підготовки фахівців з ІПЗ. Виокремлено фундаментальні складові підготовки майбутніх інженерів-програмістів. Розглянуто еволюцію підходів до проектування, впровадження, тестування та документування програмного забезпечення. Виокремлено системні наукові, технологічні підходи і методи до проектування та конструювання комп'ютерних програм. Аналіз історичних етапів розвитку ІПЗ показав, що незважаючи на загальне визнання важливості застосування при розробці програмного забезпечення математичного апарату логіки, теорії автоматів та лінгвістики, воно створювалась емпіричним способом без його використання. Фактором, що змусив програмістів-практиків звернутися до математичних основ ІПЗ, є зростання складності програмного забезпечення та нездатність емпіричних підходів до його розробки та управління впоратися з нею. У професійній підготовці інженерів-програмістів виділено проблему швидкого застарівання технологічного змісту навчання, розв'язання якої полягає у його фундаменталізації через виокремлення базових основ галузі. Визначено, що опанування основ комп'ютерних наук (інформатики) є фундаментом професійної підготовки з ІПЗ.

Ключові слова: інженерія програмного забезпечення, професійна підготовка, програмне забезпечення, програмна система, програмування, проектування, моделювання.

Software engineering: first 50 years of formation and development

Andrii M. Striuk^[0000-0001-9240-1976]

Kryvyi Rih National University, 11, Vitalii Matusevych St., Kryvyi Rih, 50027, Ukraine
andrey.n.stryuk@gmail.com

Abstract. The article analyzes the main stages of software engineering (SE) development. Based on the analysis of materials from the first SE conferences (1968-1969), it was determined how the software crisis prompted scientists and practitioners to join forces to form an engineering approach to programming. Differences in professional training for SE are identified. The fundamental components of the training of future software engineers are highlighted. The evolution of approaches to the design, implementation, testing and documentation of software is considered. The system scientific, technological approaches and methods for the design and construction of computer programs are highlighted. Analysis of the historical stages of the development of SE showed that despite the universal recognition of the importance of using the mathematical apparatus of logic, automata theory and linguistics when developing software, it was created empirically without its use. The factor that led practitioners to turn to the mathematical foundations of an SE is the increasing complexity of software and the inability of empirical approaches to its development and management to cope with it. The training of software engineers highlighted the problem of the rapid obsolescence of the technological content of education, the solution of which lies in its fundamentalization through the identification of the basic foundations of the industry. It is determined that mastering the basics of computer science is the foundation of vocational training in SE.

Keywords: software engineering, professional training, software, software system, programming, design, simulation.

1 Вступ

Починаючи з 2012 року, серед пріоритетних напрямів освіти і науки щодо навчання студентів та аспірантів, стажування наукових і науково-педагогічних працівників у провідних закладах вищої освіти та наукових установах за кордоном [31], що відносяться до інформатики та обчислювальної техніки, три – програмна інженерія, програмне забезпечення систем та інженерія програмного забезпечення – відносяться до однієї спеціальності: 121 – Інженерія програмного забезпечення. Крім того, значна частина інших пріоритетних напрямів (математичне та комп’ютерне моделювання; інформаційно-комунікаційні

технології; системи штучного інтелекту; системне програмування та ін.) є дотичними до неї. Технології та засоби розробки програмних продуктів і систем визначено як один із пріоритетних напрямів наукових досліджень і науково-технічних розробок в Україні на період до 2020 року [32]. Ці та низка інших законодавчих ініціатив нашої держави є свідченням нагальної суспільної потреби у компетентних фахівцях з інженерії програмного забезпечення, підготовлених на основі кращих світових стандартів та передового зарубіжного досвіду і здатних до проектування, апробації, упровадження та комерціалізації інноваційних технологій ПЗ. Аналіз світового досвіду з підготовки фахівців з ПЗ доречно почати з ретроспективного огляду еволюції самого поняття «інженерія програмного забезпечення» та основних етапів розвитку цієї галузі.

Метою статті є аналіз основних етапів розвитку ПЗ як галузі знань, виокремлення фундаментальних складових підготовки майбутніх інженерів-програмістів та визначення тенденцій розвитку цієї галузі на найближче десятиліття.

2 Виникнення інженерії програмного забезпечення

Перше вживання терміну «програмна інженерія» (software engineering) датується серпнем 1966 року [16], але роком появи інженерії програмного забезпечення (ПЗ) як галузі вважається 1968 рік, у якому в Німеччині відбулась перша конференція, що мала назву «Software engineering». Головною тематикою конференції було проектування, виробництво та обслуговування програмного забезпечення. Один із головних учасників конференції П. Наур (Peter Naur) зазначав, що робота проектувальників програмного забезпечення схожа на роботу архітекторів та інженерів-будівельників, особливо тих, хто займається проектуванням великих гетерогенних конструкцій, таких як міста та промислові підприємства [24, с. 13]. Термін «програмна інженерія» (software engineering) був навмисне обраний як провокативний: «це поняття означало, що виробництво програмного забезпечення має базуватися на тому ж типі теоретичних засад та практичних застосувань, що й у традиційних галузях інженерії» [11, с. I; 24, с. 13].

Походження програмної інженерії учасники конференції пов'язували із кризою програмного забезпечення – терміном, запропонованим головою програмного комітету Ф. Л. Бауером (Friedrich Ludwig "Fritz" Bauer). На його думку, криза полягала у неможливості застосування «кустарних» (напівінтуїтивних) методів розробки для виробництва великих масштабованих програмних систем: «Існуюче програмне забезпечення розробляється аматорами (незалежно від того, де – в університетах, компаніях чи на виробництві) за допомогою майстерності одинаків (в університетах) або великої кількості працівників («мільйон мавп») на виробництві, є ненадійним і потребує постійного «технічного обслуговування» (причому слово «обслуговування» неправильно використовується для позначення збоїв та відмов, які очікуються від виробника із самого початку), є неохайним, непрозорим та невдосконалюваним (або принаймні занадто вартісним, щоб це зробити). І нарешті, існуюче

програмне забезпечення надходить занадто пізно і коштує дорожче, ніж очікувалося, та не виправдовує сподівань, які на нього покладалися» [3]. За результатами роботи конференції у 1971 році Ф. Л. Бауер дав напівжартівливе визначення ПЗ як частини інформатики, що занадто важка для інформатиків, та більш серйозне – як створення та використання раціональних принципів інженерії для отримання економічного, надійного та ефективно працюючого на реальних комп'ютерах програмного забезпечення [3, с. 530].

На думку таких учасників конференції, А. Дж. Перліс (Alan Jay Perlis) та Ф. Л. Бауер, для проектування програмного забезпечення математична підготовка не є необхідною, але її наявність додає програмному проекту елегантності, адже програмні системи є математичними за природою та повинні бути побудовані за рівнями та модулями, що утворюють математичну структуру [24, с. 37]. Основними критеріями проектування були обрані загальні критерії (спільні для різних систем), користувацькі вимоги, надійність та логічна повнота. Серед технологій проектування обговорювались послідовність кроків проектування, структура програмного проекту, забезпечення зворотного зв'язку через моніторинг та моделювання, застосування високорівневих мов програмування тощо.

3 Професійна підготовка фахівців з інженерії програмного забезпечення

Стосовно професійної підготовки фахівців з ПЗ А. Дж. Перліс та Е. Е. Девід-молодший (Edward Emil David Jr.) поставили ряд проблемних запитань [24, с. 125-126]:

1. Чи можливо працювати інженером-програмістом без формальної освіти з відповідної спеціальності?
2. Чи співпадає ПЗ із комп'ютерними науками?
3. Як краще підготувати фахівця з ПЗ: за бакалаврською програмою в університеті, на курсах підвищення кваліфікації або за дворічною програмою підготовки після стандартної шкільної освіти?
4. Чи матимуть фахівці, підготовлені за цими програмами, ріст та перспективи у нашому суспільстві?
5. Чи будуть вони достатньо корисними для фірми, уряду чи університету, і чи є їх значення таким, що вони можуть реалізовувати свої таланти в інших видах діяльності, або вони назавжди приречені залишатися програмістами?
6. Які програми підготовки необхідні для фахівців з ПЗ незалежно від рівня освіти?
7. Чи ПЗ дійсно відрізняється від того, що ми зараз називаємо системною інженерією?
8. Що спільного мають ПЗ та комп'ютерна інженерія із традиційною інженерною освітою у Сполучених Штатах Америки або Західній Європі?

Д. Т. Росс (Douglas Taylor Ross), відповідаючи на поставлені запитання, наголошував на необхідності окремої формальної освіти з ПЗ на рівні бакалавра [24, с. 127]. Інші учасники дискусії підіймали питання практичної підготовки як фахівців, так і викладачів на відповідній спеціальності.

Конференція 1969 року [25], що відбулась в Італії, була присвячена технологіям ПЗ. На відміну від попередньої, професійна підготовка фахівців з ПЗ стала предметом обговорення на окремій секції конференції.

А. Дж. Перліс, продовжуючи розпочату на попередній конференції дискусію, акцентував увагу на трьох питаннях:

1. Чи існує реальна відмінність між ПЗ та комп'ютерними науками?
2. Якщо вона існує, то чи потрібне вивчення ПЗ як окремої дисципліни?
3. За якою формою повинні викладатися університетські курси з ПЗ?

Обговорення цих та інших питань, включно із тим, чи достатньо усталеними є комп'ютерні науки, щоб їх можна було навчати студентів, і чи мають вони певні явні базові принципи.

На перше питання А. Дж. Перліс відповідав ствердно: «Я думаю, що всі ті з нас, хто працює в університетах, добре розуміють сутність Ph. D. програми з комп'ютерних наук: здорова доза логіки, теорії автоматів та обчислень, трохи менша – чисельного аналізу, один-два курси з поглибленого програмування того чи іншого виду, трохи штучного інтелекту, і дешифру ще чогось» [25, с. 61-62]. Ф. Л. Бауер зауважив, що у Німеччині робоча група, що розробляє програми підготовки, назвала відповідний предмет інформатикою (“Informatik”): «Ми очікуємо, що наші студенти самі зроблять вибір, будуть вони фахівцями з комп'ютерних наук чи з інженерії програмного забезпечення» [25, с. 62]. На думку А. Дж. Перліса, для ПЗ фундаментальними є курси дослідження операцій та управління проектами – настільки ж фундаментальними, як й курс теорії автоматів, та більш фундаментальними, ніж будь-який математичний курс [25, с. 62].

Певним продовженням цієї дискусії є інша публікація А. Дж. Перліса [17], у якій він підкреслює суспільну значущість професії інженера з програмного забезпечення та нагальну необхідність розробки відповідних освітніх програм їх підготовки. Спеціалізацією таких інженерів є програмне забезпечення – а саме його проектування, виробництво та обслуговування. До заданих на обговорюваних конференціях проблемних запитань А. Дж. Перліс додає ще декілька:

1. Якщо підготовка фахівців з ПЗ відбуватиметься в університеті, то на якій кафедрі або факультеті?
2. Чи програма підготовки повинна бути окремою, або вона може бути варіативною частиною іншої програми?
3. Чи будуть за такою програмою фахівці підготовлені для вирішення системних проблем, що виникнуть у майбутньому?
4. Чому ми говоримо про інженерію, а не про науку?

Автор пропонує розпочинати з магістерської програми, далі поширюючи її на бакалаврат та докторат. «Метою є зосередження на відомих інструментах та їх ефективному використанні, а не на періодах інтенсивних інновацій та відкриттів. ... На мій погляд, професійна підготовка з інженерії програмного забезпечення є сплавом математики, теорії управління, комп'ютерних наук та практичного досвіду, отриманого при роботі з актуальними програмними системами та пов'язаними проблемами» [17, с. 541].

Дж. Фельдман (Jerome A. Feldman) із власного досвіду наводив приклади того, що фахівці на виробництві не читають літературу, навіть якщо в ній наявні рішення їх виробничих проблем: «Ті, хто опанували освітню програму в галузі комп'ютерних наук, в цьому відношенні є більш ефективними. Ми намагаємось подолати цю проблему в Стенфорді через започаткування програми підготовки з прикладних комп'ютерних наук, спрямованої на підготовку ефективних фахівців для промисловості» [25, с. 62]. Р. М. Макклор (Robert M. McClure) зазначив, що університети мають значну проблему, в основі якої лежить розмежування між ПЗ та комп'ютерними науками.

Б. Галлер (Bernard A. Galler) навів приклад курсу ПЗ: «Ми беремо команду з двох або трьох викладачів і 20-25 студентів та даємо їм проект. Перший – графічна математична система, другий – система програмування мовою BASIC. Ми даємо їм повну роботу: проектування, вказавши технічні характеристики та розділивши їх на групи по дві-три особи, реалізація, опис власної роботи для інших груп, розробка інтерфейсів, документування та ін. У кожному випадку продукт був практично корисним; за один семестр ви не зможете завершити щось такого масштабу. Цей вид досвіду виходить за межі написання проекту малого класу типу компілятора та надає певний досвід програмної інженерії» [25, с. 62].

Д. Т. Росс вказав на обмеженість та несистемність такого підходу, а Е. Д. Фалкофф (Adin D. Falkoff) – про те, що курс ПЗ повинен включати також роботу з апаратним забезпеченням. «Крім того, я завжди вважав, що інженерія має справу з питаннями економіки; дисципліна, пов'язана з проблемами комплексного використання ресурсів, повинна містити елементи дослідження операцій та відповідні курси» [25, с. 62-63].

Е. В. Дейкстра (Edsger Wybe Dijkstra) підіймає проблему швидкого застарівання вузькопрофільних знань: «Я вважаю неправильним навчати матеріал, який, як мені відомо, застаріє через кілька років. ... Ви повинні навчити розуміння методу, почуття якості та стилю» [25, с. 65].

Конференції з ПЗ, проведені під егідою наукового комітету НАТО у 1968 та 1969 рр., у цілому визначили сферу ПЗ та шляхи підготовки відповідних фахівців у закладах освіти:

1. Методи та засоби ПЗ застосовуються до великих складних програмних систем, які не можуть бути створені однією особою або невеликим колективом розробників.
2. Попри свою назву, ПЗ повинна включати питання взаємодії програмної та апаратної складових комп'ютерної системи.

3. Метою ІПЗ є розробка програмних систем з наперед визначеними рівнями якості, надійності та ефективності в умовах обмеження ресурсів (часових, людських, матеріальних, програмно-апаратних тощо). У зв'язку із цим, на відміну від комп'ютерних наук (інформатики), питання дослідження операцій та управління проектами для ІПЗ забезпечення є фундаментальними.
4. Підготовка фахівців з ІПЗ у ЗВО є доцільною на рівні бакалаврату. В процесі підготовки доцільно поєднувати теоретичну та практичну підготовку (на виробництві або із застосуванням запозичених із виробництва методів розробки програмного забезпечення).
5. Суттєвим для підготовки фахівців з ІПЗ є вивчення артефактів, що створюються у процесі програмної інженерії: документації, програмного коду, меморандумів, групових обговорень та ін.
6. У навчанні ІПЗ («технології програмування») із самого початку гостро постала проблема швидкого застарівання технологічного змісту навчання, розв'язання якої полягає у його фундаменталізації через виокремлення базових основ галузі.

Останнє викликало найбільшу дискусію, за результатами якої було визначено, що опанування основ комп'ютерних наук («інформатики» за Ф. Л. Бауером та «математичної інженерії» за Е. В. Дейкстрою) є фундаментом професійної підготовки з ІПЗ.

4 Тенденції розвитку професійної підготовки фахівців з інженерії програмного забезпечення

У 2018 році в статті, присвяченій 50-тиріччю підготовки фахівців з ІПЗ, Н. Р. Мід (Nancy R. Mead), Д. Гарлан (David Garlan) та М. Шоу представили сучасне трактування ІПЗ як «складової комп'ютерних наук, яка створює практичні, економічно вигідні рішення для обчислювальних задач та задач опрацювання інформації, переважно шляхом застосування наукових знань та розробки програмних систем, що служать людству» [13, с. 1]. На думку авторів, сучасна ІПЗ базується на трьох групах ключових принципів:

1. *Основні концепції комп'ютерних наук*, пов'язані зі структурами даних, алгоритмами, мовами програмування та їх семантикою, аналізом, обчислювальністю, моделями обчислень тощо:
 - абстракція надає можливість контролювати складність;
 - структурування задач часто робить їх більш доступними; існує ряд загальних структур;
 - символічні репрезентації є необхідними та достатніми для вирішення проблем, пов'язаних з інформацією;
 - точні моделі підтримують аналіз та прогнозування;
 - загальні структури задач призводять до канонічних розв'язків.

2. *Основи інженерії*, пов'язані з архітектурою, процесами інженерії, компромісами та витратами, стандартизацією, якістю та гарантіями та інші складові, що забезпечують підхід до проектування та вирішення проблем, який враховує прагматичні аспекти:

- джерелом інженерної якості є інженерні рішення;
- якість програмного продукту залежить від вірності інженера інженерному артефакту;
- інженерія вимагає узгодження суперечливих обмежень;
- інженерні навички покращуються внаслідок ретельної системної рефлексії досвіду.

3. *Соціально-економічні основи*, які включають процес створення та еволюції артефактів, а також питання, пов'язані з політикою, ринками, зручністю використання та соціально-економічними впливами; це забезпечує основу для формування інженерних артефактів, які будуть відповідати їхньому призначенню:

- обмеження на витрати та час значущі, а не просто можливі;
- технологія покращується експоненціально, на відміну від людських здібностей;
- успішна розробка програмного забезпечення залежить від командної роботи творчих людей;
- цілі бізнесу та політики так само накладають обмеження на проектування та розробку програмного забезпечення, як й технічні міркування;
- функціональність програмного забезпечення часто настільки глибоко вбудована в інституційні, соціальні та організаційні механізми, що необхідні методи спостереження, які матимуть коріння в антропології, соціології, психології та інших необхідних дисциплінах;
- замовники та користувачі, як правило, точно не знають, чого хочуть, і відповідальність розробника полягає в тому, щоб полегшити виявлення вимог.

Таким чином, протягом останніх 50 років предмет ПЗ залишався незмінними, проте аспекти, що покривались ПЗ суттєво розширювались. Так, Б. Ренделл у статті 2018 року [19] показує, як змінилось трактування ПЗ – від провокативного терміну та неозначуваної, але іменованої галузі діяльності (за М. Хемілтон) до «інженерної дисципліни, у якій зібрані всі аспекти виробництва програмного забезпечення від ранніх етапів специфікації систем до її обслуговування після уведення в експлуатацію» [19, с. 5], проте «немає універсальних методів та методів розробки програмного забезпечення, які є придатними для всіх систем та всіх компаній – навпаки, за останні 50 років склався різноманітний набір методів та інструментів розробки програмного забезпечення» [19, с. 7].

У 1990 році Ф. Я. Дзержинський і Л. Д. Райков на основі аналізу вітчизняної та зарубіжної літератури, зразків програмних засобів та інших джерел, що втілюють у собі сучасні досягнення ПЗ, пропонують наступні визначення: «інженерія програмного забезпечення – це концепції, методи, інструменти,

системи тощо, які призначені для забезпечення якісного та високопродуктивного характеру інженерної діяльності зі створення та використання програмно-інформаційних засобів обчислювальної техніки» [29, с. 67]. Дослідники, спираючись на практичний досвід, відзначають, що низка помилок під час проектування та реалізації програмного забезпечення допускається фахівцями через занадто вузьке трактування історично сформованих підходів до ПЗ. Таких «вузьких підходів» дослідники виділяють три: «машинобудівний», «адміністративний», «інструментальний».

«Машинобудівний» підхід полягає в занадто буквальній аналогії з поняттям технології машинобудівного виробництва, що передбачає жорстко регламентовану послідовність технологічних операцій, що мають гарантувати отримання продукції з заданою якістю в мінімальній залежності від індивідуальних (наприклад, творчих) можливостей окремих працівників.

«Адміністративний» підхід акцентує на важливості різноманітних організаційно-управлінських заходів з упровадження тих чи інших стандартів, регламентів робіт, типових або базових програм тощо.

«Інструментальний» підхід є найбільш розповсюдженою оманю, на думку дослідників. Його суть полягає в тому, що пошук рішення проблем підвищення продуктивності та якості результатів програмування зводиться до пошуку або розробки тих чи інших інструментальних засобів від особистих до систем «наскрізної» автоматизації робіт тощо.

«Визначимо засіб ПЗ як сукупність деяких результатів в галузі ПЗ, що мають «концептуальний» (методологічний, організаційний тощо), програмний або інший характер і мають наступні відмінності: по-перше, багатократним безпосереднім застосуванням в деякому класі робочих умов; по-друге, достатньо надійною відтворюваністю, при цьому, певного практичного ефекту (реально або імовірно корисного); по-третє, ідентифікованістю – можливістю достатньо об'єктивно і точно виокремити певний засіб від інших при розгляданні його як об'єкту застосування, впровадження, передачі тощо» [29, с. 72].

У 1976 році А. С. Вільямс до професійних засобів ПЗ, спрямованих на розв'язання кризи програмного забезпечення (причому не лише загроз перевищення термінів та бюджетів при розробці, а й нанесення шкоди і створення загрози для життя та здоров'я людини у процесі експлуатації) відносила технології програмування (низхідне, структурне, модульне) та засоби проектування (таблиці прийняття рішень, організація команди програмістів за хірургічним принципом [29, с. 27-34], текстові редактори) [27, с. 5]. Пізніше до таких засобів додалися технології об'єктно-орієнтованого програмування, CASE-системи, універсальні мови програмування, документування та стандартизації: кожен з них при уведенні позиціонувався як «срібна куля» – універсальний засіб, покликаний розв'язати усі проблеми, породжені кризою програмного забезпечення. Ф. Брукс (Frederick Phillips Brooks, Jr.), автор ідеї організації команди програмістів за хірургічним принципом, у статті [8], вказує, що «немає єдиного відкриття ні в технології, ні в методах управління, одне тільки використання якого обіцяло б протягом найближчого десятиліття на порядок підвищити продуктивність, надійність, простоту розробки програмного

забезпечення» [8, с. 10]. Серед інших способів розв'язання тривалої (з 1960-тих по 1990-ті рр.) кризи також пропонувались підвищення дисципліни програмістів та їх професіоналізму, застосування формальних методів, процесів і методологій та ін. Ф. Брукс у якості «срібних куль» розглядає також штучний інтелект, експертні системи, автоматичне програмування, графічне програмування, верифікацію програм, інтегровані середовища розробки.

У виступі на конференції [6] та відповідній статті [5] Б. Боем намагається застосувати діалектичний підхід до еволюції ПЗ: «Філософ Гегель висунув гіпотезу, що поглиблення людського розуміння йде шляхом тези (чому певні речі робляться так, як робляться) – антитези (надає краще пояснення у тих важливих випадках, які теза нездатна пояснити) – синтезу (виявляється, що антитеза відкидає занадто велику частину оригінальної тези, створюється гібрид, який вбирає в себе найкраще з тези та антитези, уникаючи їх недоліків)» [5, с. 12]. На рис. 1 показано застосування діалектичного підходу до розвитку ПЗ з доінженерного періоду (1950-ті рр.) по теперішній час.

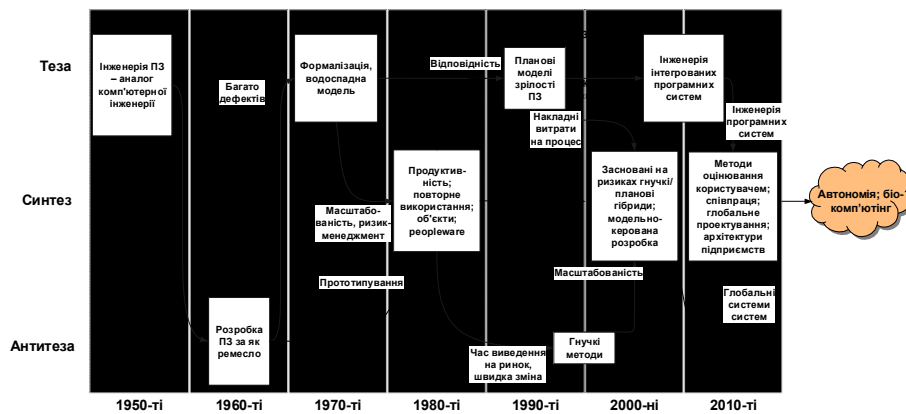


Рис. 1. Діалектичний підхід до еволюції інженерії програмного забезпечення (за Б. Боемом [6])

Для кожного етапу еволюції ПЗ Б. Боем виокремив принципи, що використовуються донині, та застарілі практики (таблиця 1).

Таблиця 1. Діалектика етапів розвитку ПЗ (за Б. Боемом [5])

Період	Категорія діалектики	«Вічні» принципи	Застарілі практики
1950-ті	теза: ПЗ – аналог комп'ютерної інженерії	не нехуйте науками, адже це перша складова інженерії, яка повинна включати в себе не лише математику та комп'ютерні науки, а також поведінкові науки, економіку та менеджмент разом із використанням наукового методу для вивчення досвіду;	унікайте жорсткого послідовного процесу – він, як правило, повільніший, і світ стає надто стійким і непередбачуваним для його використання

Період	Категорія діалектики	«Вічні» принципи	Застарілі практики
		<i>подивіться, перш ніж стрибнути</i> – передчасні зобов'язання можуть бути катастрофічними	
1960-ті	<i>антитеза</i> : розробка програмного забезпечення як ремесло	<i>думайте нестандартно</i> – рутинна інженерія ніколи не створить Інтернет або графічний інтерфейс; створюйте цікаві прототипи з низьким ризиком і потенційно високою винагородою; <i>ураховуйте відмінності у програмному забезпеченні</i> , розробку якого не можна прискорити – зробіть ці відмінності видимими та змістовними для різних зацікавлених сторін	<i>уникайте ковбойського програмування</i>
1970-ті	<i>синтез та антитеза</i> : формалізація та водоспадні процеси	<i>усувайте помилки цюжкраніше</i> , а ще краще – запобігайте їм у майбутньому за допомогою глибокого аналізу першопричин; <i>визначайте цілі системи</i> – без чіткого спільного бачення, ви, ймовірно, отримуєте хаос та розчарування	<i>уникайте низхідного проектування та редукаціонізму</i> – комерційні «коробкові» продукти, повторне використання, IKIWISI (I'll know it when I see it – «дізнаюсь, коли побачу»), швидкі зміни та виникаючі вимоги роблять це все нереалістичнішим для більшості застосувань
1980-ті	<i>синтез</i> : продуктивність та масштабованість	<i>існує багато шляхів підвищення продуктивності</i> , включаючи кадрове забезпечення, навчання, засоби, повторне використання, вдосконалення процесу, прототипування та інші; <i>те, що добре для продуктів, добре для процесу</i> , включаючи архітектуру, повторне використання, компонованість та адаптивність	<i>скептично ставтесь до «срібних куль» та панацей</i>
1990-ті	<i>антитеза</i> : паралельні процеси проти послідовних	<i>час – це гроші</i> : люди зазвичай інвестують у програмне забезпечення, щоб отримати прибуток, і чим швидше буде встановлено програмне забезпечення, тим швидше надійдуть прибутки – за умови, що воно задовільної якості; <i>робіть програмне забезпечення, корисне для людей</i> , адже це друга складова інженерії	<i>поспішай повільно</i> – надмірно амбіційні ранні версії, як правило, призводять до неповних та несумісних специфікацій і багатьох переробок
2000-ні	<i>антитеза та частковий синтез</i> : гнучкість та цінність	<i>якщо зміни є швидкими, адаптивність краще за повторюваність</i> ; <i>розглядайте та задовольняйте інноваційні пропозиції всіх зацікавлених сторін</i> – якщо критично важливими зацікавленими сторонами знехтувати, вони, як правило, контратакують або відмовляються від участі, що веде до загальних	<i>уникайте закоханих у ваші гасла</i> – принцип YAGNI (You Aren't Going to Need It – «Вам це не знадобиться») не завжди є вірним

Період	Категорія діалектики	«Вічні» принципи	Застарілі практики
		втрат	
2010-ті		реально оцінюйте власні можливості – деякі системи систем можуть бути надто великими та складними; майте стратегію відходу: керуйте очікуваннями, так, що якщо щось піде не так, буде запасний варіант	не довіряйте усьому, що читаєте

Б. Боем вказує, що студенти, які навчатимуться ПЗ у найближчі 20 років, будуть продуктивними у своїй професії у 2040-ві, 2050-ті та, ймовірно, 2060-ті роки: «Зростання темпів змін продовжує прискорюватися, як і складність програмно насичених системи або систем систем, які потребують проектування. Це створює багато серйозних, але хвилюючих викликів для освіти в галузі програмної інженерії, зокрема: підтримувати курси та програми підготовки у постійно оновлюваному стані; передбачати майбутні тенденції та готувати студентів до них; виконувати моніторинг поточних принципів і практики та відокремлювати «вічні» принципи від застарілих практик; забезпечувати масштабування освітнього досвіду до способів, що застосовуються у великих проектах; брати участь у найсучасніших наукових дослідженнях та практиці з інженерії програмного забезпечення та включати результати у навчальні плани; допомагати студентам навчатися навчатися шляхом аналізу власних досягнень, спрямованих на майбутнє навчальних ігор та вправ, а також участі в наукових дослідженнях; і навчатися протягом усього життя стільки ж, скільки інженери з програмного забезпечення продовжують займатися своєю професією» [5, с. 25].

5 Етапи розвитку інженерії програмного забезпечення

У грудні 1969 році в США відбувся третій симпозіум з ПЗ, перший том матеріалів якого відкривала програмна доповідь Ю. Т. Ту (Julius T. Tou) «Нова професія: інженерія програмного забезпечення» («Software Engineering – A New Profession») [26]. Автор, характеризує роль інженерії у розвитку людства, вказує, що вона значною мірою звільнила людину від фізичної праці та рутинних розумових дій, надавши інструменти як для нових наукових відкриттів, так й нових видів творчості. Стрімкий розвиток інженерії, поява нових її галузей змінили зміст інженерної освіти – від утилітарної до науково-теоретичної. Так само як винахід парової машини наприкінці вісімнадцятого сторіччя дозволив замінити м'язову силу людей і тварин рушійною силою машин, винахід цифрового комп'ютера після Другої світової війни дозволив замінити багато людських розумових задач, таких як арифметичні обчислення, зберігання даних та ведення обліку, комп'ютерними операціями: «Ми зараз переходимо до стадії, на якій доцільно передбачити заміну деяких вищих розумових задач людини машинами. Це включає в себе здатність розпізнавати шаблони, читати зображення, опрацьовувати дані природною мовою, отримувати інформацію та

приймати розумні рішення» [26, с. 2]. Для цього автор пропонує скористатися принципами ІПЗ. На думку Ю. Т. Ту, ІПЗ повинна охоплювати архітектуру комп'ютерів, системне та прикладне програмування [26, с. 4]. Стосовно ІПЗ автор наголошує, що «для того, щоб досягти значного прогресу у розробці програмного забезпечення, ми повинні мати набагато міцніший науковий фундамент. ... Інакше це стане ремеслом, а не галуззю знань» [26, с. 5]. Ю. Т. Ту оптимістично прогнозує, «що комп'ютер стане інструментом, без якого ніхто не зможе прожити, а інженерія програмного забезпечення стане основною галуззю нашого часу», і завершує доповідь фразою Р. К. Пучинського (Roman Conrad Pucinski) [18]: «Інженерія програмного забезпечення не лише є основним ключем до вирішення проблем науки і техніки, але й має силу для формування нового суспільства, унікального в усій історії людства» [26, с. 6].

Ф. Л. Бауер, характеризуючи проектування та виробництво програмного забезпечення як промислову галузь інженерії, формулює способи боротьби із складністю великих програмних проектів: розділення на керовані частини з визначенням інтерфейсів між ними, визначення ієрархії компонентів (наприклад, деревовидної), розділення процесу розробки на окремі стадії. «Всі процеси проектування, виробництва та обслуговування повинні автоматизуватися, ... зокрема:

- автоматичне оновлення та контроль якості документації;
- вибіркоче розповсюдження інформації серед усіх співробітників проекту;
- моніторинг термінів виконання проекту;
- збір даних для моделювання;
- збір даних для контролю якості;
- автоматичне генерування керівництв користувача та матеріалів із технічного обслуговування» [3].

Значну роль в ІПЗ Ф. Л. Бауер відводить структурному програмуванню – новому (на той час) підходу, запропонованому Е. В. Дейкстрою, ілюструючи його ієрархією абстракцій мов розв'язання задач (від людської до машинної) та уводячи поняття проміжної мови та абстрактної машини, які надають можливість перенесення програмних систем між різними апаратними платформами. Використання технології структурного програмування, на думку Ф. Л. Бауера, надає можливість розробки *мобільного та адаптивного програмного забезпечення* (portable software and adaptable software) на основі компонентного підходу (software components) та генераторів програмного коду («macro generators which allow the specification of new macros»). Подальший розвиток ІПЗ він пов'язує з розробкою відповідних технологій та засобів у співпраці університетських та промислових фахівців.

Х. Д. Мілс (Harlan D. Mills) розширює ІПЗ до математичної основи, необхідної для управління комп'ютерами у складних застосуваннях [15]. Автор вказує, що тривалий час програмне забезпечення, незважаючи на визнання важливості застосування при його розробці математичного апарату логіки, теорії автоматів та лінгвістики, створювалось емпіричним способом без його використання. Фактором, що змушує програмістів-практиків звернутися до

математичних основ ПЗ, є зростання складності програмного забезпечення та нездатність емпіричних підходів до його розробки та управління впоратися з нею [15, с. 1199]. Тому мета, яку поставили перед собою Х. Д. Міллс разом з іншими авторами [30], – показати програмістам, як, використовуючи систематичні методи аналізу і синтезу програм, зробити свою роботу більш продуктивною, ознайомити їх з прийомами проектування надійного і ефективного програмного забезпечення. Вирішення цієї проблеми приводить до двох важливих результатів: 1) дозволяє контролювати творчу діяльність – невід’ємний компонент процесу проектування програмного забезпечення та 2) робить можливим застосування математичного апарату доведення правильності до вже створених програм, як тих, що працюють, так й тих, що потребують налагодження. Так як налагодження – це складний і дорогий процес, його спрощення веде до підвищення і надійності, і продуктивності. Уведення додаткового контролю процесу проектування програмного забезпечення дозволяє приділяти більше уваги питанням його ефективності і створює сприятливі можливості для розробки програмних проектів, які враховують умови реалізації.

Х. Д. Міллс вказує, що роботи Е. В. Дейкстри і Ч. Е. Р. Хоора (Charles Antony Richard Hoare) зіграли значну роль для переосмислення та переоцінки програмного забезпечення і розгляді його як галузі математики: «Як тільки програмування вийшло за межі допустимої складності, відбувся поворот до дисципліни, головною метою якої протягом століть було застосування ефективного структурування з метою подолання складності, що, здавалося, не піддається управлінню. Ця дисципліна, всім нам більш-менш знайома, називається математикою. Якщо ми погодимося з правильністю судження про те, що математичні методи є найбільш ефективним засобом подолання складності, у нас не залишається іншого вибору, як тільки перебудувати область програмування таким чином, щоб стало можливим застосовувати ці методи, бо інших засобів не існує» [10, с. 4.2].

Х. Д. Міллс трактує програму як те, що необхідно для управління комп’ютерним обладнанням (апаратно-центричний підхід), а програмне забезпечення – як гармонійну систему, що забезпечує взаємодію програм, різноманітного апаратного забезпечення та людських ресурсів (системний підхід) [15]. На прикладі операційної системи автор вводить поняття абстрактної машини як об’єднання програмного та апаратного забезпечення; у свою чергу, множина таких машин теж може утворювати абстрактну машину, якою будуть керувати її користувачі, які, у свою чергу, можуть бути й її складовими (агентами) – тоді термін «програмне забезпечення» поширюється також на посібники користувача та керівництва оператора (для людей, які відповідають їх ролі в системі). Виходячи з цього, автор визначає програмне забезпечення як «логічну доктрину гармонійної співпраці людей і машин»: «коротше кажучи, програмне забезпечення визначається як система абстрактних машин, деякі з яких викликають інші абстрактні машини, доки люди та апаратні засоби не будуть досягнуті як найважливіші агенти дій у системі» [15, с. 1201]. Пов’язуючи розвиток програмного забезпечення з опрацюванням даних, Х. Д. Міллс підкреслює не лише важливість програмного забезпечення для опрацювання

даних, а й те, що у розвинених країнах опрацювання даних стало важливим національним активом в управлінні та організації промислових ресурсів, який суттєво залежить від якості програмного забезпечення.

Математичною основою подолання складності програмного забезпечення Х. Д. Міллс вважає формалізоване доведення правильності програм за допомогою логіки Хоора та структурного програмування Е. В. Дейкстри: так, на с. 1202-1204 статті [15] у науково-популярному виданні «Science» він наводить аксіоматику послідовних процесів у термінах слідування, розгалуження та умовного повторення, показує зв'язок структурних програм та алгебраїчних функцій, описує способи синхронізації процесів та організації абстрактних машин. Розширений варіант можна знайти у роботі 1972 року «Математичні основи структурного програмування», у передмові до якої Х. Д. Міллс пише, що «ідеї [структурного програмування] є потужним інструментом для мисленнєвого поєднання статичного тексту програми з динамічним процесом її виконання. Це нове співвідношення між програмою та процесом дозволяє досягти нового рівня точності в програмуванні – ... від розчарувань, проб та помилок до систематичної, контрольованої за якістю діяльності. Однак для того, щоб запровадити ... таке точне програмування у промислову діяльність, ідеї структурованого програмування повинні бути сформульовані як технічні стандарти» [14, с. II]. «Таким чином, вимоги реальності полягають у тому, щоб звичайні програмісти з пересічними здібностями змогли навчитися писати програми, які з самого початку не містили б помилок. Знання того, що це можливо, – вже наполовину виграна битва. А вміння писати такі програми – шлях до остаточної перемоги. Набуваючи досвіду в написанні правильних програм, програміст переходить на новий психологічний рівень, що дозволяє подолати укорінену думку про те, що оцінити правильність програм, не вдаючись до експерименту, дуже важко» [29, с. 13-14].

Слід зазначити, що навіть через чверть століття після першої згадки про ПЗ М. Шоу (Mary Show) вказувала, що ПЗ «ще не справжня галузь інженерії, але має потенціал стати нею» [22]. Таблиця 2 з [22, с. 20] показує, що за перші 20 років свого існування ПЗ пройшла розвиток, порівняний з 200 роками традиційних галузей інженерії. Ключовими у переході від «аматорського» програмування до ПЗ М. Шоу вважає відокремлення Д. Кнудом у 1967 р. алгоритму від програми, введення Р. В. Флойдом поняття формальної верифікації програм.

У 1976 році Б. Боем (Barry Boehm) визначив ПЗ як «практичне застосування наукових знань до проектування та конструювання комп'ютерних програм та пов'язаної документації, необхідної для їх розробки, експлуатації та підтримки» [7, с. 1226]. Наукові принципи Б. Боем пропонує застосовувати за 4 напрямками:

- у життєвому циклі програмного забезпечення – це принципи побудови компонентів та деталізованого проектування, практично відсутні для системного проектування та інтеграції, наприклад, алгоритми та теорія автоматів;

- у прикладному програмуванні – це деякі принципи для складних програмних систем, практично відсутні для програмного забезпечення, наприклад, дискретні математичні структури;
- у економіці програмного забезпечення – це декілька принципів, які застосовуються до економічних систем, таких як алгоритми;
- у професійній підготовці це декілька принципів, які сформульовані для засвоєння техніками-програмістами, таких як структурування коду та базові математичні бібліотеки [7, с. 1239].

Таблиця 2. Характеристика етапів розвитку ПЗ (за [22])

	1960±5 років: «програмування аби як»	1970±5 років: «програмування у малому»	1980±5 років: «програмування у великому»
Характер задачі	Невеликі програми	Алгоритми та програми	Інтерфейси, структури систем управління
Подання даних	Структури і символи	Структури даних і типи	Бази даних тривалого зберігання, символи, а також числа
Способи управління	Елементарне розуміння про потоки управління	Програми виконуються один раз і завершуються	Набори програм виконуються постійно
Подання специфікацій	Мнемоніки, точні письмові інструкції	Специфікації простого введення/виведення	Системи зі складними специфікаціями
Простір станів	Стан погано відрізняється від контролю	Невеликий, простий простір станів	Великий, структурований простір станів
Фокус менеджменту	Відсутній	Індивідуальні зусилля	Командні зусилля, обслуговування системи протягом всього часу експлуатації
Інструменти, методи	Асемблери, дампи пам'яті	Мови програмування, компілятори, компоновальники, завантажувачі	Середовища, інтегровані засоби, документи

Стаття Б. Боєма, опублікована більше 40 років тому, містить всі основні складові сучасної ПЗ – незважаючи на застарілість прикладів, виокремлені ним наукові принципи довели свою життєздатність. У 1987 році, аналізуючи історичні аспекти ПЗ [11, с. 9-12], Б. Боєм виокремлює три ранні роботи, що мали значний вплив на становлення і розвиток ПЗ.

Перша з них узагальнює досвід проектування автоматизованої системи управління авіацією та протиповітряною обороною SAGE (Semi-Automatic Ground Environment) [4] – найбільш амбітного проекту інформаційної системи протиповітряної оборони США та Канади 1950-х рр., який об'єднав провідних радарних інженерів, інженерів зв'язку, комп'ютерних інженерів та нових інженерів – з програмного забезпечення. У рамках проекту SAGE була розроблена Lincoln Labs Utility System на допомогу тисячам програмістів, що брали участь у розробці програмного забезпечення SAGE. Вона включала в себе

асемблер, бібліотеку та систему керування збірками, ряд корисних утіліт, а також засоби тестування та налагодження. Система SAGE успішно задовольняла технічним специфікаціям приблизно через один рік. Провідний розробник SAGE Г. Д. Бенінгтон (Herbert D. Benington) вказував, що йому було легко виділити той чинник, що призвів до такого успіху: «ми всі були інженерами і були навчені організувати наші зусилля на інженерних засадах». У 1956 році він узагальнив досвід розробки SAGE у описі процесу проектування великої програмної системи, що складається з 9 фаз:

1 – операційний план (operational plan) визначає широкі вимоги до проектування всієї системи керування, що складається з машини, оператора та програмної системи. Цей план повинен бути підготовлений спільно з інженерами комп'ютерних систем та кінцевими користувачами системи;

2 – експлуатаційні специфікації (machine specifications, operational specifications), які точно визначають «передавальну функцію» системи управління. У цьому поданні комп'ютер, його термінальне обладнання та системна програма розглядаються як «чорна скринька»;

3 – програмні специфікації (program specifications) описують реалізацію «чорної скриньки» системною програмою. Ці специфікації організовують програму в підпрограми-компоненти та таблиці, вказують основні канали міжпрограмної взаємодії, а також спільне використання машинного часу та даних кожною підпрограмою;

4 – після завершення операційних та програмних специфікацій, готуються детальні специфікації кодування (coding specifications), які визначають «передавальну функцію» кожного компонента підпрограми у термінах опрацювання глобальних та локальних даних;

5 – кожен компонент програмується (coding) за допомогою специфікацій кодування. В ідеалі цей етап має бути простим механічним перекладом; насправді, програмування розкриває невідповідність специфікацій кодування (а іноді й операційних специфікацій);

6 – після програмування кожна підпрограма окремо тестується на відповідність параметрів (parameter testing). На цьому етапі тестування виконується у середовищі, яке імітує відповідні частини програмної системи. Кожен тест, виконаний на цій фазі, документується у наборі специфікацій тесту, який деталізує використовуване середовище та отримані результати;

7 – після завершення тестування параметрів підпрограм, поступово компонується та перевіряється вся система (assembly testing), використовуючи спочатку модельні, а потім реальні дані;

8 – після завершення збирання програми, вона перевіряється в його операційному середовищі у стресовому режимі (shakedown);

9 – програма готова до роботи та оцінки (evaluation).

Модель Г. Д. Бенінгтона явно не передбачає повернень до попередніх фаз (хоча у статті й згадується про необхідність перегляду навіть першої фази за результатами реалізації специфікацій кодування під час програмування), тому надалі подібні моделі назвали «водоспадними».

Різке зменшення вартості машинного часу наприкінці 1950-х рр., пов'язане, насамперед зі зміною апаратної складової (зменшення розміру та енергоспоживання транзисторів, збільшення часу їх неперервної роботи, об'єднання у інтегральні схеми тощо), призвело до появи підходу «спочатку закодуй, а потім вже перевіряй та виправляй», який і призвів до обговорюваної вище кризи програмного забезпечення.

Так, вже у процесі реалізації SAGE програмна складова системи стала більш значущою, ніж апаратна: Б. Боем вказує, що «навіть у SAGE все більш значущими ставали адресовані психологам питання людино-машинної взаємодії, ніж питання, адресовані радарним інженерам... Швидке розширення попиту на програмне забезпечення перевищило пропозицію інженерів та математиків. Програма SAGE почала наймати та навчати розробки програмного забезпечення фахівців з гуманітарних, соціальних наук, іноземних мов та мистецтва. Для таких неінженерних фахівців ... й був набагато комфортніший підхід «кодууй та виправляй». Вони часто були дуже креативними, але їх виправлення часто призвело до важкого в обслуговуванні спагетті-коду. ... Суттєвою в цьому відношенні була «хакерська культура» ..., а частими рольовими моделями були «програмісти-ковбої»...» [5, с. 13-14].

Подальшим узагальненням досвіду проектування SAGE та інших оборонних систем, що потребують реагування у реальному часі (зокрема, протиракетною системою Plato) є стаття 1961 року У. Хоз'є (W. A. Hosier). На відміну від моделі Г. Д. Бенінгтона, У. Хоз'є явно показав, що майже на всіх етапах проектування у результаті зворотного зв'язку (feedback) є можливими повернення до початку проектування (включно із вибором апаратних засобів).

У. У. Ройс (Winston Walker Royce) за класифікацією Б. Боема відноситься до фахівців-емпіриків – так, він не пропонує жодних наукових принципів ПЗ, а описує виключно власний досвід розробки систем високої складності – програмних систем для планування, управління та післяпольотного аналізу місії космічних апаратів.

Для зменшення ризиків, пов'язаних із застосуванням ітеративної схеми проектування, У. У. Ройс пропонує внести до неї п'ять доповнень:

1. *етап попереднього проектування програми* між етапом визначення програмних вимог та етапом аналізу. Для реалізації цього етапу У. У. Ройс пропонує:
 - a. розпочати процес проектування саме із проектувальниками програми, а не аналітиками або програмістами;
 - b. розробити, визначити та розподілити режими обробки даних навіть за ризику помилок: способи опрацювання, функції, структуру бази даних, розподіляти час виконання, інтерфейси та режими роботи з операційною системою, описати обробку вводу та виводу та визначити попередні операційні процедури;
 - c. написати зрозумілий, інформативний та актуальний оглядовий документ, з якого кожен учасник проекту зможе отримати елементарне розуміння проєктованої системи;

2. *забезпечення актуальності та повноти документації* – за У. У. Ройсом, якомога більше: «Першим правилом управління розробкою програмного забезпечення є безжальне виконання вимог до документації. ... перший крок полягає у вивченні стану документації. Якщо з документацію серйозна проблема, моя перша рекомендація проста: замінити менеджмент проекту; зупинити всі дії, не пов'язані з документацією; привести документацію до відповідних стандартів. Управління програмним проектом просто неможливе без дуже високого ступеня документування» [20, с. 5];
3. *«подвійне» проектування* – «після документації другим найважливішим критерієм успіху полягає у новизні програмного продукту. Якщо програмне забезпечення розробляється вперше, його остаточна версія, яка поставляється клієнту, повинна бути другою версією. ... Зауважимо, що це просто процес, виконаний у мініатюрі, масштабі, відносно невеликому по відношенню до загальних зусиль. ... За допомогою моделювання [керівник проекту] може, принаймні, виконати експериментальну перевірку деяких ключових гіпотез та [занадто оптимістичних людських очікувань]» [20, с. 7];
4. *планування, управління та моніторинг тестування програмного забезпечення.* Як зауважує У. У. Ройс, більш раннє звернення до процедур, пов'язаних із тестуванням, необхідно тому, що етап тестування є одним із останніх етапів проектування, а тому може стати «вузьким місцем» проекту, подолання проблем на якому потребуватиме додаткових ресурсів;
5. *залучення замовника* – «у зв'язку із тим, проектування програмного забезпечення має широкі можливості для інтерпретації навіть після попередніх погоджень, важливо залучати замовника на формальній основі» [20, с. 8].

На рис. 2 показано деталізовану «водоспадну» модель. Складність цієї емпіричної моделі, незважаючи на її спрощеність, вже є досить велика. Ураховуючи, що вона спрямована на подолання складності програмного забезпечення, можна зробити висновок про те, що однією із головних причин кризи проектування програмного забезпечення наприкінці 1960-х рр. була перевага емпіричних підходів над науковими, про що в ретроспективі й писав Б. Боем, характеризуючи ранні «водоспадні» моделі.

У 1990 році IEEE визначав ІПЗ як [2, с. 67]:

- (1) застосування систематичного, дисциплінованого, кількісного підходу до розробки, експлуатації та обслуговування програмного забезпечення; тобто застосування інженерії до програмного забезпечення;
- (2) вивчення підходів до (1).

У визначенні IEEE 2010 року емпіричний підхід відійшов на друге місце у порівнянні із науковим:

- (1) систематичне застосування наукових та технологічних знань, методів, досвіду проектування, впровадження, тестування та документування програмного забезпечення;

- (2) застосування систематичного, дисциплінованого, кількісного підходу до розробки, експлуатації та обслуговування програмного забезпечення; тобто застосування інженерії до програмного забезпечення [1].

Е. Бьйоргер (Egon Voerger) [11, с. 12-17], розробник одного з таких наукових методів – формального методу машин абстрактних станів (ASM – Abstract State Machines) – вказує, що метод ASM є подальшим узагальненням багаторічної теорії та практики структурного програмування, що розв’язує проблему базової моделі, яка є фундаментальною для ПЗ. «Базова модель системи S , що сама по собі є дуже часто не чітко математично визначеною системою, – це математична модель, яка формалізує основні інтуїтивні уявлення, концепції та операції S без кодування таким чином, що модель може бути розпізнана системним експертом «шляхом інспектування» і у такий спосіб підтверджено, що вона є правильним, адекватним та точним поданням S . Гарні базові моделі відіграють вирішальну роль для доказу правильності специфікацій складних систем у цілому» [11, с. 15].

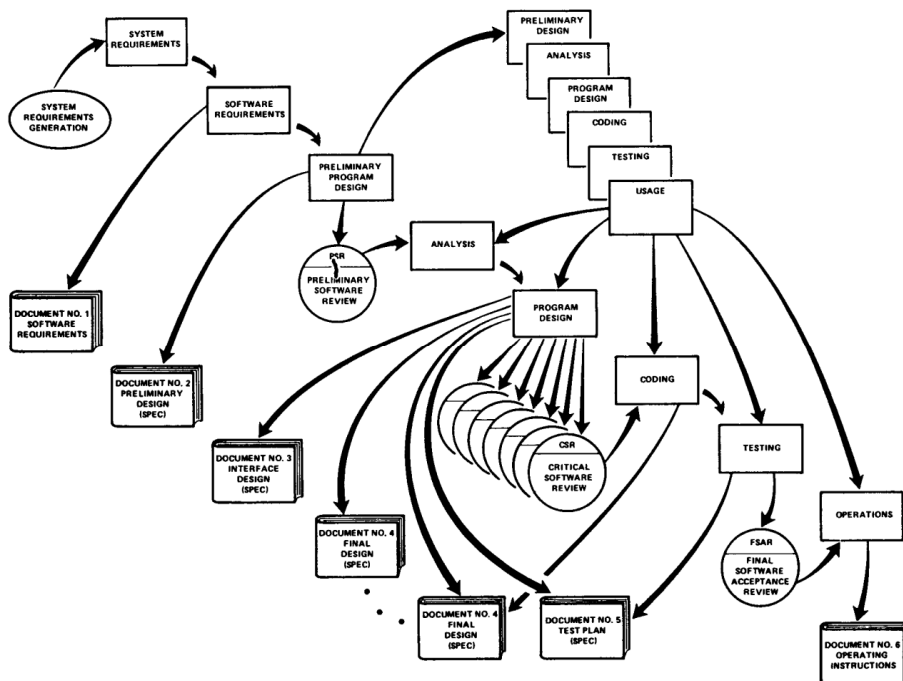


Рис. 2. Модель розробки програмного забезпечення (за У. У. Ройсом)

Сьогодні метод ASM є практичним та науково обґрунтованим методом системної інженерії, що дозволяє подолати розрив між двома боками проектування програмного забезпечення:

- людське розуміння і формулювання реальних проблем (захоплення вимог шляхом точного моделювання високого рівня на рівні абстракції, визначеною заданою галуззю застосування);
- розгортання їх алгоритмічних рішень машинами, що виконують код, на змінних платформах (визначення конструктивних рішень, деталей системи та реалізації).

Моделі ASM можуть бути перевірені як математично (через механізми логічного виведення), так й емпірично (через виконання тестів), що робить їх містком між теоретичними комп'ютерними науками та емпіричними методами ІІЗ.

Через 6 років після статті [22] М. Шоу у [23] суттєво спростила таблицю 1 та доповнила її четвертим етапом розвитку ІІЗ – 1990±5 років: «програмування у світі», яке характеризувалось розподіленими системами з відкритими та змінними специфікаціями, акцентом на взаємодії підсистем та співробітництві незалежних процесів. М. Шоу констатувала, що розвиток ІІЗ після третього етапу змінився: об'єктно-орієнтовані технології розпочали витісняти класичні технології структурного програмування, а розвиток модельного опису окремих галузей досяг рівня, що надав можливість створювати за моделями генератори програмного коду. Інші фактори змін – бурхливий розвиток Інтернет-технологій. Це розширило галузь ІІЗ з проектування великих програмних систем до підтримки програмного забезпечення, що широко використовується та створюється спільнотою.

У 1993 році Ф. Дж. Баклі (Fletcher J. Buckley) писав, що для визначення професії інженера з програмного забезпечення необхідно визначити 4 складові: професійну галузь, технічні вимоги до кваліфікації професіонала, необхідні (entry-level) вимоги до професії та набір правил, що утворюють професійну етику [9]. На думку автора, ІІЗ разом з комп'ютерною інженерією є частинами системної інженерії.

6 Висновки

Аналіз передумов виділення ІІЗ як окремої галузі знань та основних етапів її розвитку надав можливість зробити наступні висновки:

1. На сьогодні ІІЗ є невід'ємною складовою переважної більшості інновацій у всіх сферах розвитку суспільства, науки та техніки, пропонуючи системні, практичні, економічно вигідні рішення для обчислювальних задач та задач опрацювання інформації.
2. За 50 років від першого задокументованого використання терміну «інженерія програмного забезпечення», який втілював у собі загальне побажання того, що розробка програмного забезпечення має бути подібною до інженерії і ґрунтуватись на чітко визначених теоретичних засадах та перевірених методах, накопичено значний досвід проектування, впровадження, тестування та документування програмного забезпечення, виокремлено системні наукові, технологічні підходи і методи до проектування та конструювання

комп'ютерних програм. У той же час дослідники зазначають, що ІІЗ ще досі не досягла того рівня сталості, як інші галузі інженерії.

3. Метою ІІЗ є розробка програмних систем з наперед визначеними рівнями якості, надійності та ефективності в умовах обмеження ресурсів (часових, людських, матеріальних, програмно-апаратних тощо). Аналіз історичних етапів розвитку ІІЗ показав, що, незважаючи на загальне визнання важливості застосування при розробці програмного забезпечення математичного апарату логіки, теорії автоматів та лінгвістики, воно створювалась емпіричним способом без його використання. Фактором, що змушує програмістів-практиків звернутися до математичних основ ІІЗ, є зростання складності програмного забезпечення та нездатність емпіричних підходів до його розробки та управління впоратися з нею.
4. Сучасна ІІЗ базується на трьох групах ключових принципів: основні концепції комп'ютерних наук, пов'язані зі структурами даних, алгоритмами, мовами програмування та їх семантикою, аналізом, обчислювальністю, моделями обчислень тощо; основи інженерії, пов'язані з архітектурою, процесами інженерії, компромісами та витратами, стандартизацією, якістю та гарантіями та інші складові, що забезпечують підхід до проектування та вирішення проблем; соціально-економічні основи, які включають процес створення та еволюції артефактів, а також питання, пов'язані з політикою, ринками, зручністю використання та соціально-економічними впливами; це забезпечує основу для формування інженерних артефактів, які будуть відповідати їхньому призначенню.
5. Суспільна значущість професії інженера з програмного забезпечення породжує нагальну необхідність розробки, уточнення відповідних освітніх програм їх підготовки. У навчанні ІІЗ («технології програмування») із самого початку гостро постала проблема швидкого застарівання технологічного змісту навчання, розв'язання якої полягає у його фундаменталізації через виокремлення базових основ галузі. Визначено, що опанування основ комп'ютерних наук (інформатики) є фундаментом професійної підготовки з ІІЗ.

Список використаних джерел

1. 24765-2010 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Vocabulary. – New York : IEEE, 2010. – VI, 410, [2] p. – DOI : 10.1109/IEEESTD.2010.5733835.
2. 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology. – New York : IEEE, 1990. – 83 p. – DOI : 10.1109/IEEESTD.1990.101064.
3. Bauer F. L. Software Engineering / Bauer F. L. // Information Processing 71: Proceedings of IFIP Congress 71, Ljubljana, Yugoslavia, August 23-28, 1971 / Editors : C. V. Freiman, J. E. Griffith, J. L. Rosenfeld. – Amsterdam : North-Holland Publishing Company, 1972. – Vol. 1 – Foundations and Systems. – P. 530-538.
4. Benington H. D. Production of Large Computer Programs / H. D. Benington // Proceedings of Symposium on advanced programming methods for digital computers : Washington,

- D.C., June 28, 29, 1956 / US Navy Mathematical Computing Advisory Panel, US Office of Naval Research. – [Washington] : Office of Naval Research, Dept. of the Navy, [1956?]. – P. 15-28. – (ONR symposium report, ACR-15)
5. Boehm B. A View of 20th and 21st Century Software Engineering / Barry Boehm // Proceedings of the 28th international conference on Software engineering ICSE'06. Shanghai, China – May 20-28, 2006. – New York : ACM, 2006. – P. 12-29. – DOI : 10.1145/1134285.1134288.
 6. Boehm B. A View of 20th and 21st Century Software Engineering : ICSE 2006 Keynote Address [Electronic resource] / Barry Boehm. – May 25, 2006. – 54 p. – Access mode : <https://isr.uci.edu/icse-06/program/keynotes/Boehm-Keynote.ppt>.
 7. Boehm B. W. Software Engineering / Boehm B. W. // IEEE Transactions on Computers. – 1976. – Vol. 25. – Iss. 12. – P. 1226-1241.
 8. Brooks F. P. Jr. No Silver Bullet – Essence and Accident in Software Engineering / Frederick. P. Brooks, Jr. // Computer. – 1987. – Vol. 20. – Iss. 4. – P. 10-19. – DOI : 10.1109/MC.1987.1663532.
 9. Buckley F. J. Standards: defining software engineering as a profession / Fletcher J. Buckley // Computer. – 1993. – Vol. 26. – Issue 8. – P. 76-78. – DOI : 10.1109/2.223554.
 10. Dijkstra E. W. On a Methodology of Design / Dijkstra E. W. // MC-25 Informatica Symposium / Ed. Jacobus Willem Bakker. – Amsterdam : Mathematisch Centrum, 1971. – P. 4.1-4.10. – (Mathematical Centre tracts, 37)
 11. History of Software Engineering : Position Papers for Dagstuhl Seminar 9635 on August 26 – 30, 1996 organized by William Aspray, Reinhard Keil-Slawik and David L. Parnas [Electronic resource] / Editors : Andreas Brennecke, Reinhard Keil-Slawik. – Paderborn, July 1997. – III+57 p. – Access mode : <https://www.dagstuhl.de/Reports/96/9635.pdf>.
 12. Hosier W. A. Pitfalls and Safeguards in Real-Time Digital Systems with Emphasis on Programming / W. A. Hosier // IRE Transactions on Engineering Management. – 1961. – Vol. Em-8. – Iss. 2. – P. 99-115. – DOI:10.1109/iret-em.1961.5007599.
 13. Mead N. R. Half a Century of Software Engineering Education: the CMU Exemplar / Nancy R. Mead, David Garlan, Mary Shaw // IEEE Software. – 2018. – DOI : 10.1109/MS.2018.290110743.
 14. Mills H. D. Mathematical Foundations for Structured Programming [Electronic resource] / Harlan D. Mills. – Gaithersburg : Federal Systems Division, International Business Machines Corporation, 1972. – IV, 62 p. – (The Harlan D. Mills Collection). – Access mode : http://trace.tennessee.edu/utk_harlan/56.
 15. Mills H. D. Software Engineering / Harlan D. Mills // Science. – Vol. 195. – Iss. 4283. – 18 March 1977. – P. 1199-2105.
 16. Oettinger A. A. Letter to the ACM membership / Anthony A. Oettinger // Communications of the ACM. – 1966. – Vol. 9. – No. 8. – P. 545-546.
 17. Perlis A. J. Identifying and developing curricula in software engineering / Alan J. Perlis // Proceedings of the May 14-16, 1969, spring joint computer conference on XX - AFIPS '69 (Spring). Boston, Massachusetts. – New York : ACM, 1969. – P. 540-541. – DOI: 10.1145/1476793.1476877.
 18. Pucinski R. C. The Challenge for the 1970s in Information Retrieval / Roman C. Pucinski // Software Engineering COINS III : Proceedings of the Third Symposium on Computer and Information Sciences held in Miami beach, Florida, December, 1969 / Edited by Julius T. Tou. – Volume 2. – New York ; London : Academic Press, 1971. – P. XVII-XX. – DOI : 10.1016/B978-0-12-696202-4.50006-1.

19. Randell B. Fifty Years of Software Engineering - or - The View from Garmisch [Electronic resource] / Brian Randell // arXiv:1805.02742 [cs.SE]. – 2018. – 9 p. – Access mode : <https://arxiv.org/abs/1805.02742>.
20. Royce W. W. Managing the Development of Large Software Systems: Concepts and Techniques / Dr. Winston W. Royce // Proceedings of IEEE WESCON, Los Angeles, 25-28 August 1970. – P. 1-9.
21. Shaw M. Continuing Prospects for an Engineering Discipline of Software / Mary Shaw // IEEE Software. – 2009. – Vol. 26. – Issue 6. – P. 64-67. – DOI : 10.1109/ms.2009.172.
22. Shaw M. Prospects for an Engineering Discipline of Software / Mary Shaw // IEEE Software. – 1990. – Vol. 7. – No. 6. – P. 15-24.
23. Shaw M. Three Patterns that help explain the development of Software Engineering / Mary Shaw // History of Software Engineering : Position Papers for Dagstuhl Seminar 9635 on August 26 – 30, 1996 organized by William Aspray, Reinhard Keil-Slawik and David L. Parnas [Electronic resource] / Editors : Andreas Brennecke, Reinhard Keil-Slawik. – Paderborn, July 1997. – P. 52-56. – Access mode : <https://www.dagstuhl.de/Reports/96/9635.pdf>.
24. Software Engineering : Report on a Conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968 / Editors : Peter Naur and Brian Randell. – Brussels : Scientific Affairs Division, NATO, January 1969. – 231 p.
25. Software Engineering Techniques : Report on a Conference sponsored by the NATO Science Committee, Rome, Italy, 27th to 31st October 1969 / Editors : J. N. Buxton and B. Randell. – Brussels : Scientific Affairs Division, NATO, April 1970. – 164 p.
26. Tou J. T. Software Engineering – A New Profession / Julius T. Tou // Software Engineering COINS III : Proceedings of the Third Symposium on Computer and Information Sciences held in Miami beach, Florida, December, 1969 / Edited by Julius T. Tou. – Volume 1. – New York ; London : Academic Press, 1970. – P. 1-6. – DOI : 10.1016/B978-0-12-395495-4.50009-X.
27. Williams A. S. Software engineering: tools of the profession : Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science / Arrena Sue Williams ; Naval Postgraduate School. – Monterey, 1976. – 92 p.
28. Брукс Ф. П. мл. Как проектируются и создаются программные комплексы / Ф. П. Брукс мл. – М. : Наука, 1979. – 152 с. – (Библиотечка программиста)
29. Дзержинский Ф. Я. Что такое программная инженерия / Дзержинский Ф. Я., Райков Л. Д. // Программирование. – 1990. – № 2. – С. 67-79.
30. Лингер Р. Теория и практика структурного программирования / Р. Лингер, Х. Миллс, Б. Уитт. – М. : Мир, 1982. – 406 с.
31. Про затвердження Переліку пріоритетних напрямів освіти і науки щодо навчання студентів та аспірантів, стажування наукових і науково-педагогічних працівників у провідних вищих навчальних закладах та наукових установах за кордоном у 2012 році [Електронний ресурс] : Наказ № 315, Перелік / МОНмолодьспорт України. – 20.03.2012. – Режим доступу : <http://zakon2.rada.gov.ua/laws/show/z0500-1>.
32. Про затвердження переліку пріоритетних тематичних напрямів наукових досліджень і науково-технічних розробок на період до 2020 року [Електронний ресурс] : Постанова № 942, Перелік / Кабінет Міністрів України. – 07.09.2011. – Режим доступу : <http://zakon5.rada.gov.ua/laws/show/942-2011-%D0%BF>.

References (translated and transliterated)

1. 24765-2010 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Vocabulary. IEEE, New York (2010). doi:10.1109/IEEESTD.2010.5733835
2. 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology. IEEE, New York (1990). doi:10.1109/IEEESTD.1990.101064
3. Bauer, F.L.: Software Engineering. In: Freiman, C.V., Griffith, J.E., Rosenfeld, J.L. (eds.) Information Processing 71: Proceedings of IFIP Congress 71, Ljubljana, Yugoslavia, August 23-28, 1971, vol. 1 – Foundations and Systems, pp. 530–538. North-Holland Publishing Company, Amsterdam (1972)
4. Benington, H.D.: Production of Large Computer Programs. In: Proceedings of Symposium on advanced programming methods for digital computers, Washington, D.C., June 28, 29, 1956, pp. 15–28. Office of Naval Research, Dept. of the Navy, Washington (1956)
5. Boehm, B.: A View of 20th and 21st Century Software Engineering. In: Proceedings of the 28th international conference on Software engineering ICSE'06. Shanghai, China – May 20-28, 2006, pp. 12–29. ACM, New York (2006). doi:10.1145/1134285.1134288
6. Boehm, B.: A View of 20th and 21st Century Software Engineering: ICSE 2006 Keynote Address. <https://isr.uci.edu/icse-06/program/keynotes/Boehm-Keynote.ppt> (2006). Accessed 11 Jun 2018
7. Boehm, B.W.: Software Engineering. IEEE Transactions on Computers. **25**(12), 1226–1241 (1976)
8. Brooks, F.P.Jr.: No Silver Bullet – Essence and Accident in Software Engineering. Computer. **20**(4), 10–19 (1987). doi:10.1109/MC.1987.1663532
9. Buckley, F.J.: Standards: defining software engineering as a profession. Computer. **26**(8), 76–78 (1993). doi:10.1109/2.223554
10. Dijkstra, E.W.: On a Methodology of Design. In: Bakker, J.W. (ed.) MC-25 Informatica Symposium, pp. 4.1–4.10. Mathematisch Centrum, Amsterdam (1971)
11. Brennecke, A., Keil-Slawik, R. (eds.) History of Software Engineering : Position Papers for Dagstuhl Seminar 9635 on August 26 – 30, 1996 organized by William Aspray, Reinhard Keil-Slawik and David L. Parnas. <https://www.dagstuhl.de/Reports/96/9635.pdf> (1997). Accessed 6 May 2018
12. Hosier, W.A.: Pitfalls and Safeguards in Real-Time Digital Systems with Emphasis on Programming. IRE Transactions on Engineering Management. **Em-8**(2), 99–115 (1961). doi:10.1109/iret-em.1961.5007599
13. Mead, N.R., Garlan, D., Shaw, M.: Half a Century of Software Engineering Education: the CMU Exemplar. IEEE Software (2018). doi:10.1109/MS.2018.290110743
14. Mills, H.D.: Mathematical Foundations for Structured Programming. Federal Systems Division, International Business Machines Corporation, Gaithersburg. http://trace.tennessee.edu/utk_harlan/56 (1972)
15. Mills, H.D.: Software Engineering. Science. **195**(4283), 1199–2105 (1977)
16. Oettinger, A.A.: Letter to the ACM membership. Communications of the ACM. **9**(8). 545–546 (1966)
17. Perlis, A.J.: Identifying and developing curricula in software engineering. In: Proceedings of the May 14-16, 1969, spring joint computer conference on XX - AFIPS '69 (Spring). Boston, Massachusetts, pp. 540–541. ACM, New York (1969). doi:10.1145/1476793.1476877
18. Pucinski, R.C.: The Challenge for the 1970s in Information Retrieval. In: Tou, J.T. (ed.) Software Engineering COINS III: Proceedings of the Third Symposium on Computer and

- Information Sciences held in Miami beach, Florida, December, 1969, vol. 2, pp. XVII-XX. Academic Press, New York, London (1971). doi:10.1016/B978-0-12-696202-4.50006-1
19. Randell, B.: Fifty Years of Software Engineering - or - The View from Garmisch. arXiv:1805.02742 [cs.SE]. <https://arxiv.org/abs/1805.02742> (2018). Accessed 23 Jun 2018
 20. Royce, W.W.: Managing the Development of Large Software Systems: Concepts and Techniques. In: Proceedings of IEEE WESCON, Los Angeles, 25-28 August 1970, pp. 1–9 (1970)
 21. Shaw, M.: Continuing Prospects for an Engineering Discipline of Software. IEEE Software. **26**(6), 64–67 (2009). doi:10.1109/ms.2009.172
 22. Shaw, M.: Prospects for an Engineering Discipline of Software. IEEE Software. **7**(6), 15–24 (1990)
 23. Shaw, M. Three Patterns that help explain the development of Software Engineering. In: Brennecke, A., Keil-Slawik, R. (eds.) History of Software Engineering : Position Papers for Dagstuhl Seminar 9635 on August 26 – 30, 1996 organized by William Aspray, Reinhard Keil-Slawik and David L. Parnas, pp. 52–56. <https://www.dagstuhl.de/Reports/96/9635.pdf> (1997). Accessed 6 May 2018
 24. Naur, P., Randell, B. (eds.): Software Engineering: Report on a Conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968. Scientific Affairs Division, NATO, Brussels (1969)
 25. Buxton, J.N., Randell, B. (eds.): Software Engineering Techniques: Report on a Conference sponsored by the NATO Science Committee, Rome, Italy, 27th to 31st October 1969. Scientific Affairs Division, NATO, Brussels (1970)
 26. Tou, J.T.: Software Engineering – A New Profession. In: Tou, J.T. (ed.) Software Engineering COINS III: Proceedings of the Third Symposium on Computer and Information Sciences held in Miami beach, Florida, December, 1969, vol. 1, pp. 1–6. Academic Press, New York, London (1970). doi:10.1016/B978-0-12-395495-4.50009-X
 27. Williams, A.S.: Software engineering: tools of the profession. Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science, Naval Postgraduate School (1976)
 28. Brooks, F.P.Jr.: The Mythical Man Month: Essays on Software Engineering. Addison-Wesley Publishing Company, Reading (1975)
 29. Dzerjinskiy, F.Ya., Raykov, L.D.: Chto takoe programmnaia inzheneriia (What is software engineering). Programirovanie. **2**, 67–79 (1990)
 30. Linger, R.C., Mills, H.D., Witt, B.I. Structured Programming: Theory and Practice. The Systems programming series. Addison-Wesley Publishing Company, Reading (1979)
 31. Pro zatverdzhennia Pereliku priorytetnykh napriamiv osvity i nauky shchodo navchannia studentiv ta aspirantiv, stazhuvannia naukovykh i naukovo-pedahohichnykh pratsivnykiv u providnykh vyshchykh navchalnykh zakladakh ta naukovykh ustanovakh za kordonom u 2012 rotsi (On Approval of the List of Priority Areas of Education and Science for Students and Postgraduate Students, internship of scientific and scientific-pedagogical workers in leading higher educational establishments and scientific institutions abroad in 2012). <http://zakon2.rada.gov.ua/laws/show/z0500-1> (2012). Accessed 21 Mar 2012
 32. Pro zatverdzhennia pereliku priorytetnykh tematychnykh napriamiv naukovykh doslidzhen i naukovo-tekhnichnykh rozrobok na period do 2020 roku (On approval of the list of priority thematic areas of scientific research and scientific and technical developments for the period until 2020). <http://zakon5.rada.gov.ua/laws/show/942-2011-%D0%BF> (2011). Accessed 21 Mar 2012

Complex network precursors of crashes and critical events in the cryptocurrency market

Andrii O. Bielinskyi and Vladimir N. Soloviev^[0000-0002-4945-202X]

Kryvyi Rih State Pedagogical University, 54, Gagarin Ave., Kryvyi Rih, 50086, Ukraine
{krivogame, vnsoloviev2016}@gmail.com

Abstract. This article demonstrates the possibility of constructing indicators of critical and crash phenomena in the volatile market of cryptocurrency. For this purpose, the methods of the theory of complex networks have been used. The possibility of constructing dynamic measures of network complexity behaving in a proper way during actual pre-crash periods has been shown. This fact is used to build predictors of crashes and critical events phenomena on the examples of all the patterns recorded in the time series of the key cryptocurrency Bitcoin, the effectiveness of the proposed indicators-precursors of these falls has been identified.

Keywords: Cryptocurrency, Bitcoin, complex system, complex networks, measures of complexity, crash, critical events, indicator-precursor.

1 Introduction

The instability of global financial systems with regard to normal and natural disturbances of the modern market and the presence of poorly foreseeable financial crashes indicate, first of all, the crisis of the methodology of modeling, forecasting and interpretation of modern socio-economic realities. The doctrine of the unity of the scientific method states that for the study of events in socio-economic systems the same methods and criteria as those used in the study of natural phenomena are applicable. Significant success has been achieved within the framework of interdisciplinary approaches and the theory of self-organization – synergetics. The modern paradigm of synergetics is a complex paradigm associated with the possibility of direct numerical simulation of the processes of complex systems evolution, most of which have a network structure, or one way or another can be reduced to the network. The theory of complex networks studies the characteristics of networks, taking into account not only their topology, but also statistical properties, the distribution of weights of individual nodes and edges, the effects of dissemination of information, robustness, etc. [1-4].

Complex systems are systems consisting of a plurality of interacting agents possessing the ability to generate new qualities at the level of macroscopic collective behavior, the manifestation of which is the spontaneous formation of noticeable temporal, spatial, or functional structures. As simulation processes, the application of quantitative methods involves measurement procedures, where importance is given to

complexity measures. I. Prigogine notes that the concepts of simplicity and complexity are relativized in the pluralism of the languages descriptions, which also determines the plurality of approaches to the quantitative description of the complexity phenomenon [5]. Therefore, we will continue to study Prigogine's manifestations of the system complexity, using the current methods of quantitative analysis to determine the appropriate measures of complexity.

The key idea here is the hypothesis that the complexity of the system before the crashes and the actual periods of crashes must change. This should signal the corresponding degree of complexity if they are able to quantify certain patterns of a complex system. Significant advantage of the introduced measures is their dynamism, that is, the ability to monitor the change in time of the chosen measure and compare it with the corresponding dynamics of the output time series. This allowed us to compare the critical changes in the dynamics of the system, which is described by the time series, with the characteristic changes of concrete measures of complexity. It turned out that quantitative measures of complexity respond to critical changes in the dynamics of a complex system, which allows them to be used in the diagnostic process and prediction of future changes.

Cryptocurrency market is a complex, self-organized system, which in most cases can be considered either as a complex network of market agents, or as an integrated output signal of such a network – a time series, for example, prices of individual cryptocurrency. Moreover, in the cryptocurrency market, to some extent, the blockchain technology is tested in general. Thus the cryptocurrency prices exhibit such complex volatility characteristics as nonlinearity and uncertainty, which are difficult to forecast and any results obtained are uncertain. Therefore, cryptocurrency price prediction remains a huge challenge.

Unfortunately, the existing nowadays classical econometric [6] and modern methods of prediction of crisis phenomena based on machine learning methods [7-11] do not have sufficient accuracy and reliability of prediction.

Thus, lack of reliable models of prediction of time series for the time being will update the construction of at least indicators which warn against possible critical phenomena or trade changes etc. This work is dedicated to the construction of such indicators-precursors based on the theory of complex networks and adapt them in order to study the critical and crash phenomena of cryptomarket.

The paper is structured as follows. Section 2 describes previous studies in these fields. Section 3 presents classification of crashes and critical events on the Bitcoin market during the entire period (16.07.2010 – 08.12.2018). Network measures of complexity and their effectiveness as indicators of cryptomarket crashes are presented in Section 4. And finally, we discuss our results in Section 5.

2 Analysis of previous studies

Throughout the existence of Bitcoin, its complexity became much larger. Crashes and critical events that took place on this market as well as the reasons that led to them, did

not go unheeded. We determined that there are a lot of articles and papers on that topic which we will demonstrate.

Donier and Bouchaud [12] found that the market microstructure on Bitcoin exchanges can be used to anticipate illiquidity issues in the market, which lead to abrupt crashes. They investigate Bitcoin liquidity based on order book data and, out of this, accurately predict the size of price crashes.

Taking to the account studies on network analysis we can notice different papers on this topic [13-15]. Di Francesco Maesa et al. [13] have performed on the users' graph inferred from the Bitcoin blockchain, dumped in December 2015, so after the occurrence of the exponential explosion in the number of transactions. Researchers first present the analysis assessing classical graph properties like densification, distance analysis, degree distribution, clustering coefficient, and several centrality measures. Then, they analyze properties strictly tied to the nature of Bitcoin, like rich-get-richer property, which measures the concentration of richness in the network. Alexandre Bovet et al. [14] analyzed the evolution of the network of Bitcoin transactions among users and built network-based indicators of Bitcoin bubbles.

Authors [15] consider the history of Bitcoin and transactions in it. Using this dataset, they reconstruct the transaction network among users and analyze changes in the structure of the subgraph induced by the most active users. Their approach is based on the unsupervised identification of important features of the time variation of the network. Applying the widely used method of principal component analysis to the matrix constructed from snapshots of the network at different times, they show how changes in the network accompany significant changes in the price of Bitcoin.

Separately, it is necessary to highlight the work of Didier Sornette [16; 17], who built a precursor of crashes based on the generation of so-called log-periodic oscillations by the pre-crashing market. However, the actual collapse point is still badly predicted.

Thus, construction of indicators-precursors of critical and crash phenomena in the cryptocurrency market remains relevant.

3 Data

Bitcoin, despite its uncertain future, continues to attract investors, crypto-enthusiasts, and researchers. Being historically proven, popular and widely used cryptocurrency for the whole existence of cryptocurrencies in general, Bitcoin began to produce a lot of news and speculation, which began to determine its future life. Similar discussions began to lead to different kinds of crashes, critical events, and bubbles, which professional investors and inexperienced users began to fear. Thus, we advanced into action and set the tasks:

1. Classification of such bubbles, critical events and crashes.
2. Construction of such indicators that will predict crashes, critical events in order to give investors and ordinary users the opportunity to work in this market.

At the moment, there are various research works on what crises and crashes are and how to classify such interruptions in the market of cryptocurrencies. Taking into account the experience of previous researchers [18-21], we have created our classification of such leaps and falls, relying on Bitcoin time series during the entire period (16.07.2010 – 08.12.2018) of verifiable fixed daily values of the Bitcoin price (BTC) (<https://finance.yahoo.com/cryptocurrencies>).

For our classification, crashes are short, time-localized drops, with strong losing of price per each day, which are formed as a result of the bubble. Critical events are those falls that could go on for a long period of time, and at the same time, they were not caused by a bubble. The bubble is an increasing in the price of the cryptocurrencies that could be caused by certain speculative moments. Therefore, according to our classification of the event with number (1, 3-6, 9-11, 14, 15) are the crashes that are preceded by the bubbles, all the rest – critical events. More detailed information about crises, crashes and their classification in accordance with these definitions is given in the Table 1.

Table 1. BTC Historical Corrections. List of Bitcoin major corrections $\geq 20\%$

No.	Time	Days in correction	Bitcoin High Price, \$	Bitcoin Low Price, \$	Decline, %	Decline, \$
1	07.06.2011-10.06.2011	4	29.60	14.65	50	15.05
2	15.01.2012-16.02.2012	33	7.00	4.27	39	2.73
3	15.08.2012-18.08.2012	4	13.50	8.00	40	5.50
4	08.04.2013-15.04.2013	8	230.00	68.36	70	161.64
5	04.12.2013-18.12.2013	15	1237.66	540.97	56	696.69
6	05.02.2014-25.02.2014	21	904.52	135.77	85	768.75
7	12.11.2014-14.01.2015	64	432.02	164.91	62	267.11
8	11.07.2015-23.08.2015	44	310.44	211.42	32	99.02
9	09.11.2015-11.11.2015	3	380.22	304.70	20	75.52
10	18.06.2016-21.06.2016	4	761.03	590.55	22	170.48
11	04.01.2017-11.01.2017	8	1135.41	785.42	30	349.99
12	03.03.2017-24.03.2017	22	1283.30	939.70	27	343.60
13	10.06.2017-15.07.2017	36	2973.44	1914.08	36	1059.36
14	16.12.2017-22.12.2017	7	19345.5	13664.96	29	5680.53
15	13.11.2018-26.11.2018	14	6339.17	3784.59	40	2554.58

Accordingly, during this period in the Bitcoin market, many crashes and critical events shook it. Thus, considering them, we emphasize 15 periods on Bitcoin time series, whose falling we predict by our indicators, relying on normalized returns and volatility, where normalized returns are calculated as

$$g(t) = \ln X(t + \Delta t) - \ln X(t) \cong [X(t + \Delta t) - X(t)] / X(t),$$

and volatility as $V_T(t) = \frac{1}{n} \sum_{t'=t}^{t+n-1} |g(t')|$. Besides, considering that $g(t)$ should be more than the $\pm 3\sigma$, where σ is a mean square deviation.

Calculations were carried out within the framework of the algorithm of a moving window. For this purpose, the part of the time series (window), for which there were calculated measures of complexity, was selected, then the window was displaced along the time series in a one-day increment and the procedure repeated until all the studied series had exhausted. Further, comparing the dynamics of the actual time series and the corresponding measures of complexity, we can judge the characteristic changes in the dynamics of the behavior of complexity with changes in the cryptocurrency. If this or that measure of complexity behaves in a definite way for all periods of crashes, for example, decreases or increases during the pre-crashes period, then it can serve as an indicator or precursor of such a crashes phenomenon.

Calculations of measures of complexity were carried out both for the entire time series, and for a fragment of the time series localizing the crash. In the latter case, fragments of time series of the same length with fixed points of the onset of crashes or critical events were selected and the results of calculations of complexity measures were compared to verify the universality of the indicators.

In the Fig. 1 output Bitcoin time series, normalized returns $g(t)$, and volatility $V_T(t)$ calculated for the window size 100 are presented.

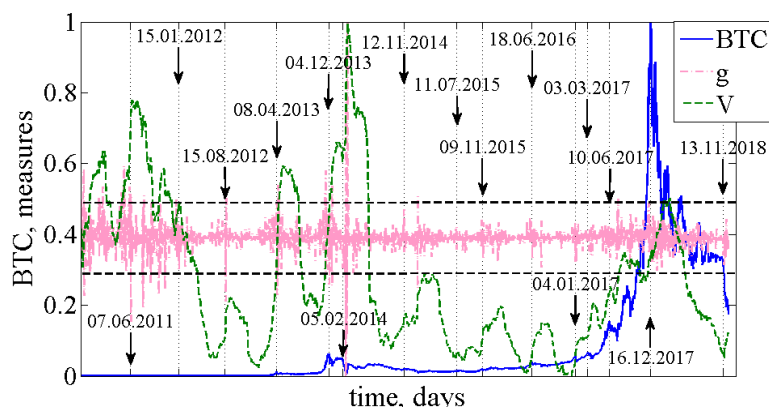


Fig. 1. The standardized dynamics, returns $g(t)$, and volatility $V_T(t)$ of BTC/USD daily values. Horizontal dotted lines indicate the $\pm 3\sigma$ borders. The arrows indicate the beginning of one of the crashes or the critical events

From Fig. 1 we can see that during periods of crashes and critical events normalized profitibility g increases considerably in some cases beyond the limits $\pm 3\sigma$. This indicates about deviation from the normal law of distribution, the presence of the “heavy tails” in the distribution g , characteristic of abnormal phenomena in the market. At the same time volatility also grows. These characteristics serve as indicators of critical and collapse phenomena as they react only at the moment of the above mentioned phenomena and don’t give an opportunity to identify the corresponding abnormal phenomena in advance. In contrast, the indicators described below respond to critical and collapse phenomena in advance. It enables them to be used as indicators-precursors of such phenomena and in order to prevent them.

4 Complex network indicators

The most commonly used methods for converting time sequences to the corresponding networks are recurrent [22], visibility graph [23] and correlation [24]. In the first case, the recurrence diagram is transformed into an adjacency matrix, on which the spectral and topological characteristics of the graph are calculated. The algorithm of the visibility graph is realized as follows. Take a time series $Y(t) = [y_1, y_2, \dots, y_n]$ of length N . Each point in the time series data can be considered as a vertex in an associated network, and the edge connects two vertices if two corresponding data points can ‘see’ each other from the corresponding point in the time series. Formally, two values of the series y_a (at the time of time t_a) and y_b (at the time of time t_b) are connected, if for any other value (y_c, t_c) , which is placed between them (i. e., $t_a < t_c < t_b$), the condition is satisfied:

$$y_c < y_a + (y_b - y_a) \frac{t_c - t_a}{t_b - t_a}$$

To construct and analyze the properties of a correlation graph, we must form a correlation matrix from the set of cryptocurrencies (as is done in Section 7), and from it we must pass to the matrix of adjacency. To do this, you must enter a value which, for the correlation field, will be the distance between the correlated assets. Such a distance may be dependent on the correlation coefficients c_{ij} of the value $x(i, j) = (2(1-c_{ij}))^{1/2}$. So, if the correlation coefficient between the two assets is significant, the distance between them is small, and, starting from some critical value, assets can be considered bound on the graph.

For constructed graph methods described above, one can calculate spectral and topological properties. We will show that some of them serve as a measure of the complexity of the system, and the dynamics of their changes allows us to build predictors of crashes or critical events in the financial markets.

Spectral theory of graphs is based on algebraic invariants of a graph – its spectra. The spectrum of graph G is the set of eigenvalues $S_p(G)$ of a matrix corresponding to a given graph. For adjacency matrix A of a graph, there exists a characteristic polynomial $|\lambda I - A|$, which is called the characteristic polynomial of a graph $P_G(\lambda)$. The eigenvalues of the matrix A (the zeros of the polynomial $|\lambda I - A|$) and the spectrum of the matrix A (the set of eigenvalues) are called respectively their eigenvalues λ and the spectrum $S_p(G)$ of graph G . The eigenvalues of the matrix A satisfy the equality $A\bar{x} = \lambda\bar{x}$ (\bar{x} – non-zero vector). Vectors \bar{x} satisfying this equality are called eigenvectors of the matrix A (or the graph G) corresponding to their eigenvalues.

From a multiplicity of spectral and topological measures we will choose only two – the maximum eigenvalue λ_{\max} of the adjacency matrix and Average path Length (ApLen). For a connected network of N nodes, the ApLen is equal

$$\langle l \rangle = \frac{2}{n(N-1)} \sum_{i>j} l_{ij},$$

where l_{ij} – the length of the shortest path between the nodes i and j .

Fig. 2 demonstrates the asymmetric response of the spectral and topological measures of network complexity. For the complete series, the calculation parameters are as follows: window width 100, step is 1 day. For local measures, the length of the fragment is 150, the width of the window is 50 and the step is 1 day.

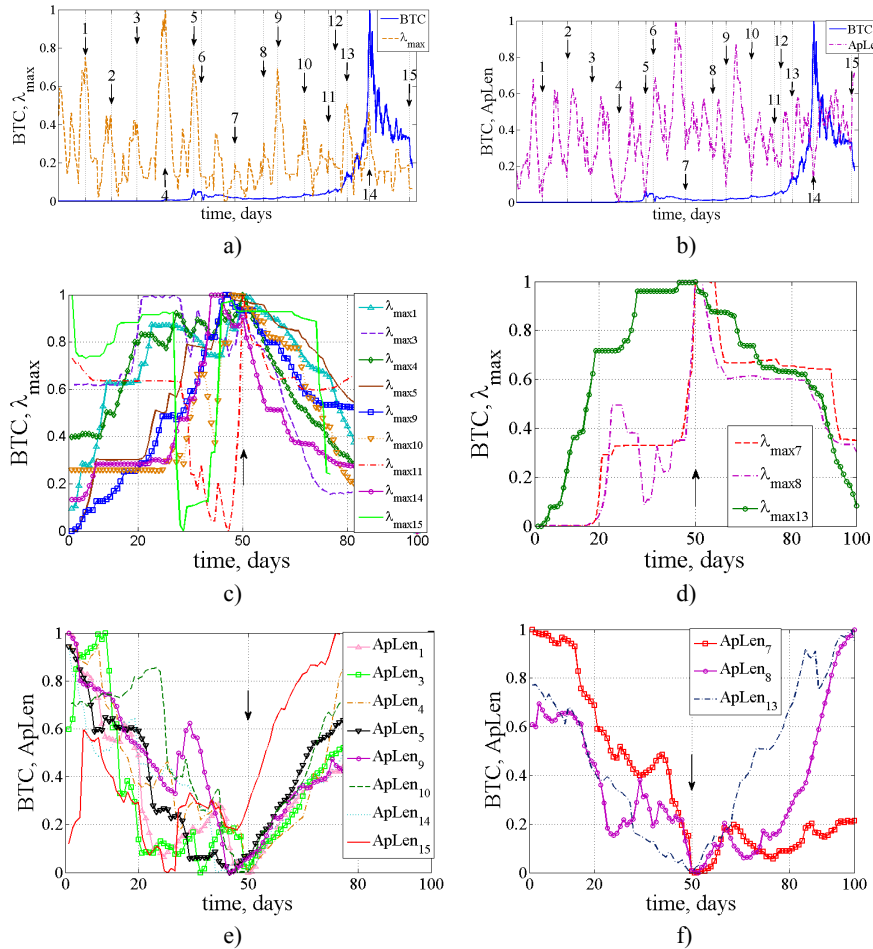


Fig. 2. Visibility graph dynamics of network measures λ_{\max} (a), ApLen (b) for all Bitcoin time series. Dynamics of network measures for local crashes (c, e) and crisis events (d, f)

The maximum actual value of the adjacency matrix of the visibility graph both for Bitcoin as a whole and for isolated segments of time series containing a crash and critical phenomenon, takes maximum value. It corresponds to the maximum complexity of the system. An especial state of the system leads to a decrease in complexity, and, accordingly, to a decrease in value λ_{\max} . Average length of the path on the graph (ApLen) is, on the contrary, minimal for complex systems and increases with the

randomization of the system. Such increase during pre-crash and pre-critical states as well as reduce λ_{\max} are indicators-precursors of the above mentioned states. You can choose other spectral and topological measures from the calculated ones, e.g. the maximum degree of the vertex and the diameter of the graph, algebraic connectivity and centrality, etc. Network measures of complexity, thus, are the most universal and informative and have obvious advantages in the selection of indicators of special states.

5 Conclusions

Consequently, in this paper, we have shown that monitoring and prediction of possible critical changes on cryptocurrency is of paramount importance. As it has been shown by us, the theory of complex networks has a powerful toolkit of methods and models for creating effective indicators-precursors of crashes and critical phenomena. In this paper, we have explored the possibility of using the network measures of complexity to detect dynamical changes in a complex time series. We have shown that the measures that have been used can indeed be effectively used to detect abnormal phenomena for the time series of Bitcoin.

References

1. Halvin, S., Cohen, R.: Complex networks. Structure, robustness and function. Cambridge University Press, New York (2010)
2. Albert, R., Barabási, A.-L.: Statistical Mechanics of Complex Networks. *Rev. Mod. Phys.* **74**, 47–97 (2002). doi:10.1103/RevModPhys.74.47
3. Newman, M., Barabási A.-L., Watts D.J.: The Structure and Dynamics of Networks. Princeton University Press, Princeton (2006)
4. Newman, M.E.J.: The Structure and Function of Complex Networks. *SIAM Reviews.* **45**(2), 167–256 (2003). doi:10.1137/S003614450342480
5. Nikolis, G., Prigogine, I.: Exploring Complexity: An Introduction. St. Martin's Press, New York (1989)
6. Andrews, B., Calder, M., Davis, R.A.: Maximum Likelihood Estimation for α -Stable Autoregressive Processes. *The Annals of Statistics.* **37**(4), 1946–1982 (2009). doi:10.1214/08-AOS632
7. Shah, D., Zhang, K.: Bayesian regression and Bitcoin. In: 2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, 30 Sept.-3 Oct. 2014. doi:10.1109/ALLERTON.2014.7028484
8. Chen, T., Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In: KDD '16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, August 13-17, 2016, pp. 785-794 (2016). doi:10.1145/2939672.2939785
9. Alessandretti, L., ElBahrawy, A., Aiello, L.M., Baronchelli, A.: Machine Learning the Cryptocurrency Market. <https://ssrn.com/abstract=3183792> (2018). doi:10.2139/ssrn.3183792. Accessed 15 Sep 2018
10. Guo, T., Antulov-Fantulin, N.: An experimental study of Bitcoin fluctuation using machine learning methods. arXiv:1802.04065v2 [stat.ML]. <https://arxiv.org/pdf/1802.04065.pdf> (2018). Accessed 15 Sep 2018

11. Peng Y., Albuquerque, P.H.M., de Sá, J.M.C., Padula, A.J.A., Montenegro, M.R.: The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with Support Vector Regression. *Expert Systems with Applications*. **97**, 177–192 (2018). doi:10.1016/j.eswa.2017.12.004
12. Donier, J., Bouchaud J.-P.: Why Do Markets Crash? Bitcoin Data Offers Unprecedented Insights. *PLoS ONE* **10**(10): e0139356 (2015). doi:10.1371/journal.pone.0139356
13. Di Francesco Maesa, D., Marino, A., Ricci, L.: Data-driven analysis of Bitcoin properties: exploiting the users graph. *International Journal of Data Science and Analytics*. **6**(1), 63–80 (2018). doi:10.1007/s41060-017-0074-x
14. Bovet, A., Campajola, C., Lazo, J.F., Mottes, F., Pozzana, I., Restocchi, V., Saggese, P., Vallarano, N., Squartini, T., Tessone, C.J.: Network-based indicators of Bitcoin bubbles. arXiv:1805.04460v1 [physics.soc-ph]. <https://arxiv.org/pdf/1805.04460> (2018). Accessed 11 Sep 2018
15. Kondor, D., Csabai, I., Szüle, J., Pósfai, M., Vattay, G.: Inferring the interplay of network structure and market effects in Bitcoin. *New Journal of Physics*. **16**, 125003 (2014). doi:10.1088/1367-2630/16/12/125003
16. Wheatley, S., Sornette, D., Huber, T., Reppen, M., Gantner, R.N.: Are Bitcoin Bubbles Predictable? Combining a Generalized Metcalfe’s Law and the LPPLS Model. *Swiss Finance Institute Research Paper No. 18-22*. <https://ssrn.com/abstract=3141050> (2018). doi:10.2139/ssrn.3141050. Accessed 15 Sep 2018
17. Gerlach, J.-C., Demos, G., Sornette, D.: Dissection of Bitcoin’s Multiscale Bubble History from January 2012 to February 2018. arXiv:1804.06261v2 [econ.EM]. <https://arxiv.org/pdf/1804.06261> (2018). Accessed 15 Sep 2018
18. Soloviev, V., Belinskij, A.: Methods of nonlinear dynamics and the construction of cryptocurrency crisis phenomena precursors. In: Ermolayev, V., Suárez-Figueroa, M.C., Yakovyna, V., Kharchenko, V., Kobets, V., Kravtsov, H., Peschanenko, V., Prytula, Y., Nikitchenko, M., Spivakovsky, A. (eds.) *Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops*, Kyiv, Ukraine, May 14-17, 2018. *CEUR Workshop Proceedings*. **2014**, 116–127. http://ceur-ws.org/Vol-2104/paper_175.pdf (2018). Accessed 30 Sep 2018
19. Casey, M.B.: Speculative Bitcoin Adoption/Price Theory. <https://medium.com/@mcasey0827/speculative-bitcoin-adoption-price-theory-2eed48ecf7da> (2016). Accessed 25 Sep 2018
20. McComb, K.: [2018] Bitcoin Crash: Analysis of 8 Historical Crashes and What’s Next. <https://blog.purse.io/bitcoin-crash-e112ee42c0b5> (2018). Accessed 25 Sep 2018
21. Amadeo, K.: Stock Market Corrections Versus Crashes and How to Protect Yourself: How You Can Tell If It’s a Correction or a Crash. <https://www.thebalance.com/stock-market-correction-3305863> (2018). Accessed 25 Sep 2018
22. Donner, R.V., Small, M., Donges, J.F., Marwan, N., Zou, Y., Xiang, R., Kurths, J.: Recurrence-based time series analysis by means of complex network methods. *International Journal of Bifurcation and Chaos*. **21**(04), 1019–1046 (2011). doi:10.1142/S0218127411029021
23. Lacasa, L., Luque, B., Ballesteros, F., Luque, J., Nuño, J.C.: From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences of the United States of America*. **105**(13), 4972–4975 (2008). doi:10.1073/pnas.0709247105
24. Burnie, A.: Exploring the Interconnectedness of Cryptocurrencies using Correlation Networks. arXiv:1806.06632 [q-fin.CP]. <https://arxiv.org/pdf/1806.06632> (2018). Accessed 25 Sep 2018

Модифікація аналітичного гамма-алгоритму пласкої укладки графа

Богдан Вікторович Гребенюк

Криворізький національний університет,
вул. Віталія Матусевича, 11, м. Кривий Ріг, 50027, Україна
bogdan020699@gmail.com

Анотація. Планарність графів – це один із ключових розділів теорії графів. Хоча граф є абстрактним математичним об'єктом, найчастіше саме візуалізація графа спрощує вивчення або розробку у певній сфері, наприклад, інфраструктури міста, менеджменту компанії або веб-сторінки сайту. Взагалі у вигляді графа можна зобразити будь-які структури, що мають зв'язки між елементами. Але часто подібні структури збільшуються до таких розмірів, що важко визначити, чи можливо представити їх на площині без перетину зв'язків. Існує багато алгоритмів, що вирішують це питання. Одним із таких є гамма-алгоритм. У статті визначені його проблеми та запропоновані методи їх вирішення, а також досліджені шляхи їх досягнення.

Ключові слова: дискретні структури, теорія графів, планарність.

Modification of the analytical gamma-algorithm for the flat layout of the graph

Bohdan V. Hrebeniuk

Kryvyi Rih National University, 11, Vitalii Matusevych St., Kryvyi Rih, 50027, Ukraine
bogdan020699@gmail.com

Abstract. The planarity of graphs is one of the key sections of graph theory. Although a graph is an abstract mathematical object, most often it is graph visualization that makes it easier to study or develop in a particular area, for example, the infrastructure of a city, a company's management or a website's web page. In general, in the form of a graph, it is possible to depict any structures that have connections between the elements. But often such structures grow to such dimensions that it is difficult to determine whether it is possible to represent them on a plane without intersecting the bonds. There are many algorithms that solve this issue. One of these is the gamma method. The article identifies its

problems and suggests methods for solving them, and also examines ways to achieve them.

Keywords: discrete structures, graph theory, planarity.

1 Вступ

Розвиток теорії графів в основному зобов'язаний великій кількості застосувань. Мабуть, з усіх математичних об'єктів графи займають найбільш чільне місце в якості формальних моделей реальних систем.

Графи знайшли застосування практично у всіх галузях наукових знань: фізиці, біології, хімії, математиці, історії, лінгвістиці, соціальних науках, техніці та ін. Найбільшою популярністю теоретико-графові моделі користуються при дослідженні комунікаційних мереж, систем інформатики, хімічних [1] і генетичних [2] структур, електричних ланцюгів [3] та інших систем мережевої структури.

Планарність графів – це один з ключових розділів теорії графів. Хоча граф є абстрактним математичним об'єктом, найчастіше саме візуалізація графа спрощує вивчення або розробку у певній сфері, наприклад, інфраструктури міста, менеджменту компанії або веб-сторінки сайту. Взагалі у вигляді графа можна зобразити будь-які структури, що мають зв'язки між елементами. Але часто подібні структури збільшуються до таких розмірів, що важко визначити, чи можливо представити їх на площині без перетину зв'язків.

Існує багато алгоритмів, які визначають планарність графа. Відома формула Ейлера [4] описує плаский граф наступним чином:

$$|V(G)| - |E(G)| + |F(G)| = 2,$$

де G – граф, $|V(G)|$ – кількість вершин, $|E(G)|$ – кількість ребер, $|F(G)|$ – кількість граней.

Ключовим словом тут є «визначення»: за допомогою формули визначається, чи можливо взагалі укласти граф. Призначення аналітичних алгоритмів – практична допомога в укладці графа.

На практиці аналітичні алгоритми являють собою послідовність перетворень графа, що призводять до його укладки. Одним з найпоширеніших алгоритмів є гамма-алгоритм [5]. Без сумніву, простота формулювання забезпечила йому популярність, адже невеликі розрахунки можна виконати вручну, а ключові моменти легко реалізувати на аркуші паперу. Однак, протилежним боком простоти стає обмеженість можливостей. Докладно всі проблеми оригінального гамма-алгоритму ми визначимо нижче, також ми запропонуємо ідеї та методи усунення цих проблем. Розглянемо покрокову реалізацію оригінального гамма-алгоритму.

Отже, на вхід алгоритму подається граф з наступними властивостями:

1. Граф зв'язний.
2. Граф містить хоча б один цикл.

3. Граф не має мостів.

Якщо порушена властивість 1, то граф потрібно укласти окремо за компонентами зв'язності. Якщо порушена властивість 2, то граф – це дерево, і зобразити його пласку укладку тривіально.

Докладніше розглянемо ситуацію, коли в графі G порушена властивість 3. У такому випадку спочатку потрібно видалити всі мости. Наступним кроком буде окрема укладка всіх компонент за схемою: укладемо одну компоненту зв'язності, а наступну компоненту, що пов'язана з першою мостом в графі G , малюватимемо в тій грані, де лежить вершина, що належить мосту. Інакше можлива ситуація, коли кінцева вершина моста буде знаходитися всередині плаского графа, а наступна компонента – зовні. Таким чином ми зможемо з'єднати мостом потрібні вершини. Надалі будемо використовувати цей метод для кожної нової компоненти.

Перший крок алгоритму – ініціалізація. Обираємо будь-який простий цикл в G , укладаємо його на площину і отримуємо дві грані: зовнішню і внутрішню. Уже укладену під час роботи алгоритму частину будемо позначати G_{plane} .

Другий крок алгоритму – спільний крок. Будується множина сегментів S . Кожен сегмент S відносно вже побудованого графа може бути одним з двох:

- ребро, обидва кінці якого належать G_{plane} , але саме воно йому не належить;
- зв'язна компонента G/G_{plane} , що доповнена усіма такими ребрами графа G , у яких один з кінців належить зв'язній компоненті, а другий належить графу G_{plane} .

Вершини, які одночасно належать G_{plane} і будь-якому сегменту, назвемо контактними вершинами.

Нехай грань Γ вміщує сегмент S , якщо номери всіх контактних вершин S належать цій грані, $S \subset \Gamma$. Вочевидь, таких граней може бути декілька. Множину таких граней позначимо $\Gamma(S)$, а їх кількість – $|\Gamma(S)|$.

Отже, розглянемо всі сегменти S_i і для кожного визначимо кількість $|\Gamma(S_i)|$. Якщо знайдеться такий номер i , для якого $|\Gamma(S_i)| = 0$, то граф є непланарним, і алгоритм завершує роботу. Інакше вибираємо такий сегмент S_i , для якого кількість $|\Gamma(S_i)|$ найменша. Якщо таких сегментів декілька, то можна вибрати будь-який з них.

У сегменті знайдемо довільний ланцюг між двома контактними вершинами і укладемо його в будь-яку з граней множини $\Gamma(S)$. При цьому дана грань розіб'ється на дві. Вже укладена частина графа G' після укладки ланцюга збільшиться, а сегмент, з якого виїнято ланцюг, зникне або розпадеться на менші з новими контактними вершинами, що ведуть до вершин G' .

Тепер необхідно повторювати укладку сегментів доти, поки ми не використаємо всі сегменти, або поки не буде отримана відповідь, що граф непланарний.

2 Проблеми традиційного гамма-алгоритму

Під час використання гамма-алгоритму було виявлено наступні недоліки:

1. Алгоритм не намагається мінімізувати перетини, а шукає лише «ідеальний» варіант.

Алгоритм, зіткнувшись із сегментом, який нікуди не вдається укласти, одразу припиняє свою роботу. Однак, практика показує, що часто сегменти не піддаються укладці через якийсь один ланцюг. Якщо такий ланцюг пропустити – визнати, що в цей момент ми не можемо його укласти, і просто відтермінувати рішення – граф можна буде укласти далі. Потрібно таку можливість реалізувати, оскільки здебільшого цей алгоритм нечасто використовується саме через цей недолік.

2. Алгоритм не вміє укладати граф різними конфігураціями.

В алгоритмі в якості сегмента використовується доволі «скута» модель – ціла компонента зв'язності. При цьому ланцюги, які ми знаходимо в компоненті, мають бути укладені лише в ті грані, в які дозволяє ця компонента, *хоча може існувати багато граней, що підходять для ланцюга, але не підходять для компоненти*. Граф, побудований за принципом оригінального алгоритму, матиме, в кращому випадку, одну конфігурацію, тоді як в реальній ситуації можуть існувати багато зауважень, що роблять поточну конфігурацію неприйнятною. Необхідна наявність можливості генерувати багато конфігурацій укладки графа.

3. Алгоритм не підтримує графи, що складаються з декількох компонент зв'язності та графи, які мають мости/точки зчленування.

У сучасному світі **час** є одним із пріоритетних ресурсів у розробці/вивченні певної сфери діяльності. Маючи масивну структуру з тисячі елементів, чи буде проектувальник ліній електропередач розбиратися, де мости, де точки зчленування, де компоненти зв'язності? Завдання – спроектувати ефективно, вигідно, у найкоротший термін. Навіть якщо розкласти граф на компоненти зв'язності, розбити мости, розділити точки зчленування і отримати множину укладених (а може і неукладених) компонент, як зібрати їх до купи? До речі, алгоритм дає хибну інформацію при роботі з такими графами, тобто доопрацювання його просто **необхідне**.

3 Запропоновані варіанти вирішення проблем

Щоб досягти мети, яку ми перед собою поставили, необхідно здійснити найбільш глобальну зміну. Як раніше було відзначено, представлення сегмента у вигляді компоненти зв'язності позбавляє алгоритм гнучкості. У сегменті можуть бути ланцюги, які неможливо укласти в деякі грані через те, що компонента забороняє

це робити. Звідси випливає висновок – компоненту потрібно скасувати, замінивши на щось інше. У запропонованій нами реалізації використано наступне рішення даної проблеми: **замінити компоненту зв'язності множиною всіх відсутніх ланцюгів**. При цьому ми застосовуємо однакові дії до наших «нових» і до «старих» сегментів (так само рахуємо кількість граней, куди ланцюг може бути укладений, все так же використовуємо ланцюг з мінімальною кількістю граней і т. ін.). Отже, з цією зміною ми матимемо:

1. Інформацію про кількість придатних для укладки граней для кожного ланцюга, тоді як компонента давала інформацію лише про спільні грані ланцюгів.
2. Множину унікальних конфігурацій графа шляхом перебору граней для укладки ланцюга.
3. Множину не унікальних конфігурацій графа шляхом перебору ланцюгів.
4. Можливість позначати ланцюг як «неможливий для укладки» для конкретної конфігурації і продовжувати алгоритм без нього.

За допомогою такої модифікації ми вирішили пункт 2 у нашому переліку проблем і частково пункт 1 – використовуючи ланцюги в ролі сегмента ми «розв'язали руки» нашого алгоритму. Тепер ми можемо укласти що хочемо і куди хочемо, а також спокійно відкладати сегменти, які не вдається укласти.

Що стосується пункту 3, то оригінальний алгоритм пропонує наступні дії з графом, у якому є мости: «... спочатку всі мости необхідно видалити, надалі виконати окрему укладку всіх компонент наступним чином: укладемо одну компоненту зв'язності, а наступну компоненту, пов'язану з першою в графі G мостом, будемо малювати в тій грані, в якій лежить вершина, що належить мосту».

Перш за все, необхідно зробити виправлення в умові: якщо є мости і **точки зчленування**. Незрозуміло, чому в алгоритмі немає ані слова про них, адже вони можуть привести алгоритм в таке саме невизначене становище, як і мости. Також необхідно дотримуватися певного порядку укладки компонент і слідувати певним правилам. Сформулюємо їх в наступному розділі.

4 Модифікований гамма-алгоритм

Отже, на вхід алгоритму подається будь-який граф. Після цього необхідно виконати наступні дії:

1. Знайти всі компоненти зв'язності.
2. У кожній з компонент знайти точки зчленування і мости.
3. Якщо такі є, розбити компоненту наступним чином (рис. 1).

На рис. 1 компонента $\{1, 2, 3, 4, 5\}$ має точку зчленування – вершину 3. Після поділу у нас вийшло дві компоненти, які поділяють спільну вершину. Те ж саме відбувається і з компонентою, яка має міст (рис. 2).

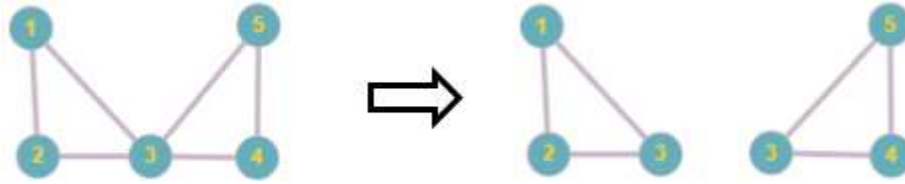


Рис. 1. Поділ по точці зчленування

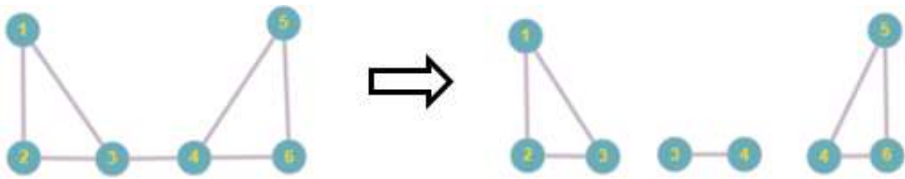


Рис. 2. Поділ по мосту

Також потрібно відзначити, що наступний граф розіб'ється в такий спосіб (рис. 3):



Рис. 3. Поділ компонентів

Хоча у даного графа є 3 точки зчленування – $\{3, 4, 5\}$, ми можемо їх об'єднати в дерево. Тобто фактично нам необхідно знайти всі подібні дерева в графі і «роз'єднати» компоненту.

4. Після того, як були знайдені всі компоненти зв'язності і проведено їх роз'єднання, необхідно укласти їх у певному порядку.

Ми пропонуємо наступний алгоритм укладання компонент.

- a. Укласти деяку компоненту зв'язності C .
- b. Серед компонент, що залишилися, визначити «зв'язані» (не плутати зі зв'язними компонентами), тобто ті, які мають спільні вершини з компонентою C .
- c. Якщо таких немає, повернутися до 1 пункту.
- d. Інакше кожену зі знайдених компонент необхідно укласти так, щоб спільна вершина мала доступ до зовнішньої грані. Після цього ми зможемо укласти побудовані нами компоненти у відповідні грані, де знаходяться спільні вершини.

- e. Для кожної зі знайдених компонент знов визначити неукладені компоненти, що мають спільні вершини. Повторювати пункти 4-5 доти, поки не вичерпаємо всі зв'язані компоненти.
- f. Якщо все зв'язані компоненти укладені, але не всі компоненти вичерпані, перейти до пункту 1.

Зупинимося на моменті вибору «початкової» компоненти C. Існує багато критеріїв, за якими можна оцінювати оптимальний варіант: це може бути компонента з найбільшою кількістю вершин, з найбільшою кількістю ребер, з найбільшою кількістю загальних вершин і т. ін. У загальному випадку краще вибирати найбільш «складну», наймасивнішу компоненту, тому що при першому укладанні не існує ніяких обмежень, отже, можна генерувати **всі можливі конфігурації** даної компоненти. Коли б така компонента укладалася не першою, ми могли б отримувати лише ті конфігурації, у яких загальна вершина має доступ до зовнішньої грані. У нашій реалізації в якості критерія ми будемо використовувати кількість вершин компоненти (дерева не братимуть участі у розрахунках). Але дуже легко обрати інший критерій пошуку, змінивши буквально один рядок коду.

- 5. Отже, перейдемо безпосередньо до укладки компоненти. Потрібно:
 - a. Ініціалізувати будь-який простий цикл в компоненті.
 - b. Якщо такого немає, то компонента – дерево, і укладка її не потребує особливих зусиль.
 - c. Інакше знаходимо всі відсутні ланцюги компоненти. Ви можете помітити, що деякі ребра повторюються в різних ланцюгах. При укладанні кожного ланцюга необхідно буде позбавлятися від уже укладених ребер. Маючи цикл і сегменти, можна почати генерацію конфігурацій. Генерувати ми будемо рекурсивно, тому для простоти пояснень за допомогою псевдокоду коротко, **поверхово** опишемо цю функцію.

```
function RecurSearch(cycles, segments)
  if segments.length == 0 then
    someComplexThings()
    return 1
  facesOfSegments ← getFacesOfSegments(cycles, segments)
  minSegment ← getMinSegment(facesOfSegments)
  if minSegment == 0 then
    anotherComplexThings()
    return 0
  For each segment in facesOfSegments with length == minSegment
  do
    For each face in facesOfSegments[segment] do
      Data ← getPreparedData(cycles, segments, segment, face)
      if RecurSearch(Data) != 0 do
        return 1
  return 0
```


де:

- **cycles** – відображення, що характеризує укладку, елементи якої утворюють пари «цикл»: «ланцюг»;
- **segments** – послідовність ще не укладених ланцюгів;
- **getFacesofSegments** – функція, яка повертає відображення, в якому елементи утворюють пари «ланцюг»: «послідовність циклів» (цикл у контексті графа);
- **getMinSegment** – функція, яка повертає число, яке є мінімальним з кількості граней, що підходять для укладки;
- **getPreparedData** – функція, що виконує укладку сегмента, розбиття грані і т. ін.;
- **someComplexThings**, **anotherComplexThings** – деякі дії, до яких ми ще не готові, але поки ми про них і не повинні замислюватися.

Насправді, навіть з таким поверхневим оглядом ми вже досить глибоко занурилися в деталі реалізації алгоритму, тому давайте словесно його опишемо. На кожному кроці рекурсії ми для кожного ланцюга знаходимо такі грані, куди ланцюг можна укласти. Знаходимо ланцюги, для яких кількість таких граней **однакова і мінімальна**. Тепер запускаємо рекурсивний пошук у кожному грані кожного ланцюга, перед цим розділивши грань, уклавши сегмент і т. ін. Тут також необхідно позбутися від уже укладених ребер, які ще можуть бути в ланцюгах.

Зверніть увагу, що при такому підході ми будемо отримувати результати, які з першого погляду можуть здатися однаковими. Відрізнятися вони будуть лише порядком укладки сегментів у грані, що теж іноді буває важливим. Щоб генерувати тільки унікальні конфігурації, необхідно запускати рекурсивний пошук у кожному грані тільки для одного «мінімального» сегмента.

Якщо є деякі ланцюги, що неможливо укласти в жодну грань, потрібно пропустити їх. Фактично це ми і будемо робити в функції **anotherComplexThings**, але іншим шляхом, про який ми поговоримо вже у самій реалізації. Тобто при кожному виклику рекурсії ланцюгів ставатиме все менше і менше, і в кінцевому результаті алгоритм їх повністю вичерпає.

Як тільки будуть вичерпані всі сегменти (а це означає, що алгоритм здійснив деяку укладку графа), необхідно перевірити, чи відповідає укладена компонента умовам, які зазначені в пункті 4. Для цього призначена функція **someComplexThings**. Якщо дана конфігурація пройшла всі перевірки – компонента успішно укладена.

Повний опис алгоритму та програмна реалізація мовою Python подані у [6].

Список використаних джерел

1. Семеріков С. О. Застосування методів машинного навчання у навчанні моделювання майбутніх учителів хімії / С. О. Семеріков // Технології навчання хімії у школі та ЗВО : збірник тез доповідей Всеукраїнської науково-практичної Інтернет-конференції / заг. ред. Т. В. Старова (вид. 1-е). – Кривий Ріг : КДПУ, 2018. – С. 10-19.

2. Komarova O. V. Computer Simulation of Biological Processes at the High School [Electronic resource] / Olena V. Komarova, Albert A. Azaryan // Augmented Reality in Education : Proceedings of the 1st International Workshop (AREdu 2018). Kryvyi Rih, Ukraine, October 2, 2018 / Edited by : Arnold E. Kiv, Vladimir N. Soloviev. – P. 24-32. – (CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2257). – Access mode : <http://ceur-ws.org/Vol-2257/paper03.pdf>
3. Modlo Ye. O. Modernization of Professional Training of Electromechanics Bachelors: ICT-based Competence Approach [Electronic resource] / Yevhenii O. Modlo, Serhiy O. Semerikov, Ekaterina O. Shmeltzer // Augmented Reality in Education : Proceedings of the 1st International Workshop (AREdu 2018). Kryvyi Rih, Ukraine, October 2, 2018 / Edited by : Arnold E. Kiv, Vladimir N. Soloviev. – P. 148-172. – (CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2257). – Access mode : <http://ceur-ws.org/Vol-2257/paper15.pdf>
4. Euler L. Solutio problematis ad geometriam situs pertinentis / Euler L. // Commentarii Acad. Sci. Imp. Petrop., 8, 1736. – P. 128-140.
5. Ирине́в А. Алгоритм плоской укладки графов [Электронный ресурс] / Ирине́в Антон, Каширин Виктор. – [2006]. – 10 с. – Режим доступа : <http://rain.ifmo.ru/cat/data/theory/graph-coloring-layout/layout-2006/article.pdf>.
6. Гребенюк Б. В. gamma [Электронный ресурс] / [Б. В. Гребенюк]. – 2018. – Режим доступа : <https://github.com/BogdanGrebenuk/gamma>.

References (translated and transliterated)

1. Semerikov, S.O.: Zastosuvannia metodiv mashynnoho navchannia u navchanni modeliuvannia maibutnikh uchyteliv khimii (Application of machine learning methods in teaching simulation of future chemistry teachers). In: Starova, T.V. (ed.) Technologies of teaching chemistry in school and university, All-Ukrainian scientific and practical Internet conference, Kryvyi Rih, November 30, 2018, pp. 10–19. KDPU, Kryvyi Rih (2018)
2. Komarova, O.V., Azaryan, A.A.: Computer Simulation of Biological Processes at the High School. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 24–32. <http://ceur-ws.org/Vol-2257/paper03.pdf> (2018). Accessed 30 Nov 2018
3. Modlo, Ye.O., Semerikov, S.O., Shmeltzer, E.O.: Modernization of Professional Training of Electromechanics Bachelors: ICT-based Competence Approach. . In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 148–172. <http://ceur-ws.org/Vol-2257/paper15.pdf> (2018). Accessed 30 Nov 2018
4. Euler, L.: Solutio problematis ad geometriam situs pertinentis. Commentarii Acad. Sci. Imp. Petrop., **8**, 128–140 (1736)
5. Irinev, A., Kashirin, V.: Algoritm ploskoi ukladki grafov (Algorithm for flat styling graphs). <http://rain.ifmo.ru/cat/data/theory/graph-coloring-layout/layout-2006/article.pdf> (2006). Accessed 1 Nov 2018
6. Hrebenuk, B.V.: gamma. <https://github.com/BogdanGrebenuk/gamma> (2018). Accessed 17 Nov 2018

Використання генератора випадкових чисел з апаратним джерелом ентропії для задач симетричної криптографії

Маргарита Геннадіївна Долотій,
Павло Володимирович Мерзликін^[0000-0002-0752-411X]

Криворізький державний педагогічний університет,
пр. Гагаріна, 54, м. Кривий Ріг, 50086, Україна
mdolotiy@gmail.com, linuxoid@i.ua

Анотація. Метою дослідження є перевірка можливості використання розробленого генератора випадкових чисел [1], який використовує як джерело ентропії шуми звукової карти, в алгоритмах симетричної криптографії.

Ключові слова: генератор випадкових чисел, джерело ентропії, статистичний критерій, закон розподілу, симетрична криптографія.

Using the random number generator with a hardware entropy source for symmetric cryptography problems

Marharyta H. Dolotii, Pavlo V. Merzlykin^[0000-0002-0752-411X]

Kryvyi Rih State Pedagogical University, 54, Gagarin Ave., Kryvyi Rih, 50086, Ukraine
mdolotiy@gmail.com, linuxoid@i.ua

Abstract. The aim of the research is to test the possibility of using the developed random number generator [1], which utilizes the sound card noise as an entropy source, in the symmetric cryptography algorithms.

Keywords: random number generator, source of entropy, test statistic, probability distribution, symmetric cryptography.

1 Вступ

Генератори випадкових чисел найчастіше застосовують в криптографії, адже випадковість і криптографія дуже сильно взаємопов'язані. Складно знайти коректно розроблене криптографічне прикладне забезпечення, яке не

використовує випадкові числа. Вектори ініціалізації, модифікатори геш-функцій, унікальні параметри при роботі з цифровими підписами повинні прийматися випадковими [2]. При використанні генераторів випадкових чисел в криптографічних системах генератори випадкових чисел повинні відповідати наступним вимогам [3]:

- послідовність, що генерується повинна мати максимально великий період;
- послідовність, що генерується, не повинна мати прихованих періодичностей;
- послідовність, що генерується, повинна мати рівномірний спектр.

Для демонстрації статистичних властивостей генераторів випадкових чисел використовуються різні підходи до статистичного тестування. Найчастіше набір і методику тестування пропонував сам розробник генератора. Таким чином, склалася ситуація, яка характеризується тим, що неможливо було об'єктивно порівняти різні генератори з єдиних позицій. Щоб подолати дану ситуацію, необхідно використовувати деякий стандартний набір статистичних тестів, об'єднаних єдиною методикою розрахунку необхідних показників ефективності генератора й прийняття рішення про випадковість генерованих послідовностей.

2 Методологія

У 1999 році фахівцями NIST в рамках проекту AES (Advanced Encryption Standard) був розроблений набір статистичних тестів NIST Statistical Test Suite), а також запропонована методика проведення статистичного тестування генераторів випадкових чисел [4], які на даний момент найкраще відповідають потребам всіх зацікавлених сторін.

У даній роботі розглядаються критерії прийняття рішення про проходження послідовністю статистичного тесту, набір статистичних тестів NIST і наводяться результати експериментальних досліджень властивостей ГВЧ, описаного в роботі [1].

Порядок тестування окремої двійкової послідовності S виглядає наступним чином.

1. Висувається нульова гіпотезу H_0 – припускаємо, що дана двійкова послідовність S випадкова.
2. За послідовністю S розраховується статистику тесту $c(S)$.
3. З використанням спеціальної функції та статистики тесту розраховується значення ймовірності $P=f(c(S))$, $P \in [0; 1]$.
4. Значення ймовірності P порівнюється з рівнем значущості α , $\alpha \in [0,001; 0,01]$. Якщо $P \geq \alpha$, то гіпотеза H_0 приймається. В іншому випадку приймається альтернативна гіпотеза.

Обраний пакет статистичних тестів може використовуватися для вирішення наступних завдань:

- виявлення ГВЧ (ГПВЧ), які формують «погані» двійкові послідовності;

- розробка нових ГВЧ (ГПВЧ);
- перевірка коректності реалізації ГВЧ (ГПВЧ);
- вивчення генераторів, описаних в стандартах;
- дослідження ступеня випадковості реально використовуваних ГВЧ (ГПВЧ).

3 Обговорення результатів

Для перевірки було генеровано три випадкові послідовності в діапазоні [0; 255] розміром 10 000 елементів.

Пакет NIST STS було завантажено з офіційного сайту Національного інституту стандартів і технологій (National Institute of Standards and Technology) [5], також для інтерпретації отриманих результатів використовувалась офіційна інструкція із сайту [6].

Перед тим, як було протестовано кожен з трьох послідовностей, був побудований загальний графік розподілу для послідовностей, показаний на рис. 1. По осі x відкладено значення байту, а по осі y кількість байтів з таким значенням у послідовності.

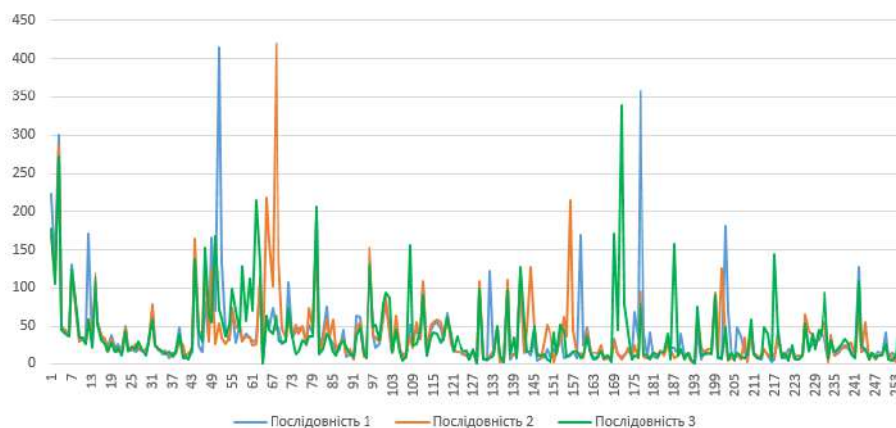


Рис. 1. Графік розподілів трьох генерованих послідовностей

З рис. 1 видно, що кожна із послідовностей має свої піки та особливості, тому вони підходять для перевірки пакетом NIST STS. Першим тестом, проведеним над генерованими послідовностями, був Binary Matrix Rank Test. Основну увагу в даному тесті приділяють рангу непересічної підматриці всієї послідовності. Метою цього тесту є перевірити лінійну залежність між підрядками фіксованої довжини оригінальної послідовності. Варто зауважити, що цей тест також використовуються в серії тестів DIEHARD.

Інтерпретація цього тесту: великі значення χ^2 (obs) вказують на те, що відхилення рангового розподілу від того, що відповідає випадковій послідовності, є значимим. В результаті проходження даного тесту результатом, за

яким можна зазначати, що послідовність випадкова, є значення p . Якщо обчислене $p < 0,01$, то вважається, що послідовність не випадкова. Інакше послідовність є випадковою.

Результати проходження тесту Binary Matrix Rank Test для першої послідовності показані на рис. 2.

Для даної послідовності значення $p=0,203766$, що свідчить про випадковість генерованої послідовності. Також значення $\chi^2=3,181562$ є малим, тобто відхилення рангового розподілу від того, що відповідає випадковій послідовності, є незначним.

```

                                RANK TEST
-----
                                COMPUTATIONAL INFORMATION:
-----
(a) Probability P_32 = 0.288788
(b)              P_31 = 0.577576
(c)              P_30 = 0.133636
(d) Frequency   F_32 = 5
(e)              F_31 = 3
(f)              F_30 = 1
(g) # of matrices = 9
(h) Chi^2       = 3.181562
(i) NOTE: 784 BITS WERE DISCARDED.
-----
SUCCESS          p_value = 0.203766

```

Рис. 2. Binary Matrix Rank Test для першої послідовності

Для послідовностей 2 і 3 значення $p=0,374306$ та $p=0,648387$, що також вказує на їх випадковість. Відповідні значення χ^2 : 1,965364 та 0,866535 є досить малими, що знову свідчить про те, що відхилення рангового розподілу від того, що відповідає випадковій послідовності, є незначним.

При цьому варто відмітити, що третя генерована послідовність з отриманих результатів є більш «випадковою» через більше значення p та менше значення χ^2 у порівнянні із послідовностями 1 та 2.

Наступний тест, що використовувався для перевірки випадковості генерованих послідовностей, це Non-overlapping Template Matching Test. Основною для цього тесту є кількість випадків попередньо заданих цільових рядків. Мета цього тесту – виявлення генераторів, що виробляють забагато випадків даної аперіодичної моделі. Для цього тесту використовується m -бітне вікно, що шукає конкретний m -бітний шаблон. Якщо шаблон не знайдено, вікно пропускає одну бітну позицію. Якщо шаблон знайдено, вікно пропускає біт після знайденого шаблону, і пошук відновлюється.

Цей тест відкидає послідовності, що демонструють занадто багато або занадто мало випадків даної аперіодичної картини.

Тест можна інтерпретувати як відхилення послідовностей, що виявляють нерегулярні входження даної неперіодичної картини.

Перша генерована послідовність за тестом Non-overlapping Template Matching Test дала результуючий файл, показаний на рис. 3.

NONPERIODIC TEMPLATES TEST												
COMPUTATIONAL INFORMATION												
LAMBDA = 2.425781			M = 1250			N = 8 m = 9 n = 10000						
Template	F R E Q U E N C Y							Chi^2	P_value	Assignment	Index	
	W_1	W_2	W_3	W_4	W_5	W_6	W_7					W_8
000000001	0	0	0	0	0	0	0	0	19.944262	0.010549	SUCCESS	0
000000011	0	0	0	0	0	0	0	0	19.944262	0.010549	SUCCESS	1
000000101	0	0	0	0	0	0	0	0	19.944262	0.010549	SUCCESS	2
000000111	0	0	0	0	0	0	0	0	19.944262	0.010549	SUCCESS	3
000001001	0	0	0	0	0	0	0	0	19.944262	0.010549	SUCCESS	4
000001011	0	0	0	0	0	0	0	0	19.944262	0.010549	SUCCESS	5

Рис. 3. Non-overlapping Template Matching Test для першої послідовності

Для кожного із 148-ми індексів значення $p=0,010549$, що дає підставу програмі видавати результат Assignment = SUCCESS (успіх), що означає успішне проходження тесту та підтверджує «випадковість» генерованої послідовності.

Для послідовності 2 і 3 значення p є аналогічним, як і помітки про успішне проходження даними послідовностями тесту Binary Matrix Rank.

Наступний тест, що використовувався для перевірки випадковості генерованих послідовностей, це Overlapping Template Matching Test. Основна увага тесту приділяється кількості випадків попередньо заданих цільових рядків. Даний тест використовує вікно m -біт для пошуку конкретного m -бітового шаблону. Якщо шаблон не знайдено, вікно пропускає бітну позицію. Коли шаблон знайдено, вікно пропускає лише один біт, перш ніж відновити пошук.

Цей тест відкидає послідовності, які показують занадто багато або дуже мало випадків m -пробілів, але може бути легко модифікований для виявлення нерегулярних випадків будь-якого періодичного малюнка.

Для першої генерованої послідовності відповідний результат продемонстровано на рис. 4:

OVERLAPPING TEMPLATE OF ALL ONES TEST										
COMPUTATIONAL INFORMATION:										
(a) n (sequence length)	= 10000									
(b) m (block length of 1s)	= 9									
(c) M (length of substring)	= 1032									
(d) N (number of substrings)	= 9									
(e) lambda [(M-m+1)/2^m]	= 2.000000									
(f) eta	= 1.000000									
F R E Q U E N C Y										
0	1	2	3	4	>=5	Chi^2	P-value	Assignment		
8	0	0	0	0	1	11.119950	0.049053	SUCCESS		

Рис. 4. Overlapping Template Matching Test першої послідовності

З отриманих результатів ми бачимо значення $p=0,049053$ та значення $\chi^2=11,119950$, що відповідають за успішне проходження генерованої послідовності даного тесту. Для другої та третьої послідовності значення p та χ^2 є аналогічними, що підтверджує їх випадковість.

Наступний тест, яким було перевірено генеровані послідовності, був Serial Test. Основна увага цього тесту сконцентрована на частотах всіх можливих перекриваючих m -бітних шаблонів по всій послідовності. Мета цього тесту полягає в тому, щоб визначити, чи є число входжень 2^m m -біт приблизно таким же, як було б очікувано для випадкової послідовності. Випадкові послідовності однорідні, тобто кожен m -бітний шаблон має такий же шанс з'явитись, як і будь-який інший m -розрядний шаблон. Варто звернути увагу, що для $m=1$ послідовний тест еквівалентний частотному тесту.

Серійний тест (генералізований) – це перелік процедур, що базується на тестуванні однорідності розподілу моделей заданих довжин.

У результаті проходження тесту першою послідовністю було отримано два значення p , та відповідні значення ψ (рис. 5).

```

                                SERIAL TEST
                                -----
                                COMPUTATIONAL INFORMATION:
                                -----
(a) Block length (m) = 16
(b) Sequence length (n) = 10000
(c) Psi_m = -10000.000000
(d) Psi_m-1 = -10000.000000
(e) Psi_m-2 = -10000.000000
(f) Del_1 = 0.000000
(g) Del_2 = 0.000000
                                -----
SUCCESS p_value1 = 1.000000
SUCCESS p_value2 = 1.000000

```

Рис. 5. Serial Test першої послідовності

З отриманих результатів можна зробити висновок, що послідовність є випадковою за Serial Test. Щодо послідовностей 2 і 3, то вони мають аналогічні показники значень, що також підтверджує випадковість розробленого генератора випадкових чисел.

Останнім тестом, що проходять генеровані послідовності, є Linear Complexity Test. Основна увага цього тесту приділена довжині регістру зсуву з лінійним зворотним зв'язком (LFSR). Метою цього тесту є визначення, чи є послідовність досить складною, щоб вважатись випадковою. Випадкові послідовності характеризуються більш тривалими LFSR. Якщо LFSR занадто короткий, то послідовність вважається не випадковою.

Для першої генерованої послідовності в результаті виконання даного тесту було отримано результати, показані на рис. 6.

```

                                L I N E A R C O M P L E X I T Y
                                -----
M (substring length) = 500
N (number of substrings) = 20
                                -----
                                F R E Q U E N C Y
                                -----
C0 C1 C2 C3 C4 C5 C6 CHI2 P-value
                                -----
Note: 0 bits were discarded!
0 0 4 7 5 4 0 9.101060 0.167974

```

Рис. 6. Linear Complexity Test першої послідовності

За отриманими результатами $p=0,167974$ та $\chi^2=9,101060$ можна зробити висновок, що генерована послідовність є випадковою. Щодо інших послідовностей: для другої послідовності $p=0,783217$ та $\chi^2=3,201098$; для третьої $p=0,167974$ та $\chi^2=9,101060$. Тобто, значення першого і третього розподілу збігається, в той же час, значення другого розподілу відрізняється, проте, всі три послідовності є випадковими.

4 Висновки

1. Найкращі результати було продемонстровано в тестах Binary Matrix Rank та Linear Complexity, де кожна із послідовностей мала власні значення p та χ^2 , що давало можливість порівняти роботу генератора випадкових чисел в однакових умовах, проте, при різних генерованих послідовностях. Не зважаючи на відмінні результати, всі три послідовності успішно пройшли тест. Загалом створений апаратний генератор випадкових чисел, згідно результатів тестування пакетом NIST STS, можна вважати таким, що задовольняє вимоги пройдених тестів.
2. Створена бібліотека генерації випадкових чисел може бути використана в проектах, які мають потребу в високоякісних послідовностях випадкових чисел. В майбутньому генератор випадкових чисел може бути вдосконалений за рахунок використання більш чистого звуку та реалізації можливості вибору користувачем розподілу, за яким генеруватимуться випадкові числа.

Список використаних джерел

1. Долотій М. Г. Генератор випадкових чисел з апаратним джерелом ентропії / Маргарита Геннадіївна Долотій, Павло Володимирович Мерзликін // Новітні комп'ютерні технології. – Кривий Ріг : Видавничий центр ДВНЗ «Криворізький національний університет», 2017. – Том XV. – С. 85-88.
2. Kelsey J. Cryptanalytic Attacks on Pseudorandom Number Generators / John Kelsey, Bruce Schneier, David Wagner, Chris Hall // Fast Software Encryption. 5th International Workshop, FSE'98. Paris, France, March 23-25, 1998 : Proceedings / Ed. : Serge Vaudenay. –Berlin, Heidelberg : Springer-Verlag, 1998. – P. 168-188. – (Lecture Notes in Computer Science, vol. 1372)
3. Долгих А. О. Генератор псевдо випадкових чисел [Електронний ресурс] / А. О. Долгих // Тези Всеукраїнської науково-практичної on-line конференції аспірантів, молодих учених та студентів, присвяченої Дню науки. 10-12 травня 2017 року // Конференції Житомирського державного технологічного університету : матеріали конференцій, проведених в ЖДТУ. – 2017. – Режим доступу : <https://conf.ztu.edu.ua/wp-content/uploads/2017/06/123-1.pdf>.
4. Soto J. Randomness Testing of the Advanced Encryption Standard Candidate Algorithms [Electronic resource] / Juan Soto, Jr. – Gaithersburg : National Institute of Standards and Technology, September 1999. – 10 p. – Access mode : <https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir6390.pdf>. – (NISTIR 6390)

5. NIST SP 800-22: Documentation and Software - Random Bit Generation | CSRC [Electronic resource]. – 2018. – Access mode : <https://csrc.nist.gov/Projects/Random-Bit-Generation/Documentation-and-Software>.
6. Bassham L. E. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications [Electronic resource] / Lawrence E. Bassham III, Andrew L. Rukhin, Juan Soto, James R. Nechvatal, Miles E. Smid, Elaine Barker, Stefan D. Leigh, Mark Levenson, Mark Vangel, David L. Banks, Alan Heckert, James Dray, San Vo. – Gaithersburg : National Institute of Standards and Technology, September 16, 2010. – 131 p. – Access mode : https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=906762. – (Special Publication (NIST SP) - 800-22 Rev 1a)

References (translated and transliterated)

1. Dolotii, M.G., Merzlykin, P.V.: Henerator vypadkovykh chysel z aparatnym dzherelom entropii (The random number generator with hardware source of entropy). *New computer technology*. **15**, 85–88 (2017).
2. Kelsey, J., Schneier, B., Wagner, D., Hall, C.: Cryptanalytic Attacks on Pseudorandom Number Generators. In: Vaudenay, S. (ed.) *Fast Software Encryption, 5th International Workshop, FSE'98*. Paris, France, March 23-25, 1998. *Lecture Notes in Computer Science*, vol. 1372, pp. 168–188. Springer-Verlag, Berlin, Heidelberg (1998)
3. Dolhykh, A.O.: Henerator psevo vypadkovykh chysel (Generator of pseudorandom numbers). In: *The theses of the All-Ukrainian scientific and practical on-line conference of postgraduates, young scientists and students devoted to the Day of Science, ZhDTU, Zhytomyr*, 10–12 May 2017. <https://conf.ztu.edu.ua/wp-content/uploads/2017/06/123-1.pdf> (2017). Accessed 12 Nov 2018
4. Soto, J.Jr.: *Randomness Testing of the Advanced Encryption Standard Candidate Algorithms*. National Institute of Standards and Technology, Gaithersburg, September 1999. (NISTIR 6390). <https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir6390.pdf> (1999). Accessed 30 Nov 2018
5. NIST SP 800-22: Documentation and Software - Random Bit Generation | CSRC. <https://csrc.nist.gov/Projects/Random-Bit-Generation/Documentation-and-Software> (2018)
6. Bassham III, L.E., Rukhin, A.L., Soto, J., Nechvatal, J.R., Smid, M.E., Barker, E., Leigh, S.D., Levenson, M., Vangel, M., Banks, D.L., Heckert, A., Dray, J., Vo, S.: *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. National Institute of Standards and Technology, Gaithersburg, September 16, 2010 (Special Publication (NIST SP) - 800-22 Rev 1a). https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=906762 (2010). Accessed 30 Nov 2018

Актуальність дослідження програм семантичного аналізу текстів та огляд методів їх реалізації

Аркадій Анатолійович Зеленський

Криворізький національний університет,
вул. Віталія Матусевича, 11, м. Кривий Ріг, 50027, Україна
arkasha9606@gmail.com

Анотація. Одним із найголовніших завдань прикладної лінгвістики є розв'язання проблеми якісної автоматизованої обробки природної мови. Найпопулярнішими методами обробки природномовних текстових відповідей з метою екстракції та репрезентації семантики мають бути системи, які ґрунтуються на ефективному поєднанні лінгвістичних технологій аналізу та методів аналізу. Серед існуючих методів аналізу текстових даних чинне місце посідає метод з використанням векторної моделі. Іншим ефективним і актуальним засобом вилучення семантики із тексту та її представлення є метод латентно-семантичного аналізу (ЛСА). Метод ЛСА пройшов випробування та підтвердив свою ефективність у таких напрямках обробки природної мови, як моделювання концептуальних знань людини; інформаційний пошук, при реалізації якого ЛСА показує набагато кращі результати порівняно із звичайними векторними методами.

Ключові слова: семантика, прикладна лінгвістика, обробка текстів.

Relevance of research of programs for semantic analysis of texts and review of methods of their realization

Arkadii A. Zelenskyi

Kryvyi Rih National University, 11, Vitalii Matusevych St., Kryvyi Rih, 50027, Ukraine
arkasha9606@gmail.com

Abstract. One of the main tasks of applied linguistics is the solution of the problem of high-quality automated processing of natural language. The most popular methods for processing natural-language text responses for the purpose of extraction and representation of semantics should be systems that are based on the efficient combination of linguistic analysis technologies and analysis methods. Among the existing methods for analyzing text data, a valid method is

used by the method using a vector model. Another effective and relevant means of extracting semantics from the text and its representation is the method of latent semantic analysis (LSA). The LSA method was tested and confirmed its effectiveness in such areas of processing the native language as modeling the conceptual knowledge of the person; information search, the implementation of which LSA shows much better results than conventional vector methods.

Keywords: semantics, applied linguistics, text processing.

Мова – це найважливіший засіб спілкування і пізнання, мова є продовженням думки людини. Це засіб спілкування, який поки що властивий тільки людям.

Програмні продукти ще в другій половині – кінці ХХ століття почали проникати в широкі маси. Здебільшого це були розважальні програми та відеоігри. А з розвитком інформаційних технологій та телекомунікацій життя становиться все більш та більш мобільним та інформативним [1; 2]. Нові технології міцно входять в різні сфери життя та вносять в них нові стандарти.

Проблема механізмів синтаксису, що раніше відносилася виключно до «чистої науки», стала за останні роки «гостро прикладною». Такі пріоритетні області науки і техніки, як штучний інтелект [3; 4], машинний переклад, створення системи спілкування «людина – Інтернет» тощо потребують від учених максимально конкретних визначень механізмів їх дії на життєвий процес людини.

Інтернет подарував людству надзвичайно великі можливості у спілкуванні, навчанні та праці. Число користувачів мережі Інтернет зростає неймовірними кроками.

Люди щоденно генерують великі обсяги інформації в Інтернет: вони обмінюються повідомленнями, залишають відгуки, оформлюють замовлення, виконують пошук, надсилають листи та ін. Нещодавно для українців стала доступна можливість подачі замовлення на оформлення громадянського паспорту онлайн. 93 % населення України є активними Інтернет-користувачами станом на 30 червня 2017 року.

Компанія Facebook опублікувала звіт, з якого стало відомо, що протягом вересня 2018 року соціальну мережу відвідувало близько 1,49 мільярда активних користувачів щоденно, та загальна кількість користувачів за місяць становила 2,27 мільярда.

Зараз багато компаній розширює свої можливості за рахунок різних ботів, які дозволяють здійснювати найрізноманітніші операції – купувати каву, замовляти піцу, вести моніторинг залізничних квитків, а потім і придбати їх. Ботів лише почали активно впроваджувати в наше життя, в майбутньому з'явиться ще більше ботів різного призначення, і тому питання обробки тексту зараз є актуальним, як ніколи.

Прикладна лінгвістика – розділ мовознавства, який практично розв'язує питання, пов'язані з вивченням мови. З давніх давен люди вели розмову між собою, але завдяки аналізу текстів в людини може з'явитися новий співрозмовник.

Мова є продовженням думки людини. Розвиток в сфері аналізу тексту дозволить здійснити значний прорив у розробці людино-машинної взаємодії. Використовуючи звичний для людини діалог, вона зможе оперувати складними комп'ютерними операціями. Окрім цього, система, яка буде здатна приймати голосові запити користувача у вигляді звичайного тексту та адекватно на них реагувати, відкриє нові можливості в освоєнні комп'ютерних технологій для людей з обмеженими можливостями.

У ХХІ столітті змінився механізм передачі і прийняття інформації. Мова, що стала штучною, є наразі основою засобів Інтернет-спілкування. Її лексика насичує інформаційний простір, в цій мові не дотримано ні відповідності правилам, ні допустимого відсотку в тексті слів іншомовного походження, тому проблема семантичного аналізу Інтернет-повідомлень є достатньо складною задачею.

Предметом вивчення семантики є значна кількість мовних об'єктів. До них належить, наприклад, семантична структура слова, семантика мови і мовлення (висловлення), семантика сприймання висловлення, семантика граматичної системи мови та ін.

Прикладна лінгвістика використовується для вирішення низки актуальних питань сучасності таких, як інтелектуальний помічник, що спілкується з людиною, відповідає на питання та виконує команди; аналіз повідомлень, що визначає важливість повідомлення або виявляє інтереси користувача, як частина рекомендаційної системи; аналіз психологічного типу особистості, що ґрунтується на текстах, що набирає користувач та їх емоційному забарвленню; визначення області, до якої відноситься текст, а також пропонування доречної відповіді на проаналізований текст. Наступним етапом прикладної лінгвістики може стати робота з аудіо форматом тексту, де є можливість визначати інтонації та на їх основі виявляти важливі думки в тексті.

Лінгвістична обробка природномовних текстів є однією з центральних проблем інтелектуалізації інформаційних технологій, якій приділяється значна увага в розвинутих країнах Європи, США та Японії, свідченням чого є великі капіталовкладення у розроблення лінгвістичного програмного забезпечення та цілісних лінгвістичних технологій. Велику кількість науково-дослідних програм спрямовано на розвиток лінгвістичних інформаційних систем. У зв'язку з бурхливим розвитком Інтернет ця проблема набуває ще більшої значущості.

Аналіз одиниць лексичного рівня, тобто слів, є одним з найважливіших. До складу цього різновиду аналізу можуть входити різні словники, які містять лексичний репертуар та морфемну структуру лексичних одиниць, словозмінні парадигми тощо. Застосовують і так звані безсловникові методи, які передбачають здійснення розбиття слова на морфеми за заданими алгоритмами, забезпечення віднесення слів до словозмінних парадигматичних класів, виконання процедури лематизації (редукції текстових слів до початкових, вихідних форм тощо). На цьому ж етапі частково здійснюється і граматичний аналіз, тобто ідентифікація лексико-граматичних класів та значень граматичних категорій текстових слів. Зрозуміло, що для української мови надзвичайно

важливою є наявність словозмінної парадигми та відповідної словозмінної класифікації.

Роботу в цій сфері можна розподілити на декілька основних етапів. Першим буде очищення вхідного тексту від слів, які не несуть змістовного навантаження, та виокремлення основи слова для більш гнучкого їх порівняння.

Наступним етапом буде виділення ключових слів вхідного тексту. Третім етапом може стати виявлення області, до якої відноситься текст.

Одним з найголовніших завдань прикладної лінгвістики є розв'язання проблеми якісної автоматизованої обробки природної мови. Вирішення такої проблеми може використовуватись для автоматизації перевірки відповіді студента, поданої у текстовому форматі природною мовою. Обробка природної мови буде основою розробки ефективної методики порівняння відповіді студента зі зразками правильної відповіді на запитання.

На думку А. В. Анісімова [5; 6], О. Є. Литвиненка [7] та ін., вищезгадані етапи лінгвістичного аналізу природномовних текстів людиною не є ізольованими один від одного, вони є паралельними процедурами сприйняття змісту тексту. Отже й комп'ютерні моделі семантичного, синтаксичного і морфологічного аналізу природномовних текстів доцільно розробляти як систему паралельних розподілених процесів з заданим відношенням підпорядкування.

Найпопулярнішими методами обробки природномовних текстових відповідей з метою екстракції та репрезентації семантики мають бути системи, які ґрунтуються на ефективному поєднанні лінгвістичних технологій аналізу (морфологічного, синтаксичного, семантичного) та методів аналізу, які дозволили б виявляти приховані асоціативні залежності всередині природномовних текстів.

Серед існуючих методів аналізу текстових даних чинне місце посідає метод з використанням векторної моделі. Основна ідея якого полягає в алгебраїчному представленні колекції текстових масивів векторами одного спільного для всієї колекції векторного простору. Векторна модель є основою для вирішення багатьох завдань, таких як: пошук документа за запитом, класифікація документів та кластеризація документів.

Іншим ефективним і актуальним засобом вилучення семантики із тексту та її подання є метод латентно-семантичного аналізу (ЛСА) [8].

Д. В. Ланде розглянув метод ЛСА в рамках технологій глибинного аналізу текстової інформації Text Mining, що розроблені на основі статистичного та лінгвістичного аналізу, і дозволяють не лише здійснювати відбір релевантних документів, а й виділяти їх семантику, яка досить часто буває прихованою [9].

Метод латентно-семантичного аналізу є теорією і методом екстракції і представлення контекстно-залежного змісту слів шляхом статистичної обробки великого корпусу текстів. Головною ідеєю методу є те, що сукупність усіх контекстів, в яких певне слово вживається або, навпаки, не вживається, обумовлює набір обмежень, які визначають подібність значень слів або множини слів. Отже, простежується думка, що між словами і контекстом, в якому вони вживаються, існують приховані (латентні) зв'язки. Метод ЛСА дозволяє

визначити асоціативну і семантичну близькість та вирахувати кореляції між двома термами, двома документами, або між термом і документом.

Ефективність застосування методу ЛСА в сфері знань людини підтверджена різноманітними прикладами його роботи. Зокрема, вперше зазначений метод був застосований з метою автоматичного індексування текстів та виявлення їх асоціативно-семантичної структури. Використання методу ЛСА знайшло своє відображення у системах вилучення, представлення семантичної інформації з тексту.

Метод ЛСА пройшов випробування та підтвердив свою ефективність у таких напрямках обробки природної мови як моделювання концептуальних знань людини; інформаційний пошук, при реалізації якого ЛСА показує набагато кращі результати порівняно зі звичайними векторними методами; процес підбору синонімів, якість якого була перевірена і підтверджена шляхом обробки тестів TOEFL; результат роботи методу ЛСА виявився ідентичним середнім показникам відбору синонімів людьми із неангломовних країн.

За результатами аналізу роботи методу ЛСА для вирішення вищезгаданих та інших завдань, можемо зробити висновок, що розглянутий метод є найкращим засобом для виявлення та представлення прихованих семантичних характеристик окремих слів та текстів загалом.

Отже, доходимо висновку, що найпопулярнішими методами обробки природномовних текстових відповідей мають бути системи, що ґрунтуються на ефективному поєднанні лінгвістичних технологій аналізу та методу ЛСА, що дозволить виявляти приховані асоціативні залежності всередині природномовних текстів. Таке поєднання дає можливість врахувати та частково ліквідувати недоліки, що притаманні обома методам, та удосконалити процес обробки природномовної відповіді шляхом комбінування переваг розглянутих лінгвістичних та статистичних методів.

Список використаних джерел

1. Семеріков С. О. Мобільне навчання: історія, теорія, методика / Сергій Семеріков, Ілля Теплицький, Світлана Шокалюк // Інформатика та інформаційні технології в навчальних закладах. – 2008. – № 6. – С. 72-82 ; 2009. – № 1. – С. 96-104.
2. Семеріков С. О. Мобільне навчання : історико-технологічний вимір / Семеріков С. О., Стрюк М. І., Моїсеєнко Н. В. // Теорія і практика організації самостійної роботи студентів вищих навчальних закладів : монографія / кол. авторів; за ред. проф. О. А. Коновала. – Кривий Ріг : Книжкове видавництво Киреєвського, 2012. – С. 188-242.
3. Semerikov S. O. Computer Simulation of Neural Networks Using Spreadsheets: The Dawn of the Age of Camelot [Electronic resource] / Serhiy O. Semerikov, Illia O. Teplytskyi, Yuliia V. Yechkalo, Arnold E. Kiv // Augmented Reality in Education : Proceedings of the 1st International Workshop (AREdu 2018). Kryvyi Rih, Ukraine, October 2, 2018 / Edited by : Arnold E. Kiv, Vladimir N. Soloviev. – P. 122-147. – (CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2257). – Access mode : <http://ceur-ws.org/Vol-2257/paper14.pdf>.
4. Markova O. M. CoCalc as a Learning Tool for Neural Network Simulation in the Special Course “Foundations of Mathematic Informatics” [Electronic resource] / Oksana Markova,

Serhiy Semerikov, Maiia Popel // ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer 2018 : Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops (ICTERI, 2018). Kyiv, Ukraine, May 14-17, 2018 / Edited by : Vadim Ermolayev, Mari Carmen Suárez-Figueroa, Vitaliy Yakovyna, Vyacheslav Kharchenko, Vitaliy Kobets, Hennadiy Kravtsov, Vladimir Peschanenko, Yaroslav Prytula, Mykola Nikitchenko, Aleksander Spivakovsky. – P. 388-403. – (CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2104). – Access mode : http://ceur-ws.org/Vol-2104/paper_204.pdf.

5. Анисимов А. В. Система обработки текстов на естественном языке / А. В. Анисимов, А. А. Марченко // Искусственный интеллект. – 2002. – № 4. – С. 157-163.
6. Анисимов А. В. Компьютерная лингвистика для всех: Мифы. Алгоритмы. Язык / А. В. Анисимов. – К. : Наук. думка, 1991. – 208 с.
7. Литвиненко О. Є. Інженерно-лінгвістичні принципи аналізу текстів / О. Є. Литвиненко, Д. А. Бурко // Наукоємні технології. – 2009. – Том 3, № 3. – С. 60-62. – DOI : 10.18372/2310-5461.3.5130
8. Заболева-Зотова А. В. Латентный семантический анализ: новые решения в Internet / А. В. Заболева-Зотова, А. Ю. Пастухов, П. В. Сердюков, Н. А. Козлова, С. А. Чернов // Информационные технологии. – 2001. – № 6. – С. 67-82.
9. Ландэ Д. В. Поиск знаний в Internet / Д. В. Ландэ. – М. : Вильямс, 2005. – 272 с. – (Профессиональная работа)

References (translated and transliterated)

1. Semerikov, S., Teplytskyi, I., Shokaliuk, S.: Mobilne navchannia: istoriia, teoriia, metodyka (Mobile learning: history, theory, methods). *Informatyka ta informatsiini tekhnologii v navchalnykh zakladakh*. 6, 72–82 (2008), 1, 96–104 (2009)
2. Semerikov, S.O., Striuk, M.I., Moiseienko, N.V. Mobilne navchannia: istoriko-tekhnolohichniy vymir (Mobile learning: historical and technological dimension). In: Konoval, O.A. (ed.) *Teoriia i praktyka orhanizatsii samostiinoi roboty studentiv vyshchykh navchalnykh zakladiv*, pp. 188-242. Knyzhkove vydavnytstvo Kyreievskoho, Kryvyi Rih (2012)
3. Semerikov, S.O., Teplytskyi, I.O., Yechkalo, Yu.V., Kiv, A.E.: Computer Simulation of Neural Networks Using Spreadsheets: The Dawn of the Age of Camelot. In: Kiv, A.E., Soloviev, V.N. (eds.) *Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018)*, Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 122–147. <http://ceur-ws.org/Vol-2257/paper14.pdf>. Accessed 30 Nov 2018
4. Markova, O., Semerikov, S., Popel, M.: CoCalc as a Learning Tool for Neural Network Simulation in the Special Course “Foundations of Mathematic Informatics”. In: Ermolayev, V., Suárez-Figueroa, M.C., Yakovyna, V., Kharchenko, V., Kobets, V., Kravtsov, H., Peschanenko, V., Prytula, Y., Nikitchenko, M., Spivakovsky, A. (eds.) *Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops (ICTERI, 2018)*, Kyiv, Ukraine, May 14-17, 2018. CEUR Workshop Proceedings, vol. 2104, pp. 388–403. http://ceur-ws.org/Vol-2104/paper_204.pdf. Accessed 25 Aug 2018
5. Anisimov, A.V., Marchenko, A.A.: Sistema obrabotki tekstov na estestvennom iazyke (Natural Language Processing System). *Iskusstvennyi intellekt*. 4, 157–163 (2002).

6. Anisimov, A.V.: *Kompiuternaia lingvistika dlia vseh: Mify. Algoritmy. Iazyk* (Computational linguistics for all: Myths. Algorithms. Language). Naukova dumka, Kiev (1991)
7. Lytvynenko, O.Ye., Burko, D.A.: *Inzhenerno-lingvistychni pryntsypy analizu tekstiv* (Engineering-linguistic principles of the analysis of texts). *Naukoiemni tekhnolohii*. **3**(3), 60–62 (2009). doi:10.18372/2310-5461.3.5130
8. Zabolieva-Zotova, A.V., Pastukhov, A.Iu., Serdiukov, P.V., Kozlova, N.A., Chernov, S. A.: *Latentnyi semanticheskii analiz: novye resheniia v Internet* (Latent semantic analysis: new solutions in the Internet). *Informatcionnye tekhnologii*. **6**, 67–82 (2001)
9. Lande, D.V.: *Poisk znani v Internet* (Search for knowledge on the Internet). Viliams, Moscow (2005)

Актуальність проблеми параметричної ідентифікації математичної моделі

Данило Іванович Куропятник

Криворізький національний університет,
вул. Віталія Матусевича, 11, м. Кривий Ріг, 50027, Україна
kuropyatnik27@gmail.com

Анотація. Метою статті є дослідження можливостей підвищення ефективності математичної моделі за рахунок ідентифікації параметрів об'єкта. Ключовим фактором для параметризації можна назвати врахування властивостей значень моделі в конкретний момент часу, що дозволяє глибше проаналізувати залежності даних та кореляцію між ними. Однак подібна методика працює не завжди, адже заздалегідь неможливо передбачити, що параметри можуть суттєво оптимізуватись. Окрім того, потрібно враховувати той факт, що мінімізація зменшує значення параметрів без урахування їх реальних фізичних властивостей. Правильність підсумкових значень буде опиратись на динамічно дібрані параметри, що дозволяє видозмінювати умови користування системою в режимі реального часу. У процесі розробки порівнюються значення експериментально отриманих даних з модельними, що і дозволяє зрозуміти точність мінімізації. При виборі найбільш релевантних параметрів використовуються різні мінімізаційні функції, що надає можливість охопити широкий спектр теоретичних початкових ситуацій. Перевірка правильності рішення здійснюється за допомогою функціоналу якості, за якого можна виявити точність та коректність оптимізованих параметрів. Можливо обрати різні типи функціоналу якості, залежно від особливостей початкових даних. Наявність подібних інструментів під час параметризації дозволяє різноманітно аналізувати модель, тестуючи її на різних алгоритмах, об'ємах даних та умовах гарантованої збіжності методів функціоналу.

Ключові слова: математична модель, машинне навчання, оптимізація, параметризація, функціонал якості.

Actuality of the problem of parametric identification of a mathematical model

Danylo I. Kuropiatnyk

Kryvyi Rih National University, 11, Vitalii Matusevych St., Kryvyi Rih, 50027, Ukraine
kuropiatnik27@gmail.com

Abstract. The purpose of the article is to study the possibilities of increasing the efficiency of a mathematical model by identifying the parameters of an object. A key factor for parametrization can be called the consideration of properties of the values of the model at a specific time point, which allows a deeper analysis of data dependencies and correlation between them. However, such a technique does not always work, because in advance it is impossible to predict that the parameters can be substantially optimized. In addition, it is necessary to take into account the fact that minimization reduces the values of parameters without taking into account their real physical properties. The correctness of the final values will be based on dynamically selected parameters, which allows you to modify the terms of use of the system in real time. In the development process, the values of experimentally obtained data with the model are compared, which allows you to understand the accuracy of minimization. When choosing the most relevant parameters, various minimization functions are used, which provides an opportunity to cover a wide range of theoretical initial situations. Verification of the correctness of the decision is carried out with the help of a quality function, which can identify the accuracy and correctness of the optimized parameters. It is possible to choose different types of functional quality, depending on the characteristics of the initial data. The presence of such tools during parametrization allows for varied analysis of the model, testing it on various algorithms, data volumes and conditions of guaranteed convergence of functional methods.

Keywords: mathematical model, machine learning, optimization, parametrization, quality functional.

На сучасному етапі виробництва та конструювання складних динамічних об'єктів неможливо обійтись без попередньої розробки їх математичної моделі [1]. Як правило, подібні об'єкти мають або лабораторно отримані значення, котрі не завжди є найефективнішим рішенням проблеми, або навіть така інформація про них відсутня. В такому випадку реалізація обраних процесів є досить збитковою з точок зору продуктивності та ресурсоемності через те, що дані про систему зібрані надто поверхово та недостатньо глибоко проаналізовані. Підходів рішення такої проблеми є декілька, однак замість детального вивчення властивостей об'єкта, на виробництвах все ще віддають перевагу надмірному вживанню ресурсів та емпіричному способу вимірювань. Яскравими прикладами

подібної системи можуть послужити такі моделі: печі-котла для спалювання сірководневого газу, авіаційного двигуна, рух та прогнозування його зміни у суден та інші.

Усі ці моделі об'єднує те, що в окремий момент часу відомі системі параметри можуть по різному впливати на підсумковий результат. Урахування даного фактору дозволяє проаналізувати властивості об'єкта, а також виявити закономірності змін його параметрів. Така методика дозволяє ідентифікувати рівень впливу різноманітних чинників в конкретні періоди функціонування системи, завдяки чому підвищується якість змодельованого процесу та практична ефективність подальшої фізичної реалізації.

Під час моделювання системи можливо виділити два типи інформації про об'єкт: апіорна та апостеріорна [1]. Стосовно першої можна сказати, що це фізичні властивості та закони, котрі доречні у розробці, а також теоретично виявлені умови коректного результату роботи системи. Другий тип, в свою чергу, відповідає за експериментально отримані результати за спостережень за параметрами об'єкта в процесі експлуатації. Завданням в такому випадку буде використання заздалегідь набутих та отриманих у ході роботи даних для того, щоб динамічно ідентифікувати якомога менші за значенням параметри, не втрачаючи при цьому правильності функціонування [2].

Важливо зазначити, що неможливо вказати методики, які для будь-якого об'єкта будуть однозначно гарантувати рішення задачі ідентифікації оптимальним способом, адже подібні завдання вимагають детального вивчення властивостей системи і мають умову, що в підсумку серед набору параметрів є простір для оптимізації. Також математичне моделювання не здатне в достатній мірі описати реальний процес, адже методи ідентифікації припускають знаходження параметрів без прив'язки до їх фізичного сенсу, а лише заради задоволення критерію адекватності моделі. Ці фактори критично необхідно враховувати при розробці системи параметричної ідентифікації, щоб в подальшому процесі її використання був практичний зміст [3].

Початковими умовами для розробки подібної системи є, очевидно, сконструйована математична модель досліджуваного об'єкта [4]. Іншими словами, на етапі мінімізації значень параметрів потрібно точно знати, як повинна функціонувати система, які підсумкові результати вона має повертати та яким є формат початкових даних. Також доцільно користуватися результатами апостеріорних досліджень, щоб оцінювати точність проведення оптимізації. Окрім цього, потрібно визначитись зі способом оцінки правильності отриманих значень – тобто, обрати найбільш актуальний метод оптимізації та функціонал якості.

Сам процес ідентифікації можна розглянути в декілька етапів [4]:

- знаходження значення моделі для початкових параметрів системи на проміжку заздалегідь обраного часу;
- порівняння експериментально отриманих та нових даних, знаходження нових параметрів;

— мінімізація нових параметрів із обраних даних в кожному конкретний момент часу.

Кожен із цих етапів є циклічним, тобто нові параметри можуть видозмінюватися від ітерації до ітерації. Оскільки завдання складається із знаходження нелінійних параметрів моделі, то основною задачею є багатовимірна мінімізація. Для її вирішення потрібно знайти точки мінімуму функції якомога точніше та швидше. Саме тут використовують методи оптимізації функціоналу якості.

Як правило, в актуальних моделях параметризації вживають метод градієнтного спуску, однак це не є єдиним і, тим паче, завжди найкращим рішенням. Причин цьому чимало, зокрема необхідність в диференційованості функції (цей метод використовує градієнт функції, що знаходиться за допомогою часткових похідних), а також відсутність гарантії збіжності за умови великої кількості локальних мінімумів. Також потрібно враховувати обсяг даних та обраний критерій точності обчислень.

Серед можливих методів доцільними також можна розглядати генетичний алгоритм, метод імітації відпалу та метод Нелдера-Міда [6]. Кожен з них має власну структуру та послідовність дій, через це вони покривають різні випадки та ситуації при моделюванні. Наприклад, метод диференціальної еволюції (генетичний алгоритм) має вищу ймовірність збіжності, ніж градієнтний, однак може використовувати для цього завелику кількість ресурсів і все ще не гарантує точний результат. Метод імітації відпалу, в свою чергу, завжди здатний знайти глобальний мінімум, але має найбільший час виконання. Щодо методу Нелдера-Міда, то він є дуже швидким, точним, однак не має однозначних гарантій збіжності. Усі ці методи мають перевагу над градієнтним спуском в тому, що здатні оброблювати недиференційовані функції.

Окрім вибору методу мінімізації, також є можливість перевіряти результати ідентифікації за різними критеріями точності. Для цього використовується функціонал якості – функція, або їх набір, що показує те, наскільки точно знаходить значення системи обраний алгоритм. Серед найрозповсюдженіших функціоналів для задачі параметричної ідентифікації виділяють наступні:

- *середньоквадратичний критерій* – розходження теоретичних та експериментальних значень, піднесене до другої степені. Він є нерозривним, що дозволяє використовувати метод градієнтного спуску, однак кожен показник (навіть несуттєвий, або випадковий) сильно впливає на значення;
- *мінімаксна функція* – різниця експериментальних та модельних даних за модулем. Вона менше реагує на «шуми», однак є розривною, тому недоступна для диференційованих методів;
- *функція-нев'язка* – різниця між значеннями функції та значеннями ознак із параметрами. Вона дозволяє за лінійності функцій моделі звести все до задачі лінійної ідентифікації, однак не є актуальною у випадку багатовимірної мінімізації.

Після добору алгоритму мінімізації та функціоналу якості настає етап реалізації параметричної ідентифікації. Важливо зазначити, що параметри будуть

змінюватися для кожного окремо взятого моменту часу, тобто в конструйованому об'єкті має бути можливість фізично безперервно впливати на його значення. Маючи такі дані стосовно досліджуваної математичної моделі, можна досить точно проаналізувати закономірності параметрів, враховуючи обсяги даних об'єкта та умови функціонування. Іншими словами, перед вибором тих чи інших алгоритмів, потрібно точно усвідомлювати пріоритети розробки: швидкість та ефективність виконання, чи гарантована точність та правильність результату.

Список використаних джерел

1. Полищук А. П. Автоматика : учебное пособие / А. П. Полищук, С. А. Семериков. – Кривой Рог : Издательский отдел КГПИ, 1999. – 277 с.
2. Полищук А. П. Некоторые особенности программной реализации методов экспериментальной идентификации линейных процессов / А. П. Полищук, С. А. Семериков // Комп'ютерне моделювання та інформаційні технології в науці, економіці та освіті : збірник наукових праць : в 2-х томах. – Кривий Ріг : Видавничий відділ КДПУ, 2001. – Т. 1. – С. 202-210.
3. Полищук А. П. Последовательный симплекс-поиск в задачах параметрической идентификации / А. П. Полищук, С. А. Семериков // Комп'ютерне моделювання та інформаційні технології в природничих науках. – Кривий Ріг : Видавничий відділ КДПУ, 2000. – С. 125-136.
4. Соловйов В. М. Инструментальне забезпечення курсу комп'ютерного моделювання / В. М. Соловйов, С. О. Семеріков, І. О. Теплицький // Комп'ютер у школі і сім'ї. – 2000. – № 4. – С. 28-31.
5. Миронов Е. А. Решение задачи идентификации математической модели объекта прогнозирования в условиях неопределенности [Электронный ресурс] / Миронов Е. А., Платонов С. А. // Современные проблемы науки и образования. – 2014. – № 4. – Режим доступа : <https://www.science-education.ru/ru/article/view?id=14029>.
6. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных : учебник / Петер Флах ; пер. с англ. Слинкин А. А. – М. : ДМК Пресс, 2015. – 408 с.

References (translated and transliterated)

1. Polishchuk, A.P., Semerikov, S.A.: Avtomatika (Automatics). Izdatelskii otdel KGPI, Krivoi Rog (1999)
2. Polishchuk, A.P., Semerikov, S.A.: Nekotorye osobennosti programmnoi realizatsii metodov eksperimentalnoi identifikatsii lineinykh protsessov (Some features of the software implementation of methods for the experimental identification of linear processes). In: Kompiuterne modeliuвання ta informatsiini tekhnolohii v nauksi, ekonomitsi ta osviti, vol. 1, pp. 202–210. Vydavnychiy viddil KDPU, Kryvyi Rih (2001)
3. Polishchuk, A.P., Semerikov, S.A.: Posledovatelnyi simpleks-poisk v zadachakh parametricheskoi identifikatsii (Sequential simplex search in parametric identification

- problems). In: *Kompiuterne modeliuvannia ta informatsiini tekhnolohii v pryrodnychukh naukakh*, pp. 125–136. Vydavnychiy viddil KDPU, Kryvyi Rih (2000)
4. Soloviov, V.M., Semerikov, S.O., Teplytskyi, I.O.: *Instrumentalne zabezpechennia kursu kompiuternoho modeliuvannia (Toolware of computer simulation course)*. *Kompiuter u shkoli i simi*. 4, 28–31 (2000)
 5. Mironov, E.A., Platonov, S.A.: *Reshenie zadachi identifikatsii matematicheskoi modeli obekta prognozirovaniia v usloviakh neopredelennosti (The solution to the problem of identifying a mathematical model of the object of forecasting in conditions of uncertainty)*. *Sovremennye problemy nauki i obrazovaniia*. 4. <https://www.science-education.ru/ru/article/view?id=14029> (2014). Accessed 12 Nov 2018
 6. Flach, P.: *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge University Press, New York (2012)

Квантове програмування – перспективний напрямок розвитку ІТ

Людмила Володимирівна Легка, Світлана Вікторівна Шокалюк^[0000-0003-3774-1729]

Криворізький державний педагогічний університет,
пр. Гагаріна, 54, м. Кривий Ріг, 50086, Україна
asp18-lehka@kdpu.edu.ua, shokalyuk15@gmail.com

Анотація. Метою даного дослідження є обґрунтування перспективності використання квантових комп'ютерів на потребу сучасного суспільства. Так, у статті наведені тлумачення базових понять квантової механіки, відмічені основні недоліки та певні переваги використання квантових комп'ютерів для надшвидкого розв'язання суспільно значущих задач. Окрім того, наведено узагальнені відомості про успішні спроби у розробках квантових комп'ютерів, зокрема, компаніями IBM, Intel, Google, Microsoft, та відкритості хмарного доступу до них. У якості подальших напрямів дослідження розглядається розробка методики навчання квантовому програмуванню учнів старшої школи.

Ключові слова: квантовий комп'ютер, квантові обчислення, квантове програмування, кубіт.

Quantum programming is a promising direction of IT development

Liudmyla V. Lehka and Svitlana V. Shokaliuk^[0000-0003-3774-1729]

Krivyri Rih State Pedagogical University, 54, Gagarin Ave., Kryvyi Rih, 50086, Ukraine
asp18-lehka@kdpu.edu.ua, shokalyuk15@gmail.com

Abstract. The purpose of this study is to substantiate the prospects of using quantum computers for the needs of modern society. So, in the article is given the interpretation of the basic concepts of quantum mechanics, are noted the main disadvantages and certain advantages of the using of quantum computers for ultrafast solutions of public important problems. In addition, is given the general information about the successful attempts to the development of quantum computers, in particular, by IBM, Intel, Google, Microsoft, and the openness of cloud access to them. As a further direction of research is considered the

development of a methodology of quantum programming trainings for high-school students.

Keywords: quantum computer, quantum computing, quantum programming, qubit.

Комп'ютерна техніка для сучасного суспільства є невід'ємною складовою науково-технічного прогресу. Впровадження новітніх технологій завжди супроводжується використанням більш продуктивних та ефективних комп'ютерів, еволюції яких пов'язують із «законом Мура» (рис. 1). Ще у 1965 році один із засновків компанії Intel Гордон Мур поділився виведеною закономірністю щодо темпів розвитку напівпровідникової індустрії – щільність транзисторів (їх кількість на квадратний дюйм в інтегральних схемах) буде подвоюватися кожні 18 місяців [8].

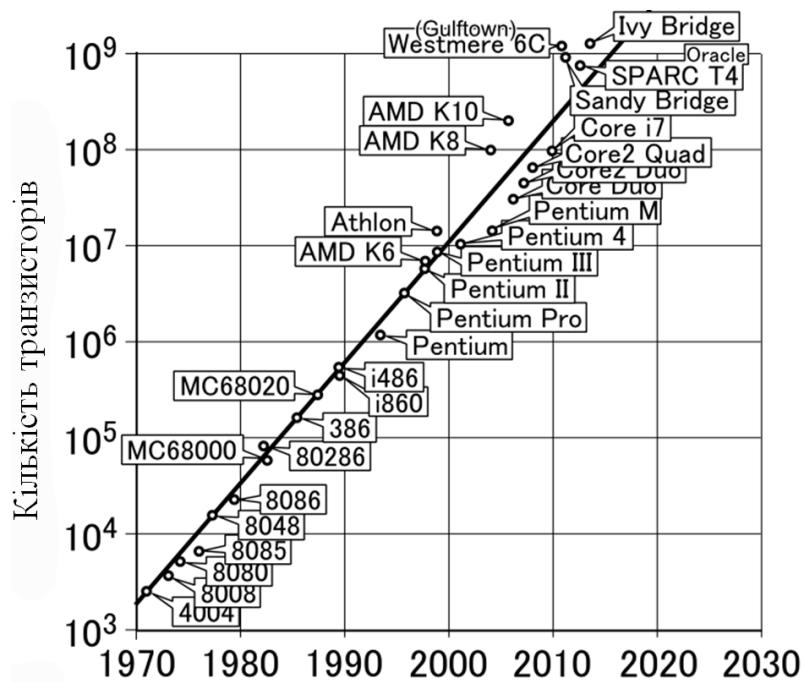


Рис. 1. Характеристика дії закону Мура: кількість транзисторів за роками

«Закон Мура» правильніше треба називати закономірністю, адже він не має ані математичного, ані фізичного підґрунтя, проте став зручним для визначення дій компаній-виробників апаратного забезпечення комп'ютерної техніки на майбутню перспективу. Наразі постає проблема у дії цієї закономірності, «закон Мура» досягає межі, оскільки виробникам все складніше розміщувати більшу кількість транзисторів на одній мікросхемі, не збільшуючи її розміру [11].

Висуваються різні припущення як «закон Мура» буде діяти у майбутньому. Всі вони базуються на необхідності пошуку принципово нової як апаратної, так і програмної реалізації сучасних інформаційних систем [8]. Нову революцію в ІТ-індустрії пов'язують з упровадженням квантових комп'ютерів [13].

Квантовими комп'ютерами називають комп'ютери, робота центральних процесорів яких базується на теоретичних засадах квантової механіки, а саме, квант, кубіт, квантова суперпозиція та квантова заплутаність.

Під *квантом* у фізиці розуміють неподільну частину будь-якої величини. *Кубіт* є новою альтернативою класичного біту, на відміну від якого, може набувати всіх можливих значень від 0 до 1 та перебувати в усіх цих станах одночасно (рис. 2).

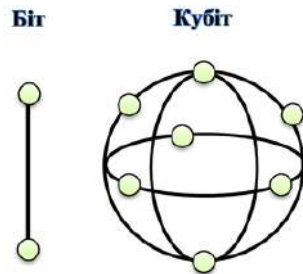


Рис. 2. Моделі класичного біту та квантового кубіту

Під *квантовою суперпозицією* розуміють здатність квантів існувати в декількох місцях або станах одночасно. Це дозволяє квантовим комп'ютерам опрацьовувати значно більшу кількість даних та пришвидшити виконання задач (задач оптимізації та криптографії, моделювання складних хімічних сполук), а також таких задач, що за допомогою класичних комп'ютерів взагалі не можуть бути розв'язані (наприклад, реалізація алгоритму Шора для розкладання цілих чисел на множники).

Під квантовою заплутаністю розуміють неможливість описати систему із кількох кубітів, використовуючи опис окремих кубітів, а не кореляцію між ними. Так, у класичному комп'ютері всі варіанти перераховуються послідовно, а у квантовому комп'ютері – паралельно, всі можливі варіанти обчислень існують одночасно, необхідно обрати правильний.

Над проектуванням (з 1981 р.) та реалізацією (з 1990 р.) квантових комп'ютерів працюють компанії IBM, Intel, Google, Microsoft та інші [13].

Так, з травня 2016 року компанія IBM надала хмарний доступ до свого квантового 5-кубітного комп'ютера, у травні 2017 року – до 16-кубітного, а у листопаді 2017 – до 20-кубітного, та проанонсувала створення 50-кубітного (рис. 3а) [4; 9]. Компанія Intel у жовтні 2017 року презентувала 17-кубітний квантовий чіп [5], а у січні 2018 – 49-кубітний (рис. 3б) [1].

З березня 2018 року новий квантовий комп'ютер Google використовує 72-кубітний квантовий процесор [2].

пропонує доступ для вивчення квантового програмування на платформі квантових обчислень QISKit з відкритим вихідним кодом [7].

Беручи до уваги значущість та перспективність квантового програмування, вважаємо актуальним і доцільним введення адаптованого змісту нового IT-напряму у шкільний курс інформатики для учнів старших класів інформаційно-технологічного профілю.

Список використаних джерел

1. Hsu J. CES 2018: Intel's 49-Qubit Chip Shoots for Quantum Supremacy [Electronic resource] / Jeremy Hsu // IEEE Spectrum. – 9 Jan 2018. – Access mode : <https://spectrum.ieee.org/tech-talk/computing/hardware/intels-49qubit-chip-aims-for-quantum-supremacy>.
2. Google Previews Bristlecone Quantum Processor [Electronic resource] // CdrInfo.com. – Mar 5, 2018. – Access mode : <https://www.cdrinfo.com/d7/content/google-previews-bristlecone-quantum-processor>.
3. IBM Q Experience [Electronic resource]. – Access mode : <https://quantumexperience.ng.bluemix.net/qx/experience>.
4. Трунин Д. IBM построила 50-кубитный квантовый компьютер [Электронный ресурс] / Дмитрий Трунин // N+1. – 13 ноября 2017. – Режим доступа : <https://nplus1.ru/news/2017/11/13/IBM-50-qubit>.
5. Moore S. K. Intel Accelerates Its Quantum Computing Efforts With 17-Qubit Chip [Electronic resource] / Samuel K. Moore // IEEE Spectrum. – 10 Oct 2017. – Access mode : <https://spectrum.ieee.org/tech-talk/computing/hardware/intel-accelerates-its-quantum-computing-efforts-with-17qubit-chip>.
6. Microsoft синтезировала уникальную частицу для создания квантового компьютера [Электронный ресурс] / Пресс-центр // Microsoft. – 10 April 2018. – Режим доступа : <https://news.microsoft.com/ru-ru/microsoft-sintezirovala-unikal-nuyu-chastitsu-dlya-sozdaniya-kvantovogo-komp-yutera>.
7. QISKit | Quantum Information Science Kit [Electronic resource] / IBM. – 2018. – Access mode : <https://qiskit.org>.
8. Аль-Килани В. Х. Будущее вычислительных технологий. Будет ли работать закон Мура дальше? [Электронный ресурс] / Аль-Килани Валерий Халидович, Умкеева Булгн Владимировна // Интернаука. – 2016. – Т. 1. – № 12 (22). – Режим доступа : <https://www.inter-nauka.com/issues/2016/12/1855>.
9. Скрипин В. Вот так выглядит 50-кубитный квантовый компьютер IBM [Электронный ресурс] / Владимир Скрипин // ИТС.ua. – 11.01.2018. – Режим доступа : <https://its.ua/blogs/vot-tak-vyiglyadit-50-kubitnyiy-kvantovyyiy-kompyuter-ibm>.
10. Душкин Р. В. Квантовые вычисления и функциональное программирование [Электронный ресурс] / Душкин Р. В. – Москва, 2014. – 318 с. – Режим доступа : <http://spkurdyumov.ru/uploads/2017/09/kvantovye-vychisleniya-i-funkcionalnoe-programmirovaniye.pdf>.
11. Fog A. Moores law hits the roof [Electronic resource] / Agner [Fog] // Agner's CPU blog. – 26.12.2015. – Access mode : <https://www.agner.org/optimize/blog/read.php?i=417>.
12. Квантовое будущее: Microsoft выпускает бесплатную предварительную версию пакета средств разработки Quantum Development Kit [Электронный ресурс] / Пресс-центр // Microsoft. – 11 December 2017. – Режим доступа : <https://news.microsoft.com/ru-ru/quantum-development-kit>.

13. Ершов А. Квантовое превосходство: всё о квантовых компьютерах [Электронный ресурс] / Александр Ершов. – 22 августа 2018. – Режим доступа : <https://www.popmech.ru/gadgets/434522-kvantovoe-prevoshodstvo-vsyo-o-kvantovyh-kompyuterah>.
14. Квантовые вычисления (Quantum computing) | Coursera [Электронный ресурс] / Coursera Inc. – 2018. – Режим доступа : <https://www.coursera.org/learn/kvantovyue-yuchisleniya>.
15. Учитесь с помощью Microsoft Quantum Katas [Электронный ресурс] / Пресс-центр // Microsoft. – 26 July 2018. – Режим доступа : <https://news.microsoft.com/ru-ru/microsoft-quantum-katas>.

References (translated and transliterated)

1. Hsu, J.: CES 2018: Intel's 49-Qubit Chip Shoots for Quantum Supremacy. IEEE Spectrum. <https://spectrum.ieee.org/tech-talk/computing/hardware/intels-49qubit-chip-aims-for-quantum-supremacy> (2018). Accessed 20 Nov 2018
2. Google Previews Bristlecone Quantum Processor. CdrInfo.com. <https://www.cdrinfo.com/d7/content/google-previews-bristlecone-quantum-processor> (2018). Accessed 20 Nov 2018
3. IBM Q Experience. <https://quantumexperience.ng.bluemix.net/qx/experience> (2018). Accessed 25 Oct 2018
4. Trunin, D.: IBM postroila 50-kubitnyi kvantovyi kompiuter (IBM built a 50-qubit quantum computer). N+1. <https://nplus1.ru/news/2017/11/13/IBM-50-qubit> (2017). Accessed 17 Aug 2018
5. Moore, S.K.: Intel Accelerates Its Quantum Computing Efforts With 17-Qubit Chip. IEEE Spectrum. <https://spectrum.ieee.org/tech-talk/computing/hardware/intel-accelerates-its-quantum-computing-efforts-with-17qubit-chip> (2017). Accessed 23 Jun 2018
6. Microsoft sintezirovala unikalnuiu chastitcu dlia sozdaniia kvantovogo kompiutera (Microsoft has synthesized a unique particle to create a quantum computer). <https://news.microsoft.com/ru-ru/microsoft-sintezirovala-unikal-nuyu-chastitsu-dlya-sozdaniya-kvantovogo-komp-yutera> (2018). Accessed 6 May 2018
7. QISKit | Quantum Information Science Kit. <https://qiskit.org> (2018). Accessed 17 Aug 2018
8. Al-Kilani, V., Umkeeva, B.: Budushchee vychislitelnykh tekhnologii. Budet li rabotat zakon Mura dalshe? (Future of computing technologies. Will work on Moore's law?). Internauka. **1**(12). <https://www.inter-nauka.com/issues/2016/12/1855> (2016). Accessed 8 Mar 2018
9. Skripin, V.: Vot tak vygliadit 50-kubitnyi kvantovyi kompiuter IBM (This is what IBM's 50-qubit quantum computer looks like). <https://itc.ua/blogs/vot-tak-vyiglyadit-50-kubitnyiy-kvantovyy-kompyuter-ibm> (2018). Accessed 23 Feb 2018
10. Dushkin, R.V.: Kvantovye vychisleniia i funktsionalnoe programmirovaniie (Quantum computing and functional programming). Moscow. <http://spkurdyumov.ru/uploads/2017/09/kvantovye-vychisleniya-i-funktsionalnoe-programmirovaniie.pdf> (2014). Accessed 6 May 2018
11. Fog, A.: Moores law hits the roof. Agner's CPU blog. <https://www.agner.org/optimize/blog/read.php?i=417> (2015). Accessed 10 May 2018
12. Kvantovoe budushchee: Microsoft vypuskaet besplatnuiu predvaritelnuiu versiiu paketa sredstv razrabotki Quantum Development Kit (Quantum Future: Microsoft Releases Quantum Development Kit Free Pre-Release Kit). <https://news.microsoft.com/ru-ru/quantum-development-kit> (2017). Accessed 5 Apr 2018

13. Ershov, A.: Kvantovoe prevoskhodstvo: vse o kvantovykh kompiuterakh (Quantum Excellence: Everything About Quantum Computers). <https://www.popmech.ru/gadgets/434522-quantovoe-prevoshodstvo-vsyo-o-quantovyh-kompyuterah> (2018). Accessed 11 Nov 2018
14. Kvantovye vychisleniia (Quantum computing). <https://www.coursera.org/learn/quantovyye-vychisleniya> (2018). Accessed 18 Nov 2018
15. Uchites s pomoshchiu Microsoft Quantum Katas (Learn with Microsoft Quantum Katas). <https://news.microsoft.com/ru-ru/microsoft-quantum-katas> (2018). Accessed 1 Sep 2018

Дослідження криптографічного алгоритму на основі тюрмітів на відповідність лавинному критерію

Лілія Олександрівна Фадєєва, Павло Володимирович Мерзликін^[0000-0002-0752-411X]

Криворізький державний педагогічний університет,
пр. Гагаріна, 54, м. Кривий Ріг, 50086, Україна
fadeyevaliliya@gmail.com, linuxoid@i.ua

Анотація. Досліджується запропонований криптографічний алгоритм на відповідність лавинному критерію. Представлено модифікацію алгоритму, що дозволяє задовольнити цей критерій без втрати інших властивостей алгоритму.

Ключові слова: криптографія, мураха Ленгтона, лавинний критерій.

The avalanche criterion satisfaction research of the turmite-based cryptographic algorithm

Liliia O. Fadiieva and Pavlo V. Merzlykin^[0000-0002-0752-411X]

Kryvyi Rih State Pedagogical University, 54, Gagarin Ave., Kryvyi Rih, 50086, Ukraine
fadeyevaliliya@gmail.com, linuxoid@i.ua

Abstract. The proposed cryptographic algorithm's avalanche criterion satisfaction is examined. The algorithm's modification to satisfy the criterion without algorithm's features loss is introduced.

Keywords: cryptography, Langton's ant, avalanche criterion.

1 Вступ

Тюрмітами називають один з двовимірних різновидів машини Тюринга, найбільш відомим з яких є Мураха Ленгтона [1]. Це клітинний автомат із дуже простими правилами. Кожна квадратна клітинка двовимірного поля може мати два стани (кольори): чорний і білий. В одній із клітинок знаходиться «мураха», яка на кожному кроці може рухатися в одному з чотирьох напрямків (у клітинку, що має спільну сторону з поточною). В чорній клітинці мураха повертає на 90° ліво, змінює колір клітинки на білий і робить крок до наступної клітинки. В білій

клітинці повертає на 90° вправо, змінює колір клітинки на чорний і робить крок до наступної клітинки. Інші варіанти тюрмітів можуть мати дещо відмінні правила руху. Незважаючи на простоту правил руху, такі системи демонструють досить складну поведінку й з ними пов'язаний ряд досі відкритих математичних проблем, що виходять за рамки цієї роботи. Однак можливість застосування таких алгоритмів в криптографічних цілях (завдяки їх здатності генерувати псевдовипадкові паттерни) лишається майже поза увагою дослідників. В цьому полягає актуальність роботи.

Якщо припустити, що в ролі початкового поля виступають певні дані (наприклад, фрагмент файлу), де двом кольорам відповідають значення бітів «0» і «1», можна зашифрувати, а згодом розшифрувати дані, знаючи кількість кроків і кінцеве положення мурахи.

Одна з найбільш успішних реалізацій криптографічного алгоритму з використанням тюрмітів [2] має такі недоліки, як обмеженість лише шифруванням зображень, а також більшу ймовірність модифікації молодших бітів, ніж старших, що може потенційно знижувати криптостійкість. З урахуванням цього, нами було запропоновано альтернативний криптографічний алгоритм з використанням тюрмітів. Його детальний огляд, оцінка швидкодії, оцінка потенційної стійкості до частотного аналізу та оптимізація параметрів алгоритму здійснені в роботі [3]. Як було показано, при достатньо великій кількості кроків одержується псевдовипадкова послідовність байтів з рівномірним розподілом, що ускладнює можливі атаки на основі частотного аналізу.

2 Методологія

Алгоритм, представлений у даній роботі, має таку схему:

- розбиваємо файл з вхідними даними на блоки однакового розміру;
- обираємо стартові позиції тюрмітів;
- далі, згідно правил руху мурахи Ленгтона, виконуємо побітове шифрування даних.

Даний алгоритм має циклічні граничні умови, тобто, якщо тюрміт наближається до границі блоку, то переходить на протилежну сторону.

Дана робота присвячена оцінці відповідності алгоритму лавинному критерію. Для того, аби криптографічний алгоритм відповідав цьому критерію, потрібно, щоб при зміні одного біту у вхідному файлі змінювалось близько половини бітів у вихідному.

3 Обговорення результатів

На рис. 1 подано матрицю різниці бітів для двох зашифрованих файлів, що відрізнялися на один біт. Чорні клітинки позначають ті біти, що відрізняються

після застосування нашого алгоритму. Можемо бачити, що у схожих файлах після шифрування відрізняється 92 біти, що становить 18 %. Це пов'язано насамперед зі способом розбиття вихідного файлу на блоки [3]. Оскільки блоки шифруються незалежно, модифікація одного біту вплине лише на розподіл бітів у відповідному блоці. Очевидно також, що відсоток відмінних бітів буде зменшуватися зі збільшенням розміру вихідних файлів.

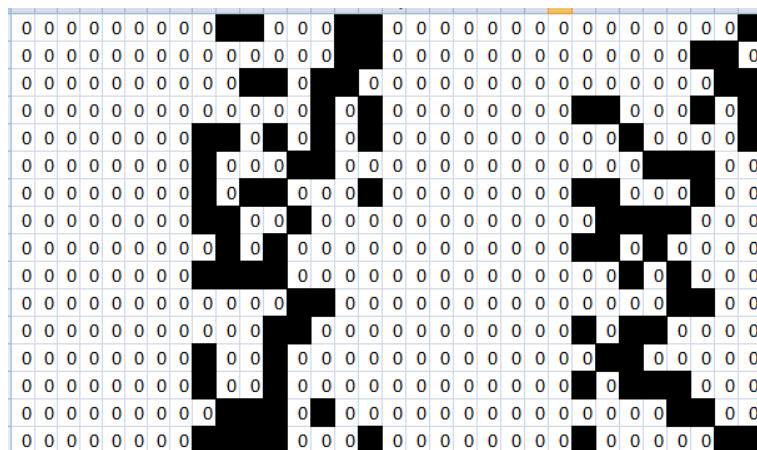


Рис. 1. Матриця різниці бітів у вихідній реалізації алгоритму

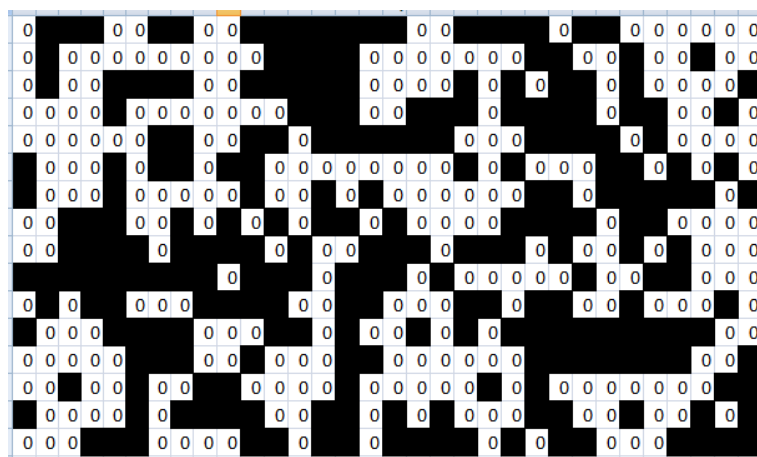


Рис. 2. Матриця різниці бітів для модифікованого алгоритму

Для вирішення цієї проблеми було внесено певні зміни в ініціалізацію початкових станів блоків. У вихідній версії алгоритму спосіб розбиття на блоки та початкове положення мурахи визначалися ключем користувача. Тобто зміна одного байту вихідного файлу впливала лише на один блок в зашифрованому

файлі. Запропонована модифікація полягає в тому, що ключ користувача впливає лише на спосіб розбиття на блоки, тоді як початкове положення й орієнтація мурахи в просторі визначаються випадково. Цей підхід видається цілком прийнятним, позаяк алгоритм є асиметричним і для дешифрування важливе кінцеве, а не початкове положення мурахи й кількість пройдених кроків. Тобто модифікація не призводить до збільшення довжини ключа й не впливає на характер розподілів бітів чи швидкодію.

Матриця різниці бітів для тих самих вихідних файлів показана на рис. 2. В зашифрованих файлах відрізняється 239 бітів, що становить 47 %. Це ілюструє відповідність алгоритму лавинному критерію. Додаткові тести на різних типах вхідних файлів показали аналогічні результати.

4 Висновки

Таким чином, запропонована модифікація алгоритму дозволяє задовольнити лавинний критерій без втрати інших його властивостей. В майбутньому планується розширити алгоритм для тюрмітів з різними правилами руху та створити модифікацію алгоритму для роботи в реальному часі, наприклад, для шифрування трафіку.

Список використаних джерел

1. Langton C. G. Studying Artificial Life with Cellular Automata / Crystopher G. Langton // *Physica D: Nonlinear Phenomena*. – 1986. – Vol. 22. – Iss. 1-3. – P. 120-149. – DOI : 10.1016/0167-2789(86)90237-X.
2. Wang X. A novel image encryption scheme using chaos and Langton's Ant cellular automaton / Xingyuan Wang, Dahai Xu // *Nonlinear Dynamics*. – 2015. – Volume 79. – Issue 4. – P. 2449-2456. – DOI : 10.1007/s11071-014-1824-0.
3. Мерзликін П. В. Криптографічний алгоритм на основі системи тюрмітів / П. В. Мерзликін, Л. О. Фадєєва, В. Ю. Іваницька, Є. Є. Іваницька // *Актуальні питання забезпечення кібербезпеки та захисту інформації*. – Київ : Видавництво Європейського університету, 2018. – С. 97-102.

References (translated and transliterated)

1. Langton, C.G.: Studying Artificial Life with Cellular Automata. *Physica D: Nonlinear Phenomena*. **22**(1–3), 120–149. (1986). doi:10.1016/0167-2789(86)90237-X
2. Wang X., Xu, D.: A novel image encryption scheme using chaos and Langton's Ant cellular automaton. *Nonlinear Dynamics*. **79**(4), 2449–2456 (2015). doi:10.1007/s11071-014-1824-0
3. Merzlykin, P.V., Fadieieva, L.O., Ivanytska, V.Yu., Ivanytska, Ye.Ye.: Kryptohrafichnyi alhorytm na osnovi systemy tiurmitiv (Turmites System Based Cryptography Algorithm). In: *Aktualni pytannia zabezpechennia kiberbezpeky ta zakhystu informatsii*, pp. 97–102. Vydavnytstvo Yevropeiskoho universytetu, Kyiv (2018)

Інтелектуальний веб-інтерфейс доступу до даних особливої важливості

Владислав Костянтинович Гуменок, Ігор Анатолійович Котов^[0000-0003-2445-6259]

Криворізький національний університет,
вул. Віталія Матусевича, 11, м. Кривий Ріг, 50027, Україна
{gumenok.vladislav, rioexito}@gmail.com

Анотація. Мета роботи – розробка і дослідження інтелектуального веб-інтерфейсу доступу до даних особливої важливості. Об'єкт проектування – інтелектуальна інформаційна технологія доступу до даних особливої важливості. У теоретичній частині проведено аналіз проблем забезпечення безпеки конфіденційних даних, розглянуті існуючі способи захисту інформації, визначено актуальність та завдання дослідження. У практичній частині роботи проведено аналіз принципів побудови систем захисту інформації, розроблено структурні та функціональні моделі, розроблено інтелектуальний блок доступу до даних особливої важливості. Досліджено рівень надійності та стійкості питань системи до набору даних, зроблені висновки, що розроблений веб-інтерфейс дозволяє підвищити рівень захищеності інформації.

Ключові слова: безпека, база даних, інтерфейс, інтелектуальний, загрози, забезпечення захисту, функціональна схема, інформаційна система, автентифікація.

Intelligent web-interface for access to data of particular importance

Vladyslav K. Humenok, Igor A. Kotov^[0000-0003-2445-6259]

Kryvyi Rih National University, 11, Vitalii Matusevych St., Kryvyi Rih, 50027, Ukraine
{gumenok.vladislav, rioexito}@gmail.com

Abstract. The purpose of the work is to develop and research the intellectual web-interface of access to data of particular importance. Design object – intelligence information technology access to data of particular importance. In the theoretical part the analysis of security problems of confidential data was carried out, existing methods of information protection were considered, the relevance and objectives of the research were determined. In the practical part of

the work, an analysis of the principles of construction of information security systems has been developed, structural and functional and structural models are developed, the intellectual access block of data of special importance has been developed. The level of reliability and stability of the system issues prior to the data selection is investigated, the conclusions are drawn that the developed web-interface allows to increase the level of information security.

Keywords: security, database, interface, intelligence, threats, security, functional diagram, information system, authentication.

1 Вступ

Забезпечення безпеки даних – одна з головних вимог в умовах сучасного глобалізованого світу. Серед джерел загроз безпеки інформації розрізняють стихійні – викликані впливом на інформаційне середовище об'єктивних фізичних процесів або природних явищ, технічні – зумовлені зношенням апаратних засобів, що зберігають інформацію та антропогенні, коли загроза безпеки даних походить від самої людини. В межах державних організацій різного рівня міститься інформація, що відноситься до категорії особливої важливості. Такі дані мають найвищий ступінь секретності та присвоюються суворо обмеженій групі документів. Розкриття інформації з таким грифом секретності несе в собі надзвичайну небезпеку для країни і може нанести значний збиток інтересам держави, оскільки містить інформацію, що стосується оборонної, економічної та контррозвідальної сфер діяльності.

Найбільш розповсюдженою причиною витоку секретної інформації є саме антропогенний фактор. Його проявами можуть бути як ненавмисні дії працівників організації через необережне поводження з інформацією, так і цілеспрямовані дії з боку зловмисників, метою яких є викрадення важливих даних.

Забезпечення і підтримка інформаційної безпеки передбачає комплекс різнопланових заходів, які запобігають, відстежують та усувають можливість несанкціонованого доступу до даних. Подібні заходи включають в себе апаратні та програмні засоби захисту. Ці засоби дозволяють забезпечити механізми автентифікації, які на підставі наданих користувачем предметів автентифікації або його біометричних параметрів приймають рішення про те, чи є користувач тим, за кого себе видає.

Автентифікація є складовою частиною систем захисту інформації (СЗІ), діяльність яких спрямована на захист від несанкціонованого доступу, пошкодження, або спотворення інформації та гарантування дотримання ключових принципів захисту інформації, які полягають у забезпеченні конфіденційності, цілісності та доступності даних.

2 Мета і задачі дослідження

Метою роботи є дослідження засобів захисту інформації на основі існуючих аналогів, та розробка інтелектуального веб-інтерфейсу доступу до даних особливої важливості.

З метою досягнення поставленої мети дослідження виникає необхідність постановки та виконання наступних задач:

- розробка загальної структури інформаційної системи;
- побудова функціональної моделі інформаційної системи;
- розробка взаємодій компонентів інформаційної системи;
- розробка структури бази даних інформаційної системи.

Об'єктом дослідження є інтелектуальна інформаційна технологія забезпечення доступу до даних особливої важливості.

Предметом дослідження є комплекс організаційних заходів і програмно-технічних засобів забезпечення безпеки інформації.

Методи дослідження. У процесі дослідження застосовувалися такі наукові методи: порівняльний метод, емпіричний метод, пояснення і узагальнення. Методи аналізу та оцінки дозволили розробити програмні модулі інформаційної системи та її бази даних.

Практичне значення одержаних результатів дослідження полягає у тому, що теоретичні і методичні положення, висновки і рекомендації доведені до рівня практичних розробок, які сприяють впровадженню інноваційної продукції, ефективної системи доступу до конфіденційних даних, що дасть можливість підвищити рівень захищеності інформації та дозволить визначити ключові напрямки для подальшого удосконалення.

3 Аналіз систем захисту інформації та постановка проблеми

Система захисту інформації – сукупність організаційних і інженерно-технічних заходів, які спрямовані на забезпечення захисту інформації від розголошення, витоку і несанкціонованого доступу [1]. Захист інформації здійснюється на всіх етапах існування інформаційних систем шляхом об'єднання відповідно до прийнятої політики безпеки в єдину систему захисту правових, організаційних, інженерно-технічних та апаратно-програмних методів, способів та засобів захисту інформації [2]. Для забезпечення належного рівня захисту інформації державними органами передбачено ведення контролю за функціональним станом СЗІ. Заходи контролю передбачають процедури атестації та ліцензування об'єктів інформації для оцінки підготовленості систем і засобів інформатизації та зв'язку до обробки інформації, що містить державну таємницю.

Безпосередній процес функціонування СЗІ передбачає розподілення між користувачами необхідних реквізитів захисту інформації (паролів, привілеїв, ключів), надання користувачам прав доступу до ресурсів системи згідно з

прийнятою політикою безпеки та їх ліквідація по закінченню потреб у доступі. Загальна схема контролю захисту інформації в межах СЗІ представлена на рис. 1.

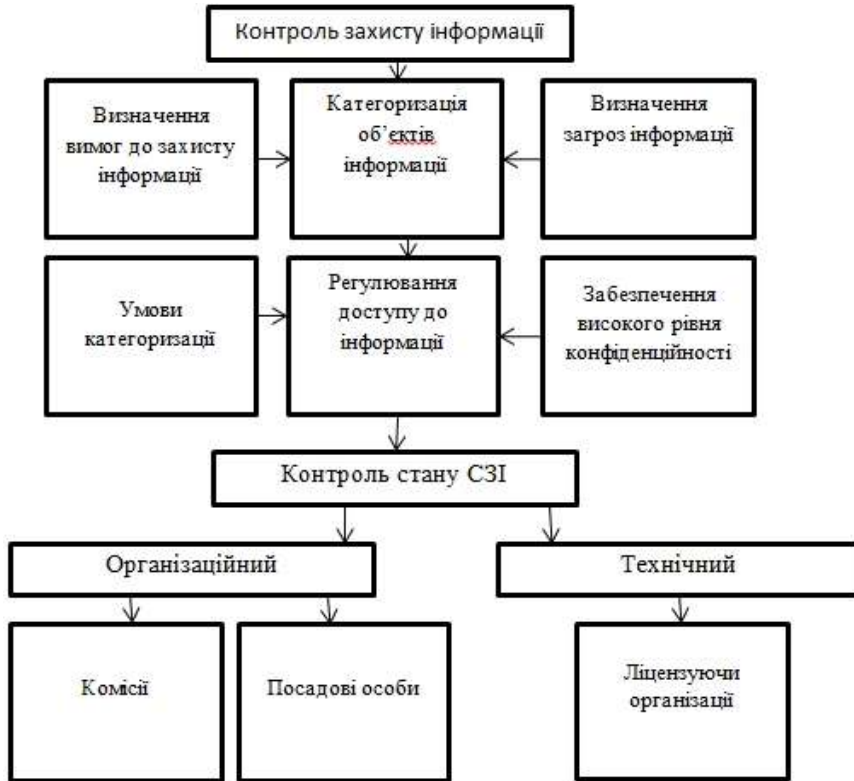


Рис. 1. Загальна схема контролю захисту інформації

Основна проблема, виявлена в процесі дослідження, полягає в тому, що забезпечення функціонування потребує використання значної кількості апаратно-технічних засобів. Це зумовлює необхідність залучення контролюючих органів та третіх осіб, що призводить до сповільнення процесу розробки та введення в експлуатацію системи захисту і підвищує ймовірність витоку конфіденційної інформації. Також виявлено, що у якості елементів автентифікації використовуються застарілі методи, такі як паролі або ключі доступу, що зберігаються на фізичних носіях та можуть бути втрачені.

4 Матеріали і методи досліджень

Система інтелектуального доступу до даних виконує ряд функцій, спрямованих на забезпечення безпеки інформації. Ключовими є автентифікація користувача та визначення його прав доступу.

Система автентифікації користувача включає в себе ряд питань, які містять особисту інформацію та забезпечують безпомилкове визначення користувача, під час отримання доступу.

Для визначення функціональних можливостей програмного продукту розробляється функціональна схема. Така схема дозволяє відобразити принципи роботи та визначити зв'язки між компонентами програми. На рис. 2 наведена функціональна схема додатку інтелектуального доступу до даних.

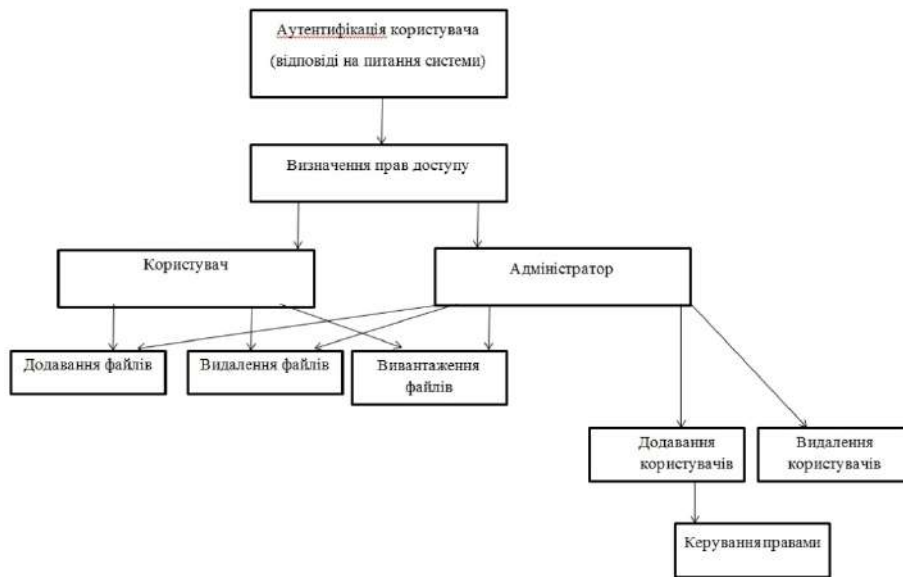


Рис. 2. Функціональна схема додатку

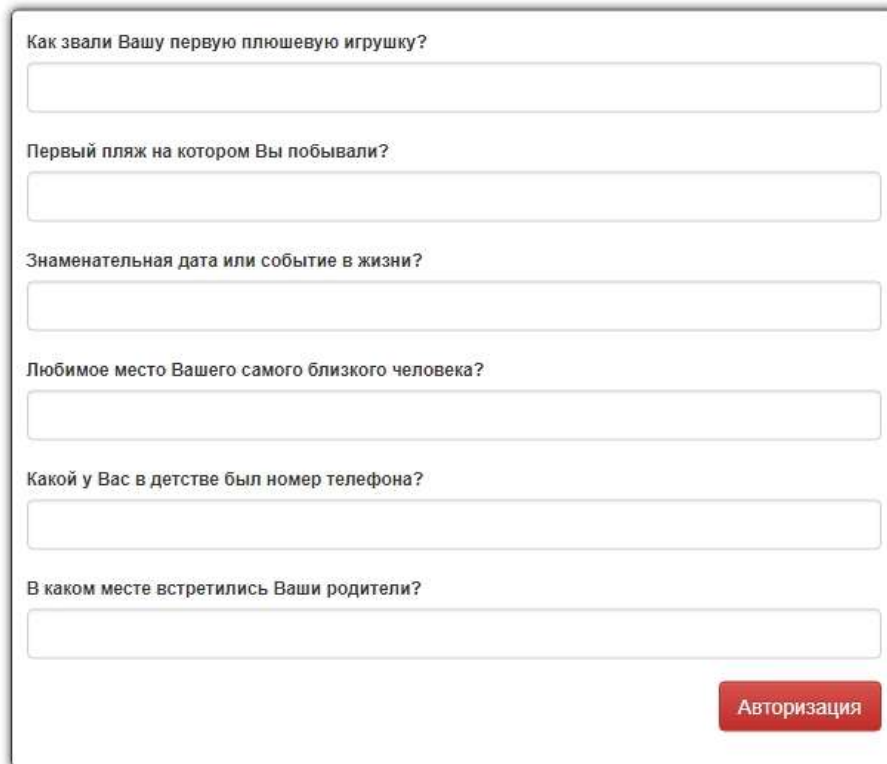
Функціональна схема містить інформацію про способи реалізації додатком заданих функцій. За такою схемою можна визначити, як здійснюються перетворення і які для цього необхідні функціональні елементи. Кожен функціональний елемент містить лише ті входи і виходи, які необхідні для його коректної роботи.

5 Результати

Оцінка програми інтелектуального доступу до даних особливої важливості буде здійснюватися за наступними критеріями:

- оцінка веб-інтерфейсу додатку;
- визначення характеристик додатку;
- оцінка захисту додатку;
- оцінка ефективності додатку в цілому.

При потраплянні на сторінку входу до системи користувача зустрічає форма вводу даних (рис. 3).



Как звали Вашу первую плюшевую игрушку?

Первый пляж на котором Вы побывали?

Знаменательная дата или событие в жизни?

Любимое место Вашего самого близкого человека?

Какой у Вас в детстве был номер телефона?

В каком месте встретились Ваши родители?

Авторизация

Рис. 3. Вікно входу до програми

Форма містить перелік питань, які передбачають введення особистої інформації. Після почергової відповіді на них користувач натискає кнопку авторизації. Інтелектуальний модуль доступу до інформації порівнює отриману інформацію з даними, що містяться в БД та визначає відсоткове відношення коректних відповідей. У разі, якщо даний відсоток досягає певного значення система визначає можливість отримання доступу до інформації та визначає права користувача на використання її ресурсів.

У цілому програма виконує свої функції, що полягають у забезпеченні інтелектуального доступу до інформації. Вона дозволяє здійснювати додавання та видалення інформації у разі необхідності й забезпечує інформацію від несанкціонованих дій з боку інших користувачів. Розроблений веб-ресурс забезпечує створення єдиної системи доступу до даних особливої важливості.

6 Висновки

У роботі досягнута основна мета, яка полягає у дослідженні теоретико-методичних та наукових підходів до забезпечення безпеки даних особливої важливості та на основі досліджень сучасних програмних аналогів розроблено веб-інтерфейс, що здатний надати можливість безпечного доступу до важливої інформації. На підставі цього можна зробити наступні висновки:

- розроблено загальну структури інформаційної системи;
- побудована функціональна моделі інформаційної системи;
- розроблено взаємодію компонентів інформаційної системи;
- розроблена структура бази даних інформаційної системи.

У результаті проведеного дослідження щодо формування ціни на інформаційну систему виявлено, що кінцева ціна продукту значно менше від ціни існуючих аналогів, а рентабельність її виготовлення становить 12,3 %. Таким чином, розробка та впровадження інтелектуального веб-інтерфейсу доступу до даних особливої важливості є економічно доцільними.

Список використаних джерел

1. Про затвердження Правил забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах [Електронний ресурс] : Постанова № 373 / Кабінет Міністрів. – Київ, 29 березня 2006 р. – Режим доступу : <http://zakon.rada.gov.ua/laws/show/373-2006-%D0%BF>.
2. Про інформацію [Електронний ресурс] : Закон України. – 02 жовтня 1992 р. – Режим доступу : <http://zakon.rada.gov.ua/laws/main/2657-12>.

References (translated and transliterated)

1. Pro zatverdzhennia Pravyl zabezpechennia zakhystu informatsii v informatsiinykh, telekomunikatsiinykh ta informatsiino-telekomunikatsiinykh systemakh (On Approval of the Rules for the protection of information in information, telecommunication and information and telecommunication systems). <http://zakon.rada.gov.ua/laws/show/373-2006-%D0%BF> (2006). Accessed 13 Sep 2018
2. Zakon Ukrainy Pro informatsiiu (Law of Ukraine On information). <http://zakon.rada.gov.ua/laws/main/2657-12> (1992). Accessed 13 Sep 2018

Автоматизована система оперативного розподілу ресурсів ліквідації комплексних надзвичайних ситуацій

Дмитро Олександрович Ганістрат, Ігор Анатолійович Котов^[0000-0003-2445-6259]

Криворізький національний університет,
вул. Віталія Матусевича, 11, м. Кривий Ріг, 50027, Україна
{diman2pa, rioexito}@gmail.com

Анотація. Мета роботи – розробка і дослідження програмного комплексу для автоматизації розподілу ресурсів ліквідації комплексних надзвичайних ситуацій. Об'єкт проектування – інтелектуальна інформаційна технологія автоматизації управління ресурсами ліквідації надзвичайних ситуацій. Проведено критичний аналіз сучасного стану проблеми ліквідації надзвичайних станів в промисловості та природному оточенні, розглянуті існуючі аналоги програмного забезпечення. Здійснений аналіз принципів побудови автоматизованих систем та розподілених систем керування базами даних, розроблені математичні і структурні моделі, розроблена автоматизована система управління ресурсами ліквідації надзвичайних ситуацій. Досліджено забезпечення безпеки інформаційних потоків даних програмного комплексу, технічна ефективність прийнятих рішень. Зроблені висновки, що розроблений комплекс автоматизації дозволяє реалізовувати ефективне управління ресурсами ліквідації надзвичайних ситуацій.

Ключові слова: база даних, діаграма станів, моніторинг, надзвичайна ситуація, управління, потік даних, сервер.

Automated system of operational resource allocation for complex emergency situations

Dmytro O. Hanistrat and Igor A. Kotov^[0000-0003-2445-6259]

Kryvyi Rih National University, 11, Vitalii Matusevych St., Kryvyi Rih, 50027, Ukraine
{diman2pa, rioexito}@gmail.com

Abstract. The purpose of the work is the development and study of a software package for automating the allocation of resources to eliminate complex emergency situations. The object of design is an intelligent information

technology to automate the management of emergency response resources. A critical analysis of the current state of the emergency response problem in industry and the natural environment has been carried out, and existing analogues of software have been considered. The analysis of the principles of building automated systems and distributed database management systems, developed mathematical and structural models, developed an automated resource management system for emergency response. Investigated the security of information flow of data software complex, the technical effectiveness of decisions. It is concluded that the developed automation complex allows for the implementation of effective resource management for emergency response.

Keywords: database, status chart, monitoring, emergency, management, data flow, server.

1 Вступ

Безпека – одна зі складових життя. Джерелом загрози життя є різні надзвичайні ситуації (НС), які можуть трапитись несподівано. Причинами НС можуть бути різні фактори: техногенні катастрофи, аварії, пожежі, стихійні лиха чи епідемія. Головним фактором запобігання НС є інформація.

Оповіщення та надання необхідної інформації виконують автоматизовані системи раннього оповіщення населення. Ці системи допомагають людям, працівникам, населенню евакуюватись, кооперуватись, надають важливу інформацію. В службах МНС застосовують автоматизовану систему зв'язку, оповіщення і оперативного управління, що викликає різні ешелони сили, які проводять ліквідаційні роботи та рятувальні операції для постраждалих людей.

Також, необхідно застосовувати різні засоби для ліквідації загроз, будь-то специфічні засоби, такі як вогнегасники, респіратори, сокири тощо, аж до спеціальної техніки, як службові або пожежні машини. Для автоматизації управління цими засобами використовують різні СУБД, які дозволяють оптимізувати процеси управління засобами ліквідації НС.

2 Аналіз джерел і постановка проблеми

Система управління в НС – це сукупність органів і пунктів керування, систем комунікацій, автоматизації управління і оповіщення [1]. За планом реалізації оперативного виявлення та протидії НС, заздалегідь призначаються найближчі територіальні сили і засоби діяльних і місцевих підсистем єдиної державної системи (ЄДС), які застосовують заходи для ліквідації небезпеки та її наслідків [2]. Кількість потрібних сил та методів встановлюється шляхом урахування масштабу і змісту заходів.

Суттєву частину ліквідаційних робіт виконують аварійні відділи потенційно небезпечного об'єкта (ПНО), на котрому виникла НС, а також допомагають їм такі підрозділи, як: Державна служба з надзвичайних ситуацій (ДСНС), Міністерство внутрішніх справ (МВС) і Міністерство охорони здоров'я (МОЗ).

Для усунення НС та її наслідків зможуть приєднуватись підвідділи збройних сил та військових частин у порядку, прописаному в Законі України «Про правовий режим надзвичайного стану» та «Про Збройні Сили України» [3]. Реалізація сил та методів має бути відповідно обрахована для суб'єктів впливу та для оперативного їх здійснення. Щоб ліквідувати НС державного і обласного ступенів, організуються ешелонні групування органами управління.

У зоні НС за рішенням органу управління, нарощення груп сил та методів здійснюється планово для оптимізації реалізації задачі з організації та невідкладних робіт. Дійсний керівник, який призначається для ліквідації НС або її наслідків, організовує керівництво над силами і службами з усуненням НС [4]. Для керування та координації дій сил та органу управління створюється дійовий центр з усунення НС, який організовує аварійно-рятувальні роботи. Ієрархія органів управління в НС показана на рис. 1.

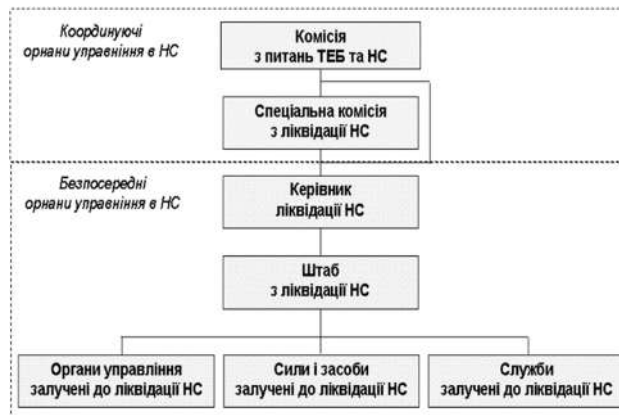


Рис. 1. Загальна схема органів управління в надзвичайних ситуаціях

Основна проблема полягає в тому, що на сучасному етапі інтенсивність НС значно зростає. При цьому основні методи розподілу ресурсів залишилися колишніми і не відповідають сучасним вимогам. Головним недоліком є жорстка прив'язка ресурсів ліквідації НС до баз зберігання і відсутність можливостей їх оперативного переміщення і обліку.

3 Мета і задачі дослідження

У службах МНС застосовують автоматизовану систему зв'язку, оповіщення і оперативного управління (АСЗОУ), яка розроблена для підняття ефективності використання сил і засобів підрозділів оперативно-рятувальних служб (ОРС) шляхом автоматизації рішення управлінських задач.

Однак, дослідження аналогів існуючих автоматизованих систем виявило, що державна автоматизована система управління засобами ліквідації НС потребує

значного удосконалення її програмних компонентів. Тому проблема дослідження є актуальною.

Метою роботи є дослідження автоматизованих систем управління на основі існуючих аналогів, аналізу факторів автоматизованого управління та розробка автоматизованої системи управління засобами ліквідації НС.

Об'єктом дослідження є інтелектуальна інформаційна технологія автоматизації управління ресурсами ліквідації НС.

Предметом дослідження є сукупність різних автоматизованих систем та СУБД для оптимізації управління, суб'єкти та об'єкти управління засобами ліквідації НС.

Досягнення поставленої мети обумовило необхідність постановки та виконання наступних *задач*:

- побудова структурної моделі інформаційного забезпечення АСУ;
- розробка загальної структури програмного комплексу;
- побудова моделі інформаційних потоків системи автоматизованого управління;
- розробка структури блоків інформаційного забезпечення;
- побудова функціональної моделі інформаційного забезпечення;
- розробка взаємодій компонентів програмного комплексу;
- розробка комплексної моделі інформаційного забезпечення системи автоматизованого управління;
- розробка об'єктів класів єдиної інформаційної бази;
- розробка бази даних єдиної інформаційної бази;
- оптимізація інформаційних потоків даних програмного комплексу.

У процесі дослідження застосовувалися такі *методи*: порівняльний метод, емпіричний метод, пояснення і узагальнення. Методи аналізу та оцінки дозволили розробити програмні модулі автоматизованої системи та її бази даних.

Практичне значення одержаних результатів дослідження полягає у тому, що теоретичні і методичні положення, висновки і рекомендації доведені до рівня практичних розробок, які сприяють впровадженню інноваційної продукції, ефективної системи автоматизації управління, що дасть можливість підвищити рівень розробки програмних продуктів та надання ідей для подальшого удосконалення.

4 Вихідні дані

АСЗООУ складається з взаємопов'язаних функціональних елементів, до яких відносяться:

- система оперативного-диспетчерського управління (СОДУ);
- система оперативного-диспетчерського зв'язку (СОДЗ);
- система організаційного і правового забезпечення (СОПЗ);
- інформаційно-керуюча обчислювальна система (ІКОС).

СОДУ включає в себе обчислювальну підсистему і підсистему телевізійної обробки даних, які призначені для вирішення оперативно-тактичних задач керування силами і засобами ОРС.

При створенні програмного продукту виділяються багаторазово використовувані модулі, проводиться їх типізація і уніфікація. На рис. 2 наведена структура комплексу інформаційного забезпечення автоматизованого управління засобами ліквідації НС, що складається з окремих програмних модулів, бази даних, об'єктів, бібліотек процедур та функцій.

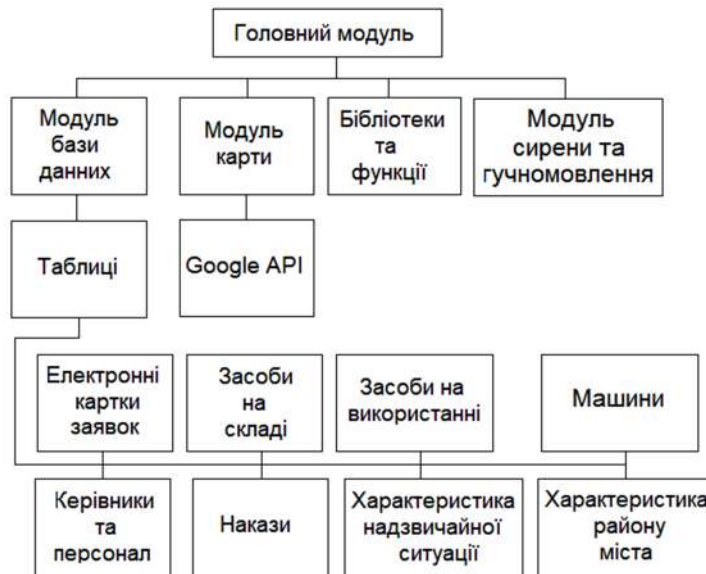


Рис. 2. Структурна схема АСЗОУ

Основна мета діаграми інформаційних потоків полягає в тому, щоб джерела, які надсилають та отримують інформацію, могли бути акуратно відображені та проаналізовані. Це дозволяє глядачам бачити передачу інформації та аналіз різних ситуацій. Створення IFD в більшості випадків є першим кроком у аналізі інформації. Інформація, що рухається вздовж діаграми, представлена як інформаційні елементи, або конкретні класифікатори.

IFD використовуються для:

- розробки високого рівня огляду потоку інформації в організації;
- виділення детальних потоків в індивідуальному завданні;
- описання потоку інформації всередині та навколо організацій та між відділами.

Основна вхідна інформація в автоматизованій системі управління засобами ліквідації НС буде надходити за допомогою людини. Користувач, оператор чи

персонал будуть додавати інформацію до бази даних, де вона буде оброблятися та зберігатись.

Узагальнена діаграма руху інформаційних потоків автоматизованої системи управління засобами ліквідації НС зображена на рис. 3.

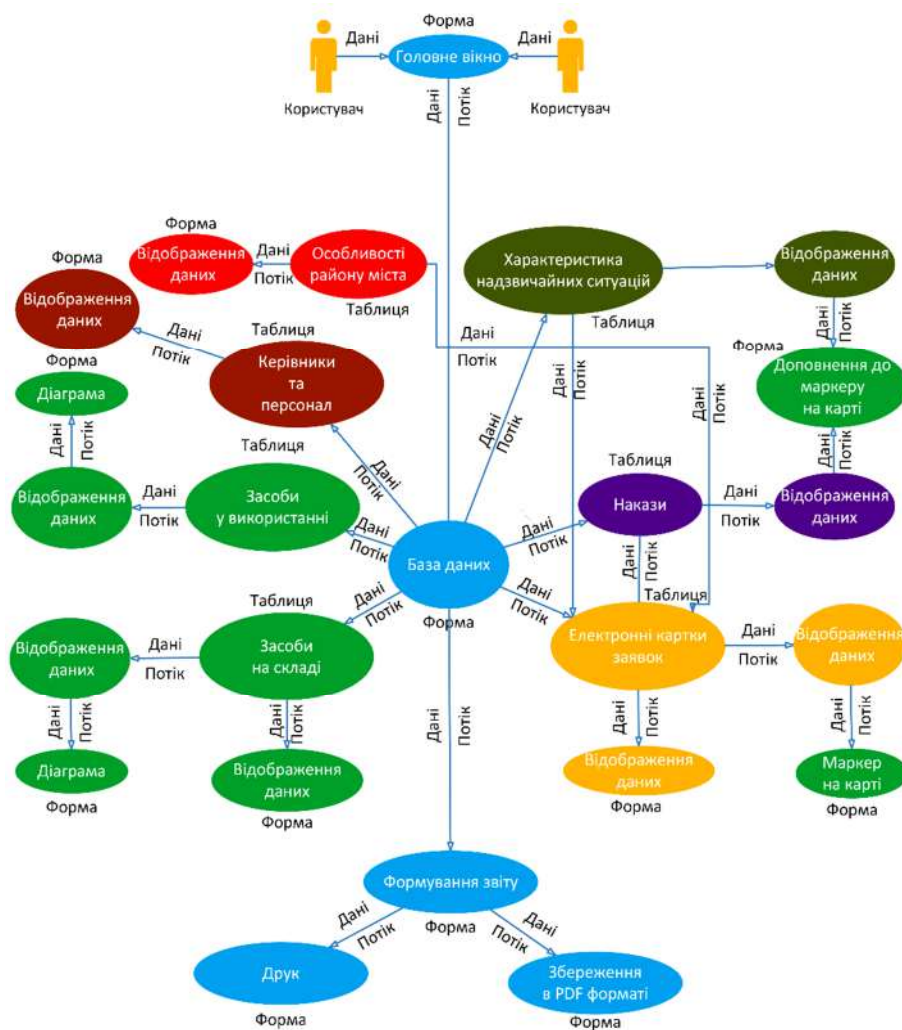


Рис. 3. Узагальнена діаграма руху інформаційних потоків

Вся вхідна інформація заноситься в програму і додається в базу даних, де обробляється. Інформація про засоби може відобразитись у вигляді діаграми для кращого сприйняття, а заявки – відобразитись на карті у вигляді маркерів. Виведення інформації із системи виконує функція звітування, яка надає можливість зберегти потрібну інформацію у файлі або роздрукувати документ.

5 Результати

Оцінка програми автоматизованої системи управління засобами ліквідації НС проводилась за наступними критеріям:

- оцінка інтерфейсу програми;
- визначення характеристик програми;
- оцінка захисту програми;
- мова програми;
- оцінка ефективності програми в цілому.

При запуску програми і при введенні логіну та паролю, користувача зустріне головне вікно, в якому будуть міститись три головних модуля із чотирьох, такі як: сирена; гучне мовлення; карти; та функція блокування програми; а на іншій формі, яка викликається через кнопку на головній формі, міститься база даних, де виконуються всі операції з обліку заявок та засобів НС.

Коли програма запускається, вона по замовченню відкривається, від самого початку розробки програми, як вікно. В такому режимі розташування всіх елементів управління і модулів робить програму лаконічною та зручною. Такий режим форми зручний для користувачів, у яких бракує місця на робочому столі або мала роздільна здатність екрану. Навіть у режимі вікна користуватися картою цілком зручно. На рис. 4. зображена форма програми, яка відкрита у вікні.

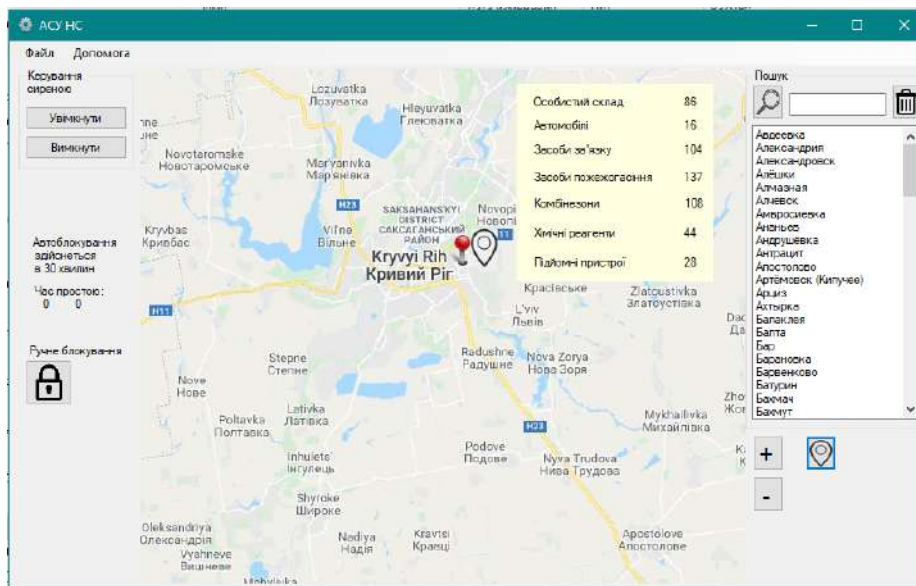


Рис. 4. Загальний режим роботи програми

У цілому програма виконує функції оповіщення та передавання інформації до бази даних. Карти працюють як окремий модуль без використання браузера, що надає «легкість» програмі та не навантажує комп'ютер. Не менш значимою функцією є логування всіх дій та подій, які відбуваються у програмі та які будуть інформаційними для користувача.

6 Висновки

У роботі досліджено підходи до оптимізації управління засобами ліквідації НС та розроблена програма для автоматизації управління засобами ліквідації НС. У результаті проведеного дослідження щодо формування ціни на програмну інновацію виявлено, що кінцева ціна автоматизованої системи значно менше ціни конкуренту, а рентабельність її виготовлення становить 15,6 %. Таким чином, розробка й упровадження програмного комплексу для автоматизованого розподілу ресурсів ліквідації НС є економічно доцільними.

Список використаних джерел

1. Про Концепцію захисту населення і територій у разі загрози та виникнення надзвичайних ситуацій [Електронний ресурс] : Указ № 284/99 / Президент України. – 26.03.1999. – Режим доступу : <http://zakon.rada.gov.ua/laws/show/284/99>.
2. Про єдину державну систему запобігання і реагування на надзвичайні ситуації техногенного та природного характеру [Електронний ресурс] : Постанова № 1198 / Кабінет Міністрів України. – 03.08.1998. – Режим доступу : <http://zakon.rada.gov.ua/laws/show/1198-98-%D0%BF>.
3. Про правовий режим надзвичайного стану [Електронний ресурс] : Закон України № 1550-III. – 16.03.2000. – Режим доступу : <http://zakon.rada.gov.ua/laws/show/1550-14>
4. Про затвердження Положення про штаб з ліквідації надзвичайної ситуації техногенного та природного характеру [Електронний ресурс] : Постанова № 1201 / Кабінет Міністрів України. – 19.08.2002. – Режим доступу : <http://zakon.rada.gov.ua/laws/show/1201-2002-%D0%BF>.

References (translated and transliterated)

1. Ukaz Prezydenta Ukrainy Pro Kontseptsiiu zakhystu naseleння i terytorii u razi zahrozy ta vynyknennia nadzvychainykh sytuatsii (Decree of the President of Ukraine On the concept of protection of population and territories in the event of a threat and emergency). <http://zakon.rada.gov.ua/laws/show/284/99> (1999)
2. Postanova Kabinetu Ministriv Ukrainy Pro yedynu derzhavnu systemu zapobihannia i reahuvannia na nadzvychaini sytuatsii tekhnohennoho ta pryrodnoho kharakteru (Resolution of the Cabinet of Ministers of Ukraine On the unified state system for prevention and response to emergencies of man-made and natural characteristics). <http://zakon.rada.gov.ua/laws/show/1198-98-%D0%BF> (1998)
3. Zakon Ukrainy Pro pravovyi rezhym nadzvychainoho stanu (The Law of Ukraine On the legal status of the state of emergency). <http://zakon.rada.gov.ua/laws/show/1550-14> (2000)

4. Postanova Kabinetu Ministriv Ukrainy Pro zatverdzhennia Polozhennia pro shtab z likvidatsii nadzvychainoi sytuatsii tekhnohenoho ta pryrodnoho kharakteru (Resolution of the Cabinet of Ministers of Ukraine On approval of the regulations on the headquarters for liquidation of the state of emergency of man-made and natural character). <http://zakon.rada.gov.ua/laws/show/1201-2002-%D0%BF> (2002)

Програмна розробка алгоритму адаптації дизайну сайту для людей з порушенням кольоросприйняття

Владислав Миколайович Пірогов, Анна Михайлівна Горло,
Ірина Сергіївна Мінтій^[0000-0003-3586-4311]

Криворізький державний педагогічний університет,
пр. Гагаріна, 54, м. Кривий Ріг, 50086, Україна
{pirogov1818, stifal1996}@gmail.com, irina.mintiy@kdpu.edu.ua

Анотація. Метою даного дослідження є програмна реалізація основних алгоритмів адаптації дизайну сайту для людей з порушенням кольоросприйняття. У роботі розглянуто реалізацію наступних алгоритмів: конвертації із колірної моделі rgb до lms моделі (для моделювання різних видів порушень кольоросприйняття існує спеціальний алгоритм саме в колірній моделі lms), моделювання різних видів дальтонізму в lms моделі (перетворення нормальних значень в колірній моделі lms до значень з різними видами порушень кольоросприйняття), конвертації даних із колірної моделі lms до rgb моделі, конвертації з колірної моделі rgb до hsl моделі, фільтрації кольорів у hsl моделі і конвертації з hsl моделі до колірної моделі rgb.

Ключові слова: порушення кольоросприйняття, колірна модель, веб-дизайн.

Software development of the algorithm of adapting of the website design for people with color-blindness

Vladyslav M. Pirohov, Anna M. Horlo and Iryna S. Mintiy^[0000-0003-3586-4311]

Kryvyi Rih State Pedagogical University, 54, Gagarin Ave., Kryvyi Rih, 50086, Ukraine
{pirogov1818, stifal1996}@gmail.com, irina.mintiy@kdpu.edu.ua

Abstract. The purpose of the study is the program implementation of the basic algorithms of the website design adaptation for people with color perception. This article examines the implementation of the following algorithms: the conversion algorithm from the rgb color model to the lms model (there is a special algorithm precisely in the color model lms for modeling various types of color perception violations), simulation of different color blindness types in the lms-model (this is the transformation of normal values in the color model lms to values with different types of color perception violations), convert data from lms

color model to rgb model, conversion from color rgb to hsl model, color filtering in hsl-model and conversion from hsl model to rgb color model.

Keywords: color-blindness, software engineer, learning programming.

Переважну частину відомостей про навколишнє середовище людина отримує завдяки зоровій системі. Через зір людина сприймає та усвідомлює різноманітні відомості про предмети (розмір, колір, форму, розташування у просторі та ін.). Зір відіграє одну з ключових ролей у більшості видів діяльності, а тому й має величезне значення у житті кожної людини.

Упровадження інформаційних технологій у всі сфери життя людини логічно призвело і до збільшення кількості користувачів мережі Інтернет. Наразі є правилом веб-представництво не лише для комерційних структур, але й для закладів освіти. Враховуючи, що у світі близько 5-8 % чоловіків і 0,5 % жінок мають порушення кольоросприйняття, дуже важливо при проектуванні дизайну сайту враховувати проблеми людей з аномаліями кольоросприйняття.

У попередніх роботах проведено аналіз програмних засобів для людей із порушеннями зору – частковою чи повною сліпотою або ж колірною сліпотою. Зокрема, розглянуто і засоби, що моделюють різні порушення кольоросприйняття [3].

Математичну модель адаптації дизайну сайту для людей з порушеннями кольоросприйняття наведено у [1]. Метою даного дослідження є програмна реалізація основних алгоритмів адаптації дизайну сайту.

Для розробки версії дизайну сайту для людей із порушеннями кольоросприйняття, необхідно спочатку змоделювати різні можливі види порушень. Для цього існує спеціальний алгоритм, розроблений для колірної моделі lms.

Але для подальших розрахунків і відображення результатів необхідно використовувати такі колірні схеми, які наявні у веб-просторі. Для цієї цілі може підійти rgb модель. Тобто, спочатку необхідно виконати конвертацію з колірної моделі rgb до lms моделі.

Функція **rgb2lms()** конвертує колір із колірної моделі rgb у lms модель.

```
function rgb2lms($rgb) {
    $r=$rgb[0];
    $g=$rgb[1];
    $b=$rgb[2];
    $lms_coefficient=[[17.8824, 43.5161, 4.1194],
        [3.4557, 27.1554, 3.8671],
        [0.03, 0.1843, 1.4671]];
    $lms=array();
    foreach($lms_coefficient as $key){
        $value=$key[0]*$r+$key[1]*$g+$key[2]*$b;
        array_push($lms,$value);
    }
}
```

```

    return $lms;
}

```

Наступним кроком необхідно виконати перетворення нормальних значень lms моделі до значень з різними видами дальтонізму.

Функція **protano()** приймає колір, записаний у колірній моделі rgb і імітує колір, який бачитиме людина із протанопією.

```

function protano($rgb) {
    $lms_protano_coeficient=[[0, 2.0234, -2.5258],
                             [0, 1, 0],
                             [0, 0, 1]];

    $lms=rgb2lms($rgb);
    $lms_protano=array();
    foreach($lms_protano_coeficient as $key){
        $value=$key[0]*$lms[0]+$key[1]*$lms[1]+
              $key[2]*$lms[2];
        array_push($lms_protano, $value);
    }
    $rgb_protano=lms2rgb($lms_protano);
    return $rgb_protano;
}

```

Далі необхідно здійснити конвертацію даних з колірної моделі lms до rgb моделі.

Функція **lms2rgb()** конвертує колір із колірної моделі lms у rgb модель.

```

function lms2rgb($lms) {
    $rgb_coeficient=[[0.0809, -0.1305, 0.1167],
                    [-0.0102, 0.0540, -0.1136],
                    [-0.0004, -0.0041, 0.6935]];

    $rgb=array();
    foreach($rgb_coeficient as $key){
        $value=$key[0]*$lms[0]+$key[1]*$lms[1]+
              $key[2]*$lms[2];
        $value=abs($value);
        array_push($rgb, round($value));
    }
    return $rgb;
}

```

Після моделювання необхідних кольорів можна зрозуміти, чи важким для сприйняття буде дизайн майбутнього сайту для людей із порушенням кольоросприйняття.

Принцип фільтрації помилок наступний: якщо html-об'єкти знаходяться близько один біля одного і мають фонові кольори або колір тексту такі, що будуть конфліктувати у людини з порушенням кольоросприйняття, необхідно такі кольори замінити на інші з допустимого спектру.

Для того, аби замінити колір, необхідно звернутися до hsl моделі.

hsl модель має більш логічне представлення кольору, ніж rgb. Дана колірна модель представлена трьома характеристиками: насиченістю, кольоровим тоном та світлотою (яскравістю). Таку колірну модель дуже зручно використовувати для фільтрації спектру, оскільки знаходити протилежний колір спектру не викликає зусиль.

Функція **rgb2hsl()** приймає змінну кольору у вигляді запису колірної моделі rgb та конвертує у hsl модель.

```
function rgb2hsl($rgb){
    $red=$rgb[0];
    $green=$rgb[1];
    $blue=$rgb[2];
    $r=$red/255;//R'
    $g=$green/255;//G'
    $b=$blue/255;//B'
    $c_max=max($r, $g, $b);
    $c_min=min($r, $g, $b);
    $delta=$c_max-$c_min;
    if($delta==0){
        $h=0;
    }
    else{
        switch ($c_max) {
            case $r:
                $h=60*((( $g-$b)/$delta));
                break;
            case $g:
                $h=60*((( $b-$r)/$delta)+2);
                break;
            case $b:
                $h=60*((( $r-$g)/$delta)+4);
                break;
        }
    }
    $l=($c_max+$c_min)/2;
    if ($delta==0) {
        $s=0;
    }
    else {
        if($l<=0.5){
            $s=$delta/($c_max+$c_min);
        }
        else{
            $s=$delta/(2-$delta);
        }
    }
}
```

```

    }
}
$l=round(($l*100)).'%';
$s=round(($s*100)).'%';
$hsl=array($h,$s,$l);
return $hsl;
}

```

Функція **hsl2rgb()** приймає змінну у вигляді кольору колірної моделі hsl та конвертує у rgb модель.

```

function hsl2rgb($hsl){
    $hue=$hsl[0];
    $saturation=$hsl[1];
    $lightness=$hsl[2];
    if(!(($hue>=0) && ($hue<360))){
        throw new Exception("The hue value is out of range".$hue);
    }
    if(!(($saturation>=0) && ($saturation<=1))){
        throw new Exception("The saturation value is out of range");
    }
    if(!(($lightness>=0) && ($lightness<=1))){
        throw new Exception("The lightness value is out of range");
    }
    $h=$hue;
    $s=$saturation;
    $l=$lightness;
    $c=(1-abs(2*$l-1))*$s;
    $x=$c*(1-abs(($h/60)%2)-1);
    $m=$l-($c/2);
    switch ($h) {
        case (($h>=0) && ($h<60)):
            $rgb=array($c,$x,0);
            break;
        case (($h>=60) && ($h<120)):
            $rgb=array($x,$c,0);
            break;
        case (($h>=120) && ($h<180)):
            $rgb=array(0,$c,$x);
            break;
        case (($h>=180) && ($h<240)):
            $rgb=array(0,$x,$c);
            break;
        case (($h>=240) && ($h<300)):
            $rgb=array($x,0,$c);
            break;
    }
}

```

```

    case (($h>=300) && ($h<360)):
        $rgb=array($c,0,$x);
        break;
    }
    $rgb=array((round(($rgb[0]+$m)*255)),
        (round(($rgb[1]+$m)*255)),(round(($rgb[2]+$m)*255)));
    return $rgb;
}

```

На даному етапі не опрацьованими залишаються рисунки та текстури сайтів, тобто, у якості наряду подальших досліджень розглядається конвертація рисунків та текстур.

Список використаних джерел

1. Горло А. М. Адаптація дизайну сайту для людей із порушенням кольоросприйняття / Анна Михайлівна Горло, Ірина Сергіївна Мінтій // Новітні комп'ютерні технології. – Кривий Ріг : Видавничий центр ДВНЗ «Криворізький національний університет», 2018. – Том XVI. – С. 182-187.
2. Горло А. М. Розробка системи адаптації дизайну сайту для людей з порушенням кольоросприйняття [Електронний ресурс] : кваліфікаційна робота з наряду підготовки 6.040302 Інформатика / Горло Анна Михайлівна ; Криворізький державний педагогічний університет. – Кривий Ріг, 2018. – 46 с. – Режим доступу : <http://elibrary.kdpu.edu.ua/handle/0564/2311>.
3. Пірогов В. М. Програмні засоби для людей із порушеннями зору / Пірогов В. М., Мінтій І. С., Мінтій М. М., Вакалюк Т. А. // Актуальні питання сучасної інформатики: Тези доповідей III Всеукраїнської науково-практичної конференції з міжнародною участю «Сучасні інформаційні технології в освіті та науці» (08-09 листопада 2018 р.) / за ред. Т. А. Вакалюк. – Житомир : Вид-во О. О. Євенок, 2018. – Вип. 6. – С. 166-170.

References (translated and transliterated)

1. Horlo, A.M., Mintii, I.S.: Adaptatsiia dyzainu сайtu dlia liudei iz porushenniam kolorospryiniattia (Adapting of website design for people with color blindness). New computer technology. **15**, 182–187 (2018)
2. Horlo, A.M.: Rozrobka systemy adaptatsii dyzainu сайtu dlia liudei z porushenniam kolorospryiniattia (Development of website design adaptation system for people with color blindness). Kvalifikatsiina robota z napriamu pidhotovky 6.040302 Informatyka, Kryvorizkyi derzhavnyi pedahohichniy universytet (2018)
3. Pirohov, V.M., Mintii, I.S., Mintii, M.M., Vakaliuk, T.A.: Prohramni zasoby dlia liudei iz porushenniamy zoru (Software for people with visual impairment). In: Vakaliuk, T.A. (ed.) Aktualni pytannia suchasnoi informatyky, Tezy dopovidei III Vseukrainskoi naukovo-praktychnoi konferentsii z mizhnarodnoiu uchastiu «Suchasni informatsiini tekhnolohii v osviti ta nauksi» (08-09 lystopada 2018 r.), vyp. 6, pp. 166–170. Vyd-vo O. O. Yevenok, Zhytomyr (2018)

Актуальність розробки моделі адаптивного навчання

Слизова Юрійвна Тищенко, Андрій Миколайович Стрюк^[0000-0001-9240-1976]

Криворізький національний університет,
вул. Віталія Матусевича, 11, м. Кривий Ріг, 50027, Україна
{liza.tishenko, andrey.n.stryuk}@gmail.com

Анотація. Процес навчання можна зробити найбільш ефективним за рахунок переносу освітнього процесу в електронне середовище. Завдяки адаптивному тестуванню підвищується точність, якість, достовірність навчання та зацікавленість студента, яка дозволяє йому бути більш вмотивованим. Це новий підхід для засвоєння студентом більшої частини інформації. Введення системи адаптивного тестування гарантує покращення показників засвоєння матеріалу студентами. Від правильної організації контролю знань залежить ефективність системи навчально-виховного процесу. Адаптивне тестування передбачає зміну послідовності завдань у самому процесі тестування з урахуванням відповідей на вже отримані завдання. У процесі проходження тесту будується модель особи, яка навчається, для подальшого використання при виборі наступних завдань тестування залежно від рівня знань студента та його індивідуальних особливостей. При обчисленні оцінки адаптивна система тестування враховує ймовірність, що студент може вгадати відповідь, кількість спроб пройти тест і середній результат, досягнений за час усіх спроб. Комплекс завдань для адаптивного тестування може бути розроблений з урахування окремого виду сприйняття інформації кожним учнем, тобто студенту пропонуються завдання, з якими він в змозі впоратися і які цікаві саме йому, а значить він більш впевнений в своїх силах і націлений на успішне закінчення курсу.

Ключові слова: освіта, адаптивне навчання, адаптивне тестування.

The relevance of developing a model of adaptive learning

Yelyzaveta Yu. Tyshchenko and Andrii M. Striuk^[0000-0001-9240-1976]

Kryvyi Rih National University, 11, Vitalii Matusevych St., Kryvyi Rih, 50027, Ukraine
{liza.tishenko, andrey.n.stryuk}@gmail.com

Abstract. The learning process can be made most effective by transferring the educational process to the electronic environment. Thanks to adaptive testing, the accuracy, quality, reliability of training and student interest are enhanced, which

allows him to be more motivated. This is a new approach for the student to master most of the information. The introduction of an adaptive testing system ensures the improvement of student learning performance. From the proper organization of the control of knowledge depends on the effectiveness of the educational process. Adaptive testing involves changing the sequence of tasks in the testing process itself, taking into account the answers to the tasks already received. In the process of passing the test, a personality model is built that learns for later use in selecting the following testing tasks, depending on the level of knowledge of the student and his individual characteristics. When calculating the assessment, the adaptive testing system takes into account the probability that the student can guess the answer, the number of attempts to pass the test and the average result achieved during all attempts. The complex of tasks for adaptive testing can be developed taking into account a separate type of perception of information by each student, that is, the student is offered tasks that he is able to cope with and which are interesting for him, which means he is more confident in his abilities and aims at successful completion of the course.

Keywords: education, adaptive learning, adaptive testing.

У процесі інформатизації суспільства, системи освіти, розповсюдження дистанційного навчання, розвитку технологій управління персоналом актуалізуються завдання розробки автоматизованих інформаційних систем, що дозволяють швидко, об'єктивно та якісно оцінювати знання тих, хто навчається. У сучасному освітньому просторі функціонує багато форм, способів та методів контролю знань, вмінь і навичок студентів. Від правильної організації контролю знань залежить ефективність як системи навчально-виховного процесу, так і якість підготовки фахівця. Традиційним формам контролю рівня підготовки студентів притаманні суттєві недоліки, усунення яких досягається шляхом створення сучасних автоматизованих систем контролю знань [1].

Адаптивне навчання полягає в тому, що індивідуалізований метод навчання допоможе студенту вчитися швидше, ефективніше та з більшим розумінням. Зазвичай компоненти адаптивного навчання містять: моніторинг активності, інтерпретацію результатів, розуміння вимог та переваг від вивчення нових тем для полегшення процесу навчання. Головна мета адаптивного навчання – зробити процес навчання найбільш ефективним за рахунок переносу освітнього процесу в електронне середовище [2]. Завдяки адаптивному тестуванню підвищується точність, якість, достовірність навчання та зацікавленість студента, яка дозволяє йому бути більш вмотивованим. Студенту пропонуються завдання, з якими він в змозі впоратися і які цікаві саме йому, а значить він більш впевнений в своїх силах і націлений на успішне закінчення курсу.

На сьогоднішній день упровадження адаптивного навчання займає провідну роль ще й тому, що була проведена реформа освіти. Згідно концепції «Нової української школи» забезпечується обов'язкове врахування інтересів кожного учасника навчального процесу, врахування його вікових та індивідуальних особливостей розвитку, а також морально-психологічного комфорту кожного учня. І адаптивне навчання за своєю суттю спрямоване саме на реалізацію

індивідуальних запитів учня, тобто йде йому назустріч, підлаштовується і пристосовується під нього. У буквальному перекладі слово «адаптація» розуміється як процес пристосування.

У процесі впровадження адаптивного навчання можна виділити ряд значних переваг. У першу чергу, кожному студенту добирається зручний темп навчання та засвоєння конкретного матеріалу, а це може значно прискорити процес оволодіння новою інформацією. Наступною перевагою адаптивного навчання є об'єктивність результатів навчання та оцінювання кінцевого результату. При розробці методичного комплексу завдань встановлюється єдина система оцінювання, що дає змогу зробити процес навчання неупередженим. Неабиякою перевагою адаптивного навчання є те, що комплекс завдань може бути розроблений з урахування окремого виду сприйняття інформації кожним студентом.

У сучасних інформаційних навчальних системах проблема адаптивного навчання розглядається у двох аспектах: методичному і технічному. До методичних аспектів адаптивного навчання в інформаційних навчальних системах відносять планування і організацію навчального процесу, визначення типів завдань, рівнів їх складності, послідовності подачі матеріалу, проведення різних видів контролю, визначення критеріїв оцінки кожного виду завдання. До технічних аспектів відносяться: алгоритм, який пропонує перейти на новий рівень при правильному виконанні більшої частини задач або повернутися на попередній рівень з урахуванням помилок, допущених при виконанні завдань; алгоритм оцінювання навчальних досягнень студентів, тощо [3].

Метою навчання завжди було засвоєння студентами матеріалу, тобто найголовніша ціль кожного заняття – це залишити у голові студента якомога більше знань. Але мозок людини не запам'ятовує отримані знання в повній мірі, про це свідчить крива забування німецького психолога Германа Еббінгауза. Вона показує, як довго людина пам'ятає нову інформацію та який її відсоток людина забуває і як швидко. В першу годину людина можна згадати до 60 % інформації, що отримала. Через годину – лише 45 %, а після одного дня – 34 %. Через тиждень після вивчення нового людина пам'ятає до 23 %; і лише 15 % того, що було вивчено, міцно зберігається в пам'яті. Але кожен раз, намагаючись пригадати інформацію, людина «зав'язує вузлик в пам'яті». Чим їх більше – тим краще запам'ятовується матеріал. Нейронний шлях стає щільніше і потужніше – інформація залишається надовго [4].

Наукові дослідження доводять, що вправи із пригадування зміцнюють знання набагато краще, ніж повторне ознайомлення з матеріалом. Дослідження когнітивних наук показують, що тестування може бути ефективним способом навчання, якщо правильно його використовувати. Воно дає найкращий спогад фактів і більш глибоке розуміння, ніж навчання, засноване тільки на підсумкових іспитах. Повторне тестування дає значний позитивний ефект. Результати досліджень демонструють критичну роль тестування в навчанні [5].

Спрацьовує «ефект тестування», який називають ще ефектом пригадування. Пригадування – це потужний інструмент навчання. Згадуючи будь-яку інформацію, ми спрощуємо собі завдання не забути її пізніше [6].

Саме тому проведення тестування на кожному занятті є дуже актуальним, а введення системи адаптивного тестування гарантує покращення показників засвоєння матеріалу студентами.

Застосування у закладах вищої освіти автоматизованих систем тестування з метою оцінювання знань має низку переваг, а саме:

- об'єктивність результатів перевірки, у тому числі за рахунок елімінації людського фактору;
- можливість одночасного тестування великої кількості студентів;
- підвищення ефективності контролю знань за рахунок регулярності тестування;
- можливість автоматизованої перевірки результатів тестування, скорочення часу перевірки відповідей на велику кількість завдань;
- отримання достовірної інформації щодо обсягу і рівня засвоєння навчального матеріалу студентами;
- використання великої кількості запитань, якими можна охопити дисципліну загалом;
- можливість статистичної та порівняльної характеристики результатів перевірки у межах навчальної групи або окремого студента.

Адаптивно-навчальні системи функціонують у контексті сучасного підходу до технології контролю знань, запропонованого Л. А. Растригіним та М. Х. Еренштейном, відповідно до якого відносини між учнем і вчителем подаються як відносини між об'єктом керування і керувальним пристроєм [4]. Адаптивне тестування – це широкий клас методик тестування, які передбачають зміну послідовності завдань у самому процесі тестування з урахуванням відповідей студента на вже отримані завдання. У процесі проходження тесту будується модель особи, яка навчається, для подальшого використання при виборі наступних завдань тестування залежно від рівня знань студента [8]. Крім того, адаптивне тестування враховує ймовірність, що студент може вгадати відповідь. Якщо, наприклад, на запитання пропонується чотири варіанти відповіді, то ймовірність «вгадування» складає $1/4$. У такому разі, якщо межа для оцінки задовільно встановлена 30 %, то студенту необхідно знати лише 5 % правильних відповідей. Якщо з 100 балів необхідно набрати 24, то їх можна вгадати, про що доповідають фахівці центрів незалежного тестування [9]. А статистичний аналіз результатів адаптивного тесту проводиться виходячи з кількості спроб пройти тест і середнього результату, досягнутого за час всіх спроб. Такий аналітичний підхід корисний викладачеві в першу чергу для поліпшення якості навчальних матеріалів, які студент проходить на всіх етапах його навчання.

Основною особливістю адаптивної системи навчання є оптимізація процесу навчання. На сьогоднішній день не існує універсальної методики навчання, тому викладач вибирає найбільш прийнятний спосіб навчання, виходячи з власного досвіду, що не завжди буває оптимально. Тут в якості критерію ефективності ми розглядаємо глибину освоєння предмета учнем, повноту і міцність засвоєних ним знань, рівень вивчення теоретичного матеріалу та набуття практичних навичок.

На сьогоднішній день штучні нейронні мережі широко застосовуються в різних сферах інформаційних технологій [10], проте у вітчизняній освіті технології штучного інтелекту майже не використовуються. Виявлення паттернів поведінки користувачів (осіб, які підлягають тестуванню) дозволяє застосовувати апарат штучних нейронних мереж в тестуванні для вирішення проблем, пов'язаних з удосконаленням контрольно-вимірвальних матеріалів тестів, сприяючи підвищенню достовірності результатів тестування.

Нейронні мережі є адаптивними системами, вони отримують інформацію з реальних процесів, які моделювати динамічно важко. Застосування апарату нейронних мереж дозволяє вирішувати завдання, які важко або неможливо вирішувати традиційними методами [11]. Так, моделі в процесі роботи накопичують інформацію, і ефективність їх з часом зростає. Використання нейронних мереж дозволить зробити тестування більш об'єктивним і розширити його застосування.

Список використаних джерел

1. Семеріков С. О. Побудова найпростішої системи тестового контролю знань на основі Web-технологій / Семеріков С. О., Теплицький І. О. // Науковий часопис Національного педагогічного університету імені М. П. Драгоманова. – Серія № 2. Комп'ютерно-орієнтовані системи навчання : зб. наукових праць / Редада. – К. : НПУ імені М. П. Драгоманова, 2004. – №1 (8) – С. 106-116.
2. Моїсеєнко Н. В. Мобільне інформаційно-освітнє середовище вищого навчального закладу / Моїсеєнко Н. В., Моїсеєнко М. В., Семеріков С. О. // Вісник Черкаського університету. Серія «Педагогічні науки». – 2016. – № 11. – С. 20-27.
3. Борогов В. В. Психолого-педагогические основы системы адаптивного обучения / В. В. Борогов // Наука и школа. – 2001. – № 2. – С. 12-15.
4. Semerikov S. O. Computer Simulation of Neural Networks Using Spreadsheets: The Dawn of the Age of Camelot [Electronic resource] / Serhiy O. Semerikov, Illia O. Teplytskyi, Yuliia V. Yechkalo, Arnold E. Kiv // Augmented Reality in Education : Proceedings of the 1st International Workshop (AREdu 2018). Kryvyi Rih, Ukraine, October 2, 2018 / Edited by : Arnold E. Kiv, Vladimir N. Soloviev. – P. 122-147. – (CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2257). – Access mode : <http://ceur-ws.org/Vol-2257/paper14.pdf>.
5. Karpicke J. D. The Critical Importance of Retrieval for Learning / Jeffrey D. Karpicke, Henry L. Roediger III. – Science. – 2008. – Vol. 319. – Issue 5865. – P. 966-968. – DOI : 10.1126/science.1152408.
6. Brown P. C. Make It Stick: The Science of Successful Learning / Peter C. Brown, Henry L. Roediger III, Mark A. McDaniel. – Cambridge : Belknap Press, 2014. – 336 p.
7. Растрингин Л. А. Адаптивное обучение с моделью обучаемого / Л. А. Растрингин, М. Х. Эренштейн ; Риж. политехн. ин-т им. А. Я. Пельше. – Рига : Зинатне, 1988. – 160 с.
8. Кравченко О. М. Модель інтелектуальної контролюючої підсистеми з багаторівневим адаптивним тестуванням / О. М. Кравченко, Ж. М. Плаасова // Східно-Європейський журнал передових технологій. – 2010. – № 4/2 (46). – С. 21-25.
9. Катеринчук І. С. Інтелектуальна система автоматизованого контролю знань студентів вищих навчальних закладів / Катеринчук І. С., Кравчук В. В., Кулик В. М., Рачок Р. В.

// Інформаційні технології в освіті. – Вип. 4. – Херсон : Вид-во ХДУ, 2009. – С. 139-147. – DOI : 10.14308/ite000100.

10. Пермякова О. С. Застосування нейронних мереж у задачах прогнозування / О. С. Пермякова, С. О. Семеріков // Молодий науковець XXI століття : матеріали Міжнародної науково-практичної конференції (Кривий Ріг, 17–18 листопада 2008 р.). – Кривий Ріг : Видавничий центр КТУ, 2008. – С. 237-239.
11. Markova O. M. CoCalc as a Learning Tool for Neural Network Simulation in the Special Course “Foundations of Mathematic Informatics” [Electronic resource] / Oksana Markova, Serhiy Semerikov, Maiia Popel // ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer 2018 : Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops (ICTERI, 2018). Kyiv, Ukraine, May 14-17, 2018 / Edited by : Vadim Ermolayev, Mari Carmen Suárez-Figueroa, Vitaliy Yakovyna, Vyacheslav Kharchenko, Vitaliy Kobets, Hennadiy Kravtsov, Vladimir Peschanenko, Yaroslav Prytula, Mykola Nikitchenko, Aleksander Spivakovsky. – P. 388-403. – (CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2104). – Access mode : http://ceur-ws.org/Vol-2104/paper_204.pdf.

References (translated and transliterated)

1. Semerikov, S.O., Teplytskyi, I.O.: Pobudova naiprostishoi systemy testovoho kontroliu znan na osnovi Web-tekhnologii (Building a simplest system of knowledge testing on Web-based technologies). Naukovyi chasopys Natsionalnoho pedahohichnoho universytetu imeni M. P. Drahomanova, Seriya 2: Kompiuterno-orientovani systemy navchannia. 1(8), 106–116 (2004)
2. Moiseienko, N.V., Moiseienko, M.V., Semerikov, S.O.: Mobilne informatsiino-osvitnie seredovyshe vyshchoho navchalnoho zakladu (The mobile information and educational environment of higher educational institution). Cherkasy University Bulletin: Pedagogical Sciences. 11, 20–27 (2016)
3. Borogev, V.V.: Psikhologo-pedagogicheskie osnovy sistemy adaptivnogo obuchenii (Psychological and pedagogical foundations of the system of adaptive learning). Nauka i shkola. 2, 12–15 (2001).
4. Semerikov, S.O., Teplytskyi, I.O., Yechkalo, Yu.V., Kiv, A.E.: Computer Simulation of Neural Networks Using Spreadsheets: The Dawn of the Age of Camelot. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 122–147. <http://ceur-ws.org/Vol-2257/paper14.pdf>. Accessed 30 Nov 2018
5. Karpicke, J.D., Roediger III, H.L.: The Critical Importance of Retrieval for Learning. Science. **319**(5865), 966–968 (2008). doi:10.1126/science.1152408
6. Brown, P.C., Roediger III, H.L., McDaniel, M.A.: Make It Stick: The Science of Successful Learning. Belknap Press, Cambridge (2014)
7. Rastrigin, L.A., Erenshtein, M.Kh.: Adaptivnoe obuchenie s modeliu obuchaemogo (Adaptive learning with a student model). Zinatne, Riga (1988)
8. Kravchenko, O.M., Plakasova, Zh.M.: Model intelektualnoi kontroliuichochoi pidsystemy z bahatorivnevym adaptivnym testuvanniam (Model of the intellectual subsystem of knowledge control with multi-level adaptive testing). Eastern-European Journal of Enterprise Technologies. **4**(2), 21–25 (2010)

9. Katerynychuk, I. S., Kravchuk, V. V., Kulyk V. M., Rachok, R. V. Intelktualna systema avtomatyzovanoho kontroliu znan studentiv vyshchych navchalnykh zakladiv (Intelligent system of automated knowledge control of students of higher educational institutions). *Information technologies in education*. 4, 139–147 (2009). doi:10.14308/ite000100
10. Permiakova, O.S., Semerikov, S.O.: Zastosuvannia neironnykh merezh u zadachakh prohozuvannia (Application of neural networks in forecasting tasks). In: *Molodyi naukovec XXI stolittia, materialy Mizhnarodnoi naukovo-praktychnoi konferentsii, Kryvyi Rih, 17–18 lystopada 2008 r.*, pp. 237–239. Vydavnychi tsestr KTU, Kryvyi Rih (2008)
11. Markova, O., Semerikov, S., Popel, M.: CoCalc as a Learning Tool for Neural Network Simulation in the Special Course “Foundations of Mathematic Informatics”. In: Ermolayev, V., Suárez-Figueroa, M.C., Yakovyna, V., Kharchenko, V., Kobets, V., Kravtsov, H., Peschanenko, V., Prytula, Y., Nikitchenko, M., Spivakovsky, A. (eds.) *Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops (ICTERI, 2018), Kyiv, Ukraine, May 14-17, 2018. CEUR Workshop Proceedings, vol. 2104, pp. 388–403.* http://ceur-ws.org/Vol-2104/paper_204.pdf. Accessed 25 Aug 2018

Розробка програмного забезпечення для мінімізації витрат часу та підвищення продуктивності праці у сфері комунікаційних послуг

Дмитро Олександрович Курганов, Альберт Арамаїсович Азарян^[0000-0003-0892-8332]

Криворізький національний університет,
вул. Віталія Матусевича, 11, м. Кривий Ріг, 50027, Україна
{gag7023, azaryan325}@gmail.com

Анотація. Стаття присвячена дослідженню методів ефективності надання консультаційних послуг та розробці програмного забезпечення, що їх реалізує. В процесі дослідження визначено, що сучасні засоби автоматизації підвищують ефективність та коректність надання консультативних послуг. Мінімізація витрат часу та підвищення продуктивності праці в сфері комунікаційних послуг можливі за наступних умов: підвищення швидкості опрацювання питань клієнтів за рахунок вибору оптимальної методики пошуку відповіді; зменшення кількості помилок за рахунок використання шаблонів відповідей. Найчастіше для автоматизації консультативних послуг застосовуються системи самообслуговування абонентів за допомогою інтерактивних голосових меню (IVR). У роботі запропоновані заходи, що надають можливість знизити трафік в подібних системах. Розроблене програмне забезпечення може спрогнозувати питання, яке необхідно опрацювати, що пришвидшить роботу контактного центру та підвищить його ефективність. Експериментальні дослідження підтвердили підвищення швидкості опрацювання даних, зниження трафіку за рахунок використання розробленої системи.

Ключові слова: автоматизація, програмне забезпечення, кол-центр, телефонія, IVR.

Software development to minimize time costs and increase productivity in the area of communication services

Dmytro O. Kurhanov and Albert A. Azaryan^[0000-0003-0892-8332]

Kryvyi Rih National University, 11, Vitalii Matusevych St., Kryvyi Rih, 50027, Ukraine
{gag7023, azaryan325}@gmail.com

Abstract. The article is devoted to the research of methods of providing advisory services and development of software that implements them. In the course of the research, it was determined that modern automation tools increase the efficiency and correctness of providing advisory services. Minimization of time expenditures and increase of labor productivity in the field of communication services is possible under the following conditions: increasing the speed of processing clients' questions by choosing an optimal method for finding a response; reducing the number of errors due to the use of response templates. Most often, automation of advisory services uses self-service subscribers with the help of interactive voice response (IVR). The paper proposes measures that make it possible to reduce traffic in such systems. The developed software can predict the issue that needs to be worked out, which will speed up the work of the contact center and increase its efficiency. Experimental studies have confirmed the speed of processing data, reducing traffic due to the use of the developed system.

Keywords: automation, software, call center, telephony, IVR.

1 Вступ

Організація ефективного та коректного отримання консультативних послуг є на сьогодні актуальною суспільною проблемою. На початку ХХІ століття була запропонована система консультування за допомогою «гарячих ліній». Певний час не виникало складнощів щодо роботи консультантів, тому що на той момент звернень було дуже мало через невелику кількість комунікаційних пристроїв. На сучасному етапі системи інформування почали розвиватися у напрямі веб-технологій, але «голосовий» спосіб інформування залишається актуальним через те, що ступінь довіри щодо почутої інформації від людини більший, ніж ступінь довіри щодо інформації, яка відображається на екрані комп'ютера. Також кількість комунікаційних пристроїв збільшилась, що підвищило кількість звернень за допомогою телефону. Таким чином, інформування на «гарячих лініях» потребує автоматизації, що обумовлює актуальність розробки

програмного забезпечення для мінімізації витрат часу і підвищення продуктивності праці в сфері комунікаційних послуг.

2 Визначення вимог до програмного забезпечення

Типовим прикладом організації або підрозділу організації, що надає комунікаційні послуги широкому колу користувачів, є кол-центр (call-centre). Різні компанії використовують центри з опрацювання дзвінків для спілкування з клієнтами, організації та супроводу продажу товарів та послуг, організації сервісного обслуговування, технічної підтримки.

Мінімізація витрат часу та підвищення продуктивності праці в сфері комунікаційних послуг можливі за наступних умов:

1. Підвищення швидкості опрацювання питань клієнтів за рахунок вибору оптимальної методики пошуку відповіді.
2. Зменшення кількості помилок за рахунок використання шаблонів відповідей.

Для правильного формулювання відповіді система повинна використовувати дані з бази даних, а саме: особисті дані клієнта (рахунки, регламенти, ідентифікуюча інформація тощо) та заздалегідь підготовлені голосові модулі з «мітками» для підстановки динамічного тексту.

Для правильного підбору тематик необхідно проаналізувати критичні теми, тобто усі тематики звернення об'єднати в певні групи за спільними рисами. Після цього необхідно проаналізувати, яка саме інформація цікавить клієнта найбільше.

3 Дослідження сфери застосування

Для організації комунікацій можуть бути використані як телефонні мережі загального користування, так і IP-телефонія [1]. Часто застосовуються системи самообслуговування абонентів за допомогою інтерактивних голосових меню (IVR – Interactive voice response) [2]. За рахунок автоматизації телекомунікаційних мереж ми можемо зменшити кількість звернень та час діалогів, що разом утворюють так званий «трафік». Щоб знизити трафік, було запропоновано схему на основі стандартної схеми автоматизації (рис. 1). Перед та після події повинна пройти передфільтрація та постфільтрація. За допомогою цього можна знизити не тільки трафік на лініях контактної служби, а й зберегти дані так, щоб з часом, при необхідності, аналітики могли отримати дані та проаналізувати причину поточного трафіку, виявити вразливі місця та виправити ситуацію в моменти технічних складностей. На рис. 1 можна побачити, що постфільтрація має вплив на передфільтрацію, так як після створення контакту клієнта з операторами банку результати можуть дати вплив на дані клієнта й навіть на події, які є керуючими чинниками в процесі передфільтрації.

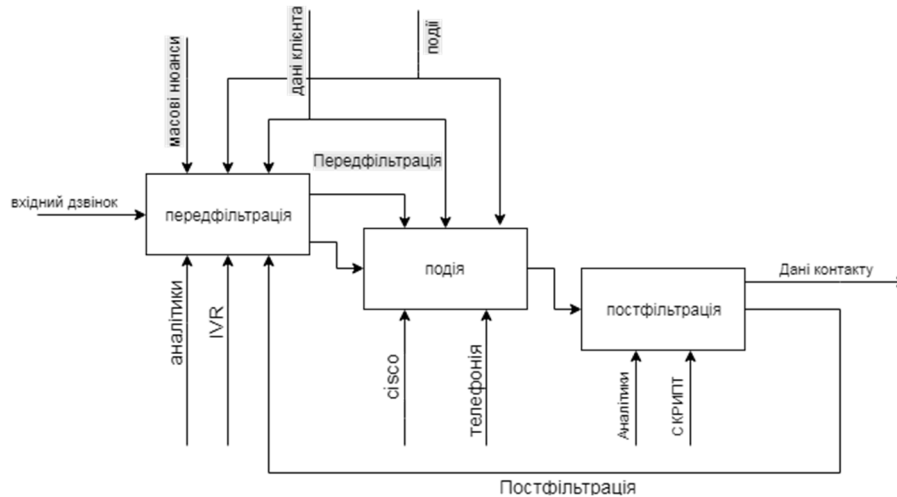


Рис. 1. Загальна схема автоматизації обробки вхідних дзвінків

Як було вказано вище, передфільтрація впливає на швидкість вирішення питання, кількість звернень та якість обслуговування. Деякі питання можна передбачити і навіть вирішити без зв'язку з оператором, тому, залежно від поставленої задачі, була створена така передфільтрація, яка включає в себе наступні пункти:

- сегментація за типом клієнта;
- сегментація за типом операцій;
- сегментація за статусом клієнта;
- сегментація за масовими складнощами.

Для передфільтрації майже всі організації, які мають «гарячу лінію» використовують клієнтське IVR меню.

IVR – система попередньо записаних голосових повідомлень, що виконує функцію маршрутизації дзвінків всередині кол-центру, користуючись інформацією, що вводиться клієнтом на клавіатурі телефону за допомогою тонального набору.

Для виконання даного проекту була взята за основу IVR-система банку «ПриватБанк». У ході експерименту (табл. 1) вдалося виявити відсоток людей, які вирішують своє питання в селф-сервісі, визначити, як за допомогою голосового меню розрізнити тип клієнта, для маршрутизації до різних операторів, використати дані зі створеної телефонії для формування динамічного інформування, визначити, який проміжок часу є у сервера для опрацювання інформації клієнта та виявлення впорядкованого списку з тематиками.

Під час експерименту було виявлено, що банк зберігає в системі інформацію про взаємодію банку з клієнтом не тільки на рівні операцій, а й на рівні отримання дзвінків, так як в деяких випадках встановлено статус клієнта після звернення (клієнт-детрактор), що підтверджує те, що інформаційна база для створення інтелектуальної телефонії існує, впровадження даної системи не вибагливе та не

потребуватиме додаткових витрат при провадженні нового програмного забезпечення.

Таблиця 1. Результати експерименту з голосовим меню

№	Подія	Заплановане питання	Дії під час процесу	Опис
1	Дзвінок після входу в клієнт-банк	Інформація по клієнт-банку	Стандартна будова голосового меню	Однією із функцій у створюваній телефонії є відстеження входу в клієнт-банк, ураховуючи те, що голосове меню не може відстежити причину звернення після входу в клієнт-банк, підтверджує необхідність інтелектуальної телефонії
2	Дзвінок після відправлення платежу за хибними реквізитами	Інформація щодо платежу	Прослуховування врізки з можливістю з'єднання з компетентним спеціалістом	Голосове меню змогло повідомити клієнта про повернення платежу з можливістю повернення коштів або уточнення реквізитів
3	Дзвінок після проведення операції в ТСО	Інформація щодо платежу	Прослуховування врізки	Врізка виникає у випадку помилок роботи системи або неуспішності платежу за рядом причин
4	Дзвінок при помилці проведення транзакції в Інтернет	Причина помилки	Інформація про причину неуспішної транзакції	Голосове меню змогло ідентифікувати причину звернення, що прискорить обслуговування клієнта
5	Дзвінок після введення некоректного ПІН коду	Причина помилки	Врізка про ПІН код	Голосове меню змогло ідентифікувати причину звернення, що прискорить обслуговування клієнта
6	Дзвінок після блокування клієнт-банку	Причина блокування	Врізка про причини блокування	Дана інформація допоможе виявити причину звернення і підготувати варіанти вирішення питання
7	Дзвінок як клієнт-детрактор	Причина	Обмеження дозвону	Не потребує вдосконалення
8	Дзвінок при провадженні платежів з клієнт-банку та з помилкою в даному сервісі	Причина помилки	Озвучення причини	Людський фактор при відображенні ситуації

4 Проектування програмного забезпечення

У процесі дослідження було створено функціональну схему голосового меню, з яким взаємодіє клієнт (рис. 2).

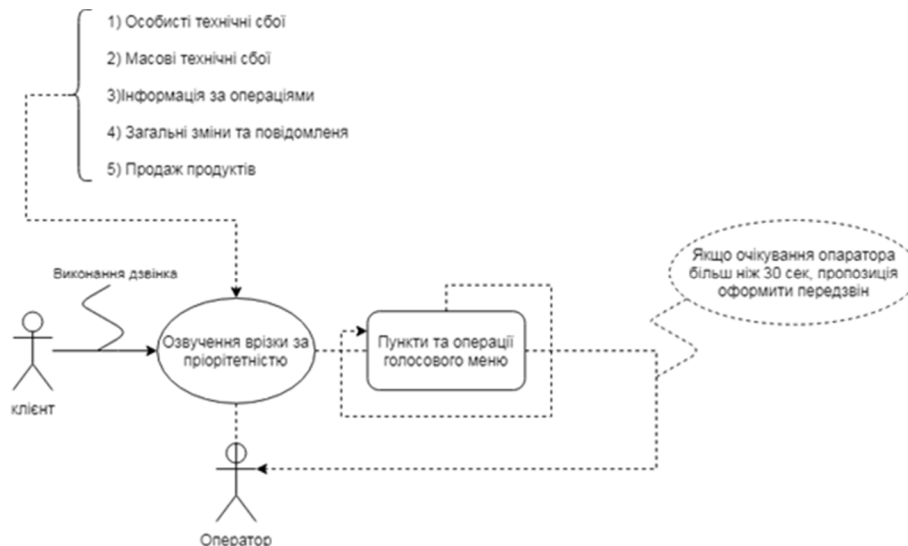


Рис. 2. Операції, які проводить передфільтрація до зв'язку з оператором

Як ми можемо бачити на рис. 2, до зв'язку з оператором клієнт може перебувати в чотирьох умовах:

- термінова подія;
- повне голосове меню;
- реєстрація CallBack;
- з'єднання з прямим номером.

Звертаємо увагу, що перші три умови доцільні тільки при виконаній четвертій умові, так як при дзвінку на номер 3700 вибраний об'єкт інформує, що перетелефонує протягом трьох хвилин. Таким чином банк знижує навантаження на мережу та дозволяє завантажити усі необхідні дані заздалегідь.

У першому випадку умова виконується вже при завантаженій інформації, а це означає, що часу на завантаження даних клієнта не потребується, так як дана подія вже виконана.

У другому випадку система має мінімум сорок три секунди для завантаження даних по одному клієнту. Дане число була визначено експериментально та є мінімальним, тому що, окрім голосового меню, клієнт може очікувати в «черзі» на зв'язок із оператором.

У третьому випадку, урахувавши те, що банк буде передзвонювати клієнту, інформація також буде завантажена заздалегідь.

Залежно від того, який клієнт звертається, буде залежати швидкість та якість обслуговування. Наприклад, консультант, який пріоритетно обслуговує фізичних осіб, не зможе дати якісну консультацію фізичній особі-підприємцю, та навіть, якщо консультацію консультант дати зможе, то швидкість опрацювання питання буде значно менша, ніж у спеціаліста, який вирішує переважно питання бізнес-клієнтів.

Тому залежно від того, який «статус» клієнта встановлений, з такою лінією він буде з'єднаний. Однією із властивостей телефонії є можливість відсортувати список залежно від звернення. Якщо звертається, наприклад, бізнес-клієнт, маючи спільну телефонію для фізичних осіб та для корпоративного сектору, можна відсортувати тематики так, що для корпоративного сектору буде відображатися пріорітетно тематики корпоративного сектору, але, якщо в ході розмови клієнту знадобиться інформація щодо іншої тематики, правильно вибравши тематику, навіть недосвідчений оператор може надати інформацію, зачитавши її з динамічного поля з інформацією (надалі «врізка»). Таким чином оператор надасть не тільки більш якісну послугу, а й знизить трансфери на іншого оператора, що знизить кількість дзвінків, що є дуже важливим параметром при розрахунку трафіку банку.

Також наявні такі типи клієнтів, яким банк обмежує доступ до операторів. Наприклад, клієнти-детрактори, жартівники чи шахраї.

Якщо клієнт проводить шахрайські дії, то банк припиняє роботу з даним клієнтом. Ця задача включена в передфільтрацію та створена телефонія (при описаній чи створеній логіці) ігнорує дзвінок даного клієнта. Клієнту-шахраю в такому випадку озвучується інформація в голосовому меню. Інформація щодо шахрая контролюється банком, але не входить в «обов'язки» інтелектуальної телефонії. Дана інформація вплинула на конструкцію бази даних при проектуванні та конструюванні інформаційної бази для обслуговування інтелектуальної телефонії.

Клієнт-шахрай або клієнт-жартівник не може додзвонитися на лінію певний час, але йому не обмежується дозвін в цілому. Це означає, що в базі даних ми повинні зберігати не тільки статус клієнта, а й час, коли статус було встановлено.

Такими діями можна вберегти організацію від марних витрат коштів у випадках, коли клієнти використовують гарячу лінію не за призначенням.

У даному проекті передфільтрація за подією відіграє дуже важливу роль, тому що у клієнта найчастіше виникає питання, коли клієнт робить певну дію та в клієнта щось не виходить, або подія пов'язана з певним календарним планом.

Для виявлення основних подій було проведено експеримент.

Головною задачею є саме пришвидшення опрацювання питання клієнтів та прогнозування питання ще до відповіді на виклик, тому наведемо приклади розрахунків, пов'язаних із оптимізацією процесу обслуговування.

Так як функціональний шаблон був побудований на банківській справі, аналітична робота проводилася щодо консультування клієнтів в саме цій сфері.

Розрахунок кількості та причин звернень проводився у декілька етапів:

- збір даних;
- знаходження чинників, які впливали на отриманий результат;
- вибір «критичних» днів звернення;
- створення ієрархічної послідовності.

На першому етапі проводився збір даних, а саме: підрахунок загальної кількості звернень, вибір тематики звернень. Перший етап проводився протягом 25 днів.

На другому етапі було проаналізовано отримані дані та знайдена закономірність причин звернень. Отриману інформацію було проаналізовано та визначено закономірність щодо днів, де дзвінків було більше. Наприклад, першого числа основні звернення були щодо списань, так як першого числа списуються відсотки щодо використання кредитних коштів. Тобто, були виведені фактори, які впливали на причину звернення, а саме: заборгованість за картою та списання деякої суми. Ця інформація була необхідна для створення інформаційного забезпечення даної роботи.

На третьому етапі було також проаналізовано інформацію і був створений певний статичний список з «критичними днями». У даний список увійшли всі дні, в які телефонують найчастіше незалежно від статусу клієнта, а саме:

- перше число місяця – списання відсотків;
- друге число місяця – списання відсотків;
- третє число місяця – нарахування пені;
- п'ятнадцяте число місяця – обзвін щодо простроченої заборгованості;
- двадцять п'яте число місяця – мінімальний платіж.

Дані «критичні дні» будуть піднімати деякі тематики на вершину ієрархічного списку, але лише в тому випадку, коли немає інших чинників, локальних, які мають більшу цінність, ніж подія, яка може виникнути за специфікацією дати.

Для покращення специфікації за датою доцільно використовувати сторонні чинники, такі як наявність заборгованості чи простроченої заборгованості, тому що клієнт навряд чи буде звертатися щодо мінімального платежу, якщо у нього немає заборгованості. Також календарні чинники будуть розширятися залежно від віку чи зовнішнього чинника, такого як зарплатний проект. Створити базу даних з датами отримання зарплат не дуже важко, але лише у випадку, коли роботодавець згодний надати інформацію щодо офіційних дат зарплат. Це не лише підвищує швидкодію системи, а й надає можливість отримувати інформацію щодо затримок заробітних плат. Якщо банк буде мати інформацію щодо затримки заробітної плати, то це значно підвищить швидкодію контактного центру та знизить рівень невдоволеності клієнтів.

Отже, проаналізувавши «критичні дні», створені за допомогою аналізу зібраної інформації, можна сказати, що дана інформація буде ефективна в тандемі з інформаційним чинником, створюючи при цьому певні «пари», такі як «дата нарахування відсотків/заборгованість» чи «дата отримання пенсії/вік людини» тощо.

На четвертому етапі було пов'язано найпопулярніші тематики звернень та чинники, через які вони можуть виникати. Було створено «верхівку ієрархічної таблиці», яка включає найбільш видимі зв'язки між тематиками та чинниками звернень.

У наступному списку нижче наведено ряд тематик, які мають найбільші пріоритети при виконанні деяких чинників:

- Інтернет-ліміт;
- успішність операції в ТСО;

- успішність операції в АТМ;
- заявка на повернення в АТМ;
- повернення коштів ТСО/АТМ;
- рух коштів/списання;
- підвищення кредитного ліміту;
- підвищення кеш-ліміту;
- зняття блоку на подвійну конвертацію;
- успішність платежу через Інтернет;
- причина відсутності доступу до особистого кабінету.

Всі інші тематики, за якими проводилися дослідження, мали не тільки чинники, а й дату, до яких ці чинники прив'язані. У таблиці 2 наведені приклади з коефіцієнтами.

Таблиця 2. Сортування за датою в місяці

№ дня	1	2	5	15	20	22	24	25	30
Баланс карт. Очікування зарахувань	X	X	1	X	2	1	5	3	3
Операції АТМ/ТСО	4	3	4	5	1	2	3	2	4
Рух за картою. Списання	1	1	2	2	4	3	4	1	5
Операції в особистому кабінеті	3	4	5	4	3	5	5	4	1
Комунікації банк/клієнт	2	6	6	1	5	6	1	5	7
СМС-банкінг	5	5	3	3	6	4	2	6	6
Оплата комунальних послуг	6	2	7	X	X	X	X	7	2

Залежно від операцій клієнта, опрацювання отриманої інформації може вплинути на подальшу консультацію клієнта. Наприклад, після отримання та використання кредитного ліміту мовний модуль повинен бути модифікований залежно від даної події.

Для прикладу можна взяти клієнта з кредитною картою. За правилами банку, залежно від потреб клієнта, на кредитній картці можуть зберігатися як власні, так й кредитні кошти. Тобто, для інформування клієнта ми маємо як мінімум два динамічних параметра. Як ми бачимо на рис. 3, з комбінації з двох елементів ми маємо три події (четверта подія з комбінації з двох елементів прирівнюється до третьої, а саме, коли відсутній кредитний ліміт, але присутня заборгованість клієнта, просто проінформує, що кредитний ліміт буде дорівнювати нулю). На кожну з подій повинен бути окремий голосовий модуль, який оператор повинен озвучувати.

На вибір типу голосового модуля у даному випадку впливає постфільтрація, так як тип тематики вибирається після дії клієнта, не важливо, чи то списання, чи то встановлення певного статусу.

5 Особливості програмної реалізації

Для роботи системи необхідно три основні компоненти:

1. Створена база даних на основі MySQL сервера;
2. IP комунікатор Cisco;
3. Створена телефонія.

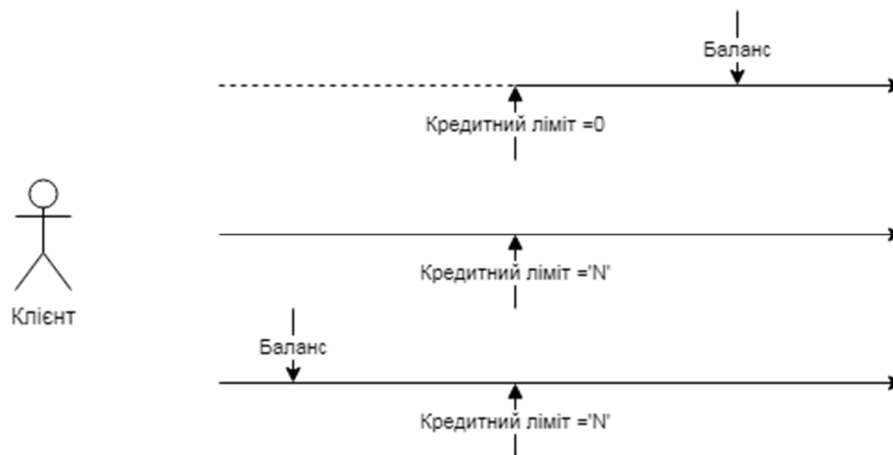


Рис. 3. Варіації одної тематики

Зв'язок між даними сутностями та їх взаємодії можна побачити на use case діаграмі (рис. 4).

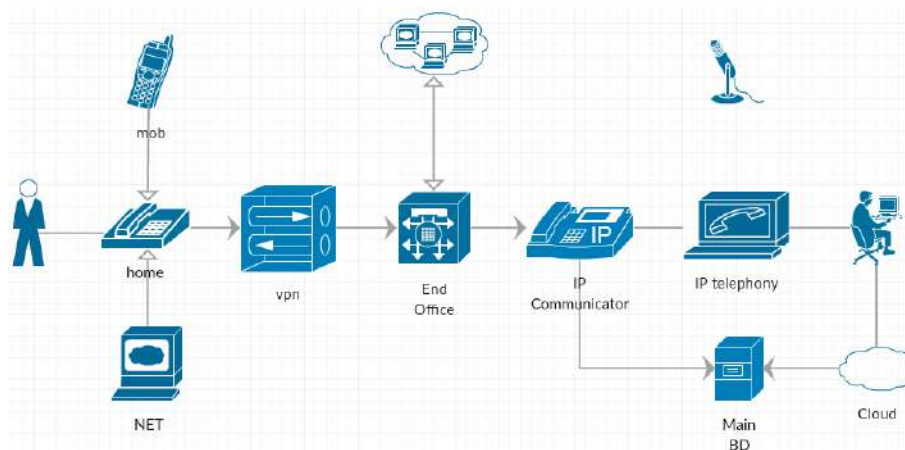


Рис. 4. Функціональна схема роботи створеної системи

Як ми бачимо на функціональній схемі, клієнт може звертатися до контактному центру будь-якого доступного для цього пристрою. Після того, як дзвінок був прийнятий, контакт переходить до «хмарної» адреси для подальшого опрацювання. Це потрібно для того, щоб зареєстрований номер міг бути один

(для зручності). Після цього, дзвінок потрапляє до ЄНД-офісу, а це значить, що даний дзвінок стає в чергу звернень. Дзвінки розмежуються між операторами шляхом сортування залежно від лінії. Після встановлення оператора дзвінок надходить на комунікатор.

Комунікатор компанії Cisco зареєстрований за певною особою за допомогою MAC-адреси мережевої карти та номеру телефонії. Номер телефонії – це певний спосіб надати можливість персоналізованої обробки даних.

Після отримання дзвінка комунікатор подає запит до бази даних та до телефонії. Телефонія завантажує дані про клієнта та дозволяє консультувати щодо завантажених даних.

На основі описаної схеми було створено демо-версію інтерфейсу з емулятором дзвінка (рис. 5).

Рис. 5. Демо-версія інтерфейсу

Так як дзвінок емульований, необхідно було ввести всі дані вручну. На рис. 5 відображена інформація, тобто відповідь на питання, яке система сама змоделювала. Критеріями вибору стали дата операції та наявність кредитного ліміту.

6 Висновки

У процесі дослідження визначено, що сучасні засоби автоматизації підвищують ефективність та коректність надання консультативних послуг. Мінімізація витрат часу та підвищення продуктивності праці в сфері комунікаційних послуг можливі за наступних умов: підвищення швидкості опрацювання питань клієнтів за рахунок вибору оптимальної методики пошуку відповіді; зменшення кількості помилок за рахунок використання шаблонів відповідей.

Найчастіше для автоматизації консультативних послуг застосовуються системи самообслуговування абонентів за допомогою інтерактивних голосових меню (IVR). У роботі запропоновані заходи, що надають можливість знизити трафік в подібних системах. Розроблене програмне забезпечення може спрогнозувати питання, яке необхідно опрацювати, що пришвидшить роботу контактного центру та підвищить його ефективність. Експериментальні дослідження підтвердили підвищення швидкості опрацювання даних, зниження трафіку за рахунок використання розробленої системи.

References

1. Khasnabish, B.: Implementing Voice over IP. John Wiley & Sons, Hoboken (2003)
2. Tolention, J.: Enhancing customer engagement with interactive voice response. <https://thenextweb.com/future-of-communications/2015/04/20/enhancing-customer-engagement-with-interactive-voice-response/#gref> (2015). Accessed 10 Nov 2018

Web-орієнтована система «Електронний розклад»

Ілля Вадимович Олексієнко, Василь Михайлович Франчук^[0000-0002-9443-6520]

Національний педагогічний університет імені М. П. Драгоманова,
вул. Пирогова, 9, м. Київ, 01601, Україна
i.oleksienko@std.npu.edu.ua, vfranchuk@npu.edu.ua

Анотація. Метою дослідження є створення web-орієнтованої системи для складання електронного розкладу з використанням хмарних сервісів. Завданням дослідження є проектування та розробка системи для введення та перегляду електронного розкладу занять навчального підрозділу закладу вищої освіти. Об'єктом дослідження є навчальний процес у закладі вищої освіти, предметом дослідження є web-орієнтована система «Електронний розклад». Результатом дослідження є впроваджена у навчальний процес web-орієнтована система для введення та перегляду розкладу занять навчального підрозділу закладу вищої освіти. Web-орієнтована система створюється з метою використання у навчальному процесі вищого навчального закладу для вільного доступу до електронного розкладу як для студентів, так і для викладачів.

Ключові слова: розклад, web-орієнтована система, хмарні сервіси; Google-календар.

Web-oriented electronic schedule

Illia V. Oleksiienko and Vasyl M. Franchuk^[0000-0002-9443-6520]

National Pedagogical Dragomanov University, 9, Pyrohov St., 9, Kyiv, 01601, Ukraine
i.oleksienko@std.npu.edu.ua, vfranchuk@npu.edu.ua

Abstract. Research objective is creating a web-oriented system for the compiling electronic schedule using cloud services. The task of the research is to design and development of a system for entering and viewing of electronic schedule training unit of higher education institution. A research object is the teaching process in higher education institution, the subject of the study is web-oriented system “Electronic schedule”. The research is implemented in the teaching process web-oriented system for entering and viewing schedule training unit of higher education institution. Web-based system is created for use in the educational process of higher educational institution for free access to electronic schedule, both for students and for teachers.

Keywords: schedule, web-oriented system, cloud services, Google-calendar.

Розклад занять – невід’ємна частина навчального процесу, оскільки як студенту, так і викладачу потрібно вчасно знати, де і коли проходять заняття. Тому редагувати та отримувати доступ до розкладу потрібно якомога швидше та зручніше.

Існують різні способи подання розкладу для студентів та викладачів. Найбільш простий та вживаний – друкований розклад, який можна переглянути лише в деканаті факультету. У такому разі студенти можуть фотографувати цей розклад та розсилати один одному у соціальних мережах. Недоліком такого методу є те, що студенти доволі часто перепитують один у одного розклад на наступний день, оскільки буває важко знайти потрібний знімок; при внесенні змін до розкладу (перенесення заняття чи зміна аудиторії) адміністрація не може швидко попередити про це студентів, а студентам потрібно постійно моніторити розклад в деканаті.

Одним із перших способів електронного розкладу занять стала Excel-таблиця, що викладається на сайті факультету. Цей спосіб є кращим за попередній, оскільки розклад можна вже подивитись у структурованому вигляді вдома. Недоліками залишається необхідність постійного моніторингу документа студентами та великі часові затрати на його створення та редагування.

На даний момент існують повноцінні реалізації «електронних розкладів» як web-сервісів. Наприклад, у Івано-Франківського національного технічного університету нафти і газу, Національного університету «Львівська політехніка» та інших, використання яких дозволяє переглядати розклад певної групи чи викладача, проте в них є такі недоліки, як відсутність спеціального експорту для мобільних пристроїв (оскільки саме мобільні пристрої використовуються студентами найбільше), неможливість швидкого перегляду розкладу групи та дізнатися, коли вільна певна аудиторія (для студентських заходів чи зборів).

Саме тому було поставлене завдання розробити систему «Електронний розклад», яка дозволяла б ефективніше працювати із розкладом як студентам та викладачам, так і адміністрації факультету.

Проект «Електронний розклад» розроблений в межах Студентського конструкторського бюро Факультету інформатики Національного педагогічного університету імені М. П. Драгоманова з метою створення системи, за допомогою якої можна отримувати доступ до розкладу занять як для студентів та викладачів, так і для адміністрації навчального підрозділу, яка складає його.

Для реалізації проекту «Електронний розклад» було запропоновано використати хмарні сервіси, зокрема від Google, а саме Google-календар, та власну розроблену web-орієнтовану систему для управління розкладом занять (рис. 1).

Web-орієнтована система складається з двох частин – клієнтської та серверної.

Клієнтська частина створена з використанням мов HTML5 та JavaScript. За допомогою JavaScript-коду отримуються необхідні дані з сервера, виводяться у зручному вигляді та надсилаються на сервер команди на виконання дій із розкладом занять.

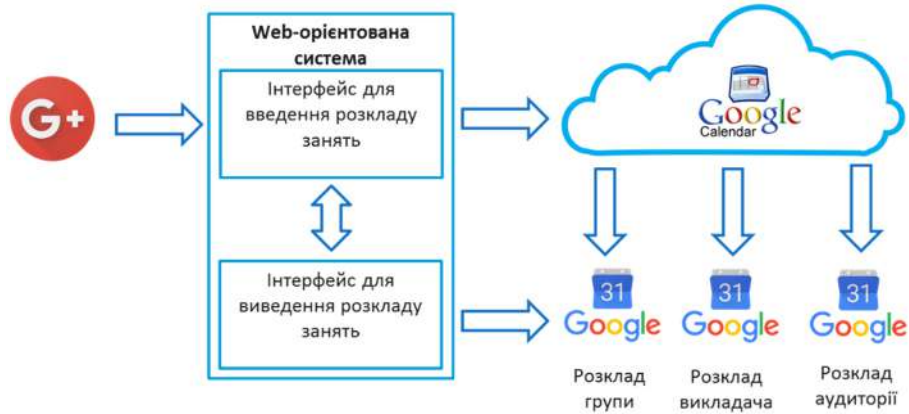


Рис. 1. Модель проекту «Електронний розклад»

Серверна частина реалізована з використанням СУБД MySQL та мови програмування PHP. Для експорту розкладу до Google-календарів використовується PHP бібліотека GoogleCalendarApi [1]. Також на сервері використаний захист від MySQL ін'єкцій, використання якого не дозволяє зловмисникам зашкодити цілісності даних.

Крім цього, система має два основних інтерфейси: для введення (рис. 2) та виведення (рис. 3) розкладу занять. Авторизація до інтерфейсу введення розкладу занять (адміністративної частини системи) виконується з використанням облікового запису Google. Доступ для редагування розкладу занять надається обліковому запису користувача, в якому створюються та зберігаються усі Google-календарі. Для облікового запису користувача задається рівень доступу, за допомогою якого визначається, які команди на сервері можуть виконуватись від його імені. Так, редагувати розклад чи список викладачів зможуть лише уповноважені особи.

Чисельник. Факультет інформатики

04.11.2016

Зберегти Сторінка викладачів Сторінка груп Сторінка аудиторій Створити календарі

Група: (v) Викладач: (v) Аудиторія: (v) День тижня: (v)

	ІІ І	ІІ КІІ	ІІ ІІІ	ІІ І	ІІ ІІІ
1	Алгебра і геометрія ств. Нестерова О.Д. 228	Комп'ютерні мережі ств. Малавек П.М. 1-14	Основи програмування ств. Лаврова І.М. 340		
2	Іноземна мова ас. Лаврова І.М. 228	Іноземна мова Задоро В. М. 402	Іноземна мова ств. Лаврова І.М. 228	Іноземна мова ас. Лаврова І.М. 340	
3	Математичний аналіз доц. Півра С. П. 319	Вища математика ств. Нестерова О.Д. 228	Українська мова (як професійним співрозумілим) доц. Кабачок О. О. 1-51	Іноземна мова ас. Лаврова І.М. 228а	Операційні системи ств. Малавек П.М. 1-51а
4		Комп'ютерні мережі ств. Малавек П.М. 119	Іноземна мова ас. Лаврова І.М. 340	Програмування ств. Болдєв Ю. П. 229	
5	Фізична освіта 401	Фізична освіта 401			

Рис. 2. Інтерфейс для введення розкладу занять

Чисельник. Факультет інформатики

04.12.2018 Друкуючи розклад Додати Google-календар Додати

Група: [v1] Викладач: [v1] Аудиторія: [v1] День тижня: [v1]

Початок	11 I	11 КЗ	11 ПІ	21 I	21 ПІ
8:30	Алгебра і геометрія ств. Пестрова О.Д. 225	Комп'ютерні мережі ств. Малажук П.М. 1-14	Українська мова (за професійною спрямованістю) Малажук П.М. ств. Пестрова О.Д. 225	Українська мова (за професійною спрямованістю) Малажук П.М. ств. Пестрова О.Д. 225	
9:20	Іноземна мова ас. Іванова Л.М. 225	Іноземна мова Зашара В. М. 402	Іноземна мова (за професійною спрямованістю) Малажук П.М. ств. Пестрова О.Д. 225	Іноземна мова ас. Іванова Л.М. 340	
10:30	Математичний аналіз доц. Пафок С. П. 319	Вища математика ств. Пестрова О.Д. 225	Українська мова (за професійною спрямованістю) доц. Кабін О. О. 1-51	Іноземна мова ас. Іванова Л.М. 225а	Операційні системи ств. Малажук П.М. 1-51а
11:00		Комп'ютерні мережі ств. Малажук П.М. 339	Іноземна мова ас. Іванова Л.М. 340	Програмування ств. Біляк Ю. П. 229	
11:30	Французька мова 401	Французька мова 401			
13:30					

Рис. 3. Інтерфейс для виведення розкладу занять

Для перегляду розкладу потрібно використати інтерфейс для перегляду розкладу занять. Для додавання потрібного розкладу до свого облікового запису Google необхідно авторизуватись.

Отже, за допомогою web-орієнтованої системи «Електронний розклад» можна створювати та переглядати розклад занять з використанням браузера та на мобільних пристроях. Оскільки система «Електронний розклад» реалізована у формі web-сервісу, доступ до неї можна отримати як з комп'ютера, так і з смартфона чи планшета, на якому є встановлений браузер. Для перегляду розкладу занять до системи було додано фільтри за групою, викладачем, аудиторією та днем тижня, використання яких дозволяє швидко знаходити потрібний розклад занять відповідно для групи, викладача, аудиторії та днем тижня.

Для власників пристроїв під управлінням операційної системи Android існує можливість додавати до Google-календаря календарі групи, викладача чи аудиторії, щоб швидко отримувати розклад групи, викладача, або дізнатися в який час є вільна певна аудиторія, на мобільному пристрої [2].

Крім цього, система зручна не тільки для перегляду, а й для редагування розкладу адміністрацією навчального підрозділу закладу вищої освіти. Усі зміни відразу заносяться до бази даних та автоматично синхронізуються з Google-календарями. Якщо потрібно отримати паперову версію розкладу, то його можна роздрукувати, попередньо відфільтрувавши необхідні дані.

References

1. API Reference | Calendar API | Google Developers. <https://developers.google.com/google-apps/calendar/v3/reference> (2017). Accessed 31 Dec 2017
2. Administrator guide to accessibility - G Suite Administrator Help. https://support.google.com/a/answer/2821355?hl=uk&ref_topic=3035040#h4_calendar (2017). Accessed 31 Dec 2017

Створення української локалізації системи комп'ютерної математики Sage

Олена Олексіївна Соменко

Кіровоградський інститут розвитку людини Відкритого міжнародного університету розвитку людини «Україна», пл. Дружби Народів, 8, м. Кропивницький, 25015, Україна
olenasmn@gmail.com

Анотація. *Цілі дослідження:* розкрити особливості створення української локалізації системи Sage засобами програми Poedit. *Завдання дослідження:* визначити основні проблеми створення україномовних інтерфейсів комп'ютерних програм; дати характеристику системи комп'ютерної математики Sage; описати методу створення української локалізації Sage Notebook у середовищі Poedit. *Об'єкт дослідження:* використання систем комп'ютерної математики у навчальному процесі. *Предмет дослідження:* створення української локалізації системи комп'ютерної математики Sage. *Методи дослідження:* аналіз наукових публікацій, програмного забезпечення, нормативно-правових документів. *Результати дослідження:* спираючись на аналіз україномовного програмного забезпечення та наукових публікацій з питань створення локалізацій програм, виявлено основні проблеми, які виникають у процесі адаптації програмного забезпечення до використання україномовними користувачами; надано характеристику системи Sage та визначено особливості процесу розробки та внесення змін у програму; розглянуто принципи роботи програми Poedit, як інструменту для створення української локалізації Sage Notebook. *Основні висновки і рекомендації:* 1) встановлено, що відсутність якісних українізованих інтерфейсів стає причиною гальмування активного процесу опанування україномовними користувачами світових інформаційних технологій; 2) визначено, що однією зі сфер, яка найбільше потерпає від відсутності якісних українських локалізацій програмних засобів, є середня та вища освіта; 3) система Sage є однією із найбільш ефективних СКМ, яка широко використовується у різних країнах як для роботи, так і в навчальному процесі, в Україні досі залишається на етапі впровадження; 4) використання програми Poedit для створення українських локалізацій програмного забезпечення є ефективним, а отже, отриманий досвід створення може бути застосований до адаптації й інших програмних продуктів для україномовного користувача.

Ключові слова: інформаційно-телекомунікаційні технології, локалізація програмного забезпечення, система комп'ютерної математики, Sage, Poedit.

Creation of Ukrainian localization of computer mathematics system Sage

Olena O. Somenko

Kirovohrad Institute of Human Development, Open International University of Human Development "Ukraine", 8, Friendship of Peoples Sq., Kropivnitskiy, 25015, Ukraine
olenasmn@gmail.com

Abstract. *Research goals:* to reveal the features of creation of Ukrainian localization system Sage by using applications Poedit. *Research objectives:* to identify the main problems of Ukrainian creating software interfaces; to characterize the system of computer mathematics Sage; describe methods of creating Ukrainian localization of Sage Notebook by using program Poedit. *Object of research:* the use of computer mathematics in the learning process. *Subject of research:* the creation of Ukrainian localization system of computer mathematics Sage. *Research methods:* analysis of scientific publications, software, legal documents. *Results of the research:* Analysis based on a Ukrainian software and publications on the establishment of localization program, revealed major problems arising in the process of adapting software for use Ukrainian-users; given the characteristics of the Sage and the peculiarities of the development and modification of the program; the principles of the program Poedit, as a tool to create a Ukrainian localization of Sage Notebook. *The main conclusions and recommendations:* 1) revealed a lack of quality Ukrainized interfaces causes inhibition of the active process of mastering the Ukrainian-world users of information technology; 2) stipulates that one of the areas that suffered most from the absence of high-quality Ukrainian localization of software, is secondary and higher education; 3) system Sage is one of the most effective SCM, which is widely used in different countries for both work and in the educational process in Ukraine is still at the stage of implementation; 4) use the program Poedit to create a Ukrainian localization software is effective, and thus create the experience can be applied to adaptation and other software products for Ukrainian-user.

Keywords: information and telecommunications technology, software localization, computer mathematics system, Sage, Poedit.

Процес формування в Україні інформаційного суспільства в умовах сучасних світових тенденцій глобалізації ставить дуже гостро перед усіма україномовними користувачами проблему неадаптованості засобів інформаційних технологій до національних, мовних і культурних особливостей. Нині більшість програмного забезпечення у світі створюється англійською мовою, що робить його функціонал зрозумілим, а використання – доступним для більшості користувачів. Однак, в той же час, цей чинник може стати і суттєвою перешкодою у процесі

освоєння нових програмних засобів, викликати труднощі у розумінні того, для чого призначена і як працює та чи інша команда, призводить до помилок у роботі з ними. Тому відсутність якісних українізованих інтерфейсів стає причиною гальмування активного процесу опанування світових інформаційних технологій [5; 7].

Загалом, описана проблема є характерною для усіх напрямків використання програмних засобів у сучасному житті. Проте, однією зі сфер, яка найбільше втрачає від відсутності українських локалізацій програмних засобів і найбільш гостро потребує створення та вдосконалення україномовних програмних інтерфейсів, є освіта. Адже, як у процесі навчання школярів, так і під час фахової підготовки молоді у системі вищої освіти, одним із визначальних чинників ефективності навчального процесу є озброєння учнів та студентів навичками роботи із різного роду сучасними програмними засобами. Однак, темпи адаптації комп'ютерних програм до україномовного середовища значно відстають від світових темпів створення та впровадження нових освітніх програмних засобів [6; 15].

Прикладом є одна із найбільш популярних у світі систем комп'ютерної математики (СКМ), вільнопоширювана система Sage, яка давно досить широко і успішно використовується у різних країнах як для роботи, так і в навчанні. В Україні методика використання СКМ Sage розробляється науковою школою С. О. Семерікова (м. Кривий Ріг) [7; 8; 9; 10; 11; 12; 13].

Метою нашої роботи є розкриття особливостей процесу створення української локалізації СКМ Sage (Sage Notebook) засобами програми для перекладу інтерфейсів програмних додатків та сайтів – Poedit.

Розробку Sage було започатковано Вільямом Стейном (університет Вашингтона, США) у 2004 році, а вже на початку 2005 року вийшла перша версія СКМ. Початковою метою створення системи Sage була розробка доступного, безкоштовного, вільнопоширюваного математичного програмного пакету, який би кожен міг використовувати для навчання, роботи та проведення досліджень. Sage створювалася на протигагу комерційним математичним програмам із закритим вихідним кодом, тому має ряд відмінностей у порівнянні з іншими математичними системами. Зокрема, система Sage використовує широковідому мову програмування Python, тоді як інші популярні математичні програмні продукти, наприклад, Maple, Mathematica і Matlab, використовують власні специфічні математичні мови. Завдяки цій особливості у користувачів системи з'являється можливість реалізувати у системі Sage будь-який код, написаний на Python, а також, відповідно, код мови Python, створений у Sage, можна інтегрувати у різноманітні програмні додатки, що значно розширює можливості та сфери застосування математичного пакету [4].

Ще однією особливістю Sage є те, що при її створенні, замість того, щоб повторно писати велику кількість основних бібліотек, як це відбувається при створенні більшості математичного програмного забезпечення, було об'єднано кращі математичні програмні продукти із відкритим вихідним кодом, такі як NumPy, SciPy, Matplotlib, SymPy, Maxima, GAP, FLINT, R та ще багато інших. У цілому, Sage об'єднує близько 100 різноманітних програм, загальний перелік

яких можна знайти за посиланням: www.sagemath.org/links-components.html. Також Sage включає і велику кількість нових власних розробок, що загалом утворює потужну платформу для математичних досліджень. Система надає унікальну можливість використовувати інструменти різних математичних пакетів та засоби програмування для розв'язання широкого спектру завдань. Sage включає як складний багатокористувацький графічний веб-інтерфейс, так і інтерфейс командного рядка, а також передбачає роботу з іншими інтерактивними середовищами розробки (IDE) мови Python [1].

Отже, такий підхід до створення нового математичного програмного забезпечення дозволив створити нову математичну систему, яка, завдяки відкритості програмного коду, передбачає широкі можливості модифікації, гнучкість роботи у комп'ютерному середовищі, швидко реагує на зміни, оперативно пристосовується до нових запитів і потреб користувачів, і, крім цього, є безкоштовною для використання та об'єднує сотні розробників по всьому світу. Це дозволяє програмному продукту постійно вдосконалюватися, враховуючи думки, побажання і, навіть, власні розробки користувачів. Набагато якісніше і швидше відбувається відслідковування та виправлення помилок програми, додавання нових можливостей і функцій.

Користувач, починаючи працювати із системою Sage, звернувшись до офіційний сайту програми www.sagemath.org, на головній сторінці може знайти такі розділи:

1. **SageMath Online** – перехід до онлайн версій Sage: хмарного сервісу SageMathCloud (SMC), що знаходиться за адресою cloud.sagemath.com, та SageMathCell – веб-інтерфейсу Sage у вигляді командного рядка за посиланням sagecell.sagemath.org.
2. **Download 7.5.1** – розділ для завантаження останньої версії Sage (на даний час це версія 7.5.1 від 13.01.2017), а також підрозділи: Changelogs – історія змін; Source 7.5.1 – вихідний код поточної версії; Packages – файли *.spkg для Sage; Git – останні зміни у репозиторії Sage на GitHub.
3. **Help/Documentation** – містить довідкові матеріали по роботі з системою Sage: Video – навчальні відео; Forums – групи для обговорень різних питань, пов'язаних із роботою Sage; Tutorial – підручник; FAQ – найбільш часті запитання; Questions? – форум для користувачів.
4. **Feature Tour** – оглядовий тур по Sage із підрозділами: Quickstart – рекомендації для швидкого початку роботи; Research – короткі інструкції для проведення математичних досліджень; Graphics – вказівки щодо роботи з графікою.
5. **Library** – бібліотека, що містить: Testimonials – відгуки; Books – книги; Publications – статті та інші публікації, пов'язані із Sage; Press Kit – загальна інформація про проект.
6. **Search** – пошук по сайту.

Sage створювався і продовжує розвиватися завдяки волонтерським зусиллям сотень людей, фінансуванню Національного наукового фонду (National Science

Foundation), підтримці і спонсорству таких організацій, як Google та Microsoft, а також благодійним внескам приватних осіб.

Оскільки, система Sage є вільно поширюваним програмним забезпеченням, вона ліцензується за GNU Public License (GPL). Ліцензія GNU GPL передбачає надання користувачам таких свобод:

- свободу запуску програми з будь-якою метою;
- свободу вивчення того, як програма працює, і її модифікації (попередньою умовою цього є гарантований доступ до вихідного коду);
- свободу поширення копій;
- свободу покращення програми й викладення покращеної програми у публічний доступ (обов'язковою умовою є гарантований доступ до вихідного коду) [3].

Згідно концепції GNU GPL, користувачі мають право на копіювання, зміну і поширення програми та зобов'язані надавати такі ж права користувачам усіх інших програмних продуктів, похідних від цієї програми. Тобто Sage не містить ліцензійних кодів, захисту від копіювання, закритих алгоритмів, а отже – не має жодної частини програми, яку б користувач не міг побачити або змінити.

У даний час робота із вихідним кодом системи Sage здійснюється за допомогою веб-сервісу для розміщення та спільної розробки ІТ-проектів – GitHub [2], яку ще називають соціальною мережею для розробників програмного забезпечення. На сторінці організації Sage Mathematical Software System (<https://github.com/sagemath>) розміщені репозиторії різних проектів, зокрема, репозиторій sage – містить файли вихідного коду системи Sage (рис. 1); sagenb – вихідний код Sage Notebook, програмного засобу для локального встановлення системи Sage на персональний комп'ютер; sagescell – вихідний код для Sage Cell Server, веб-сервісу, що дозволяє виконувати короткі обчислення безпосередньо у браузері та ін.

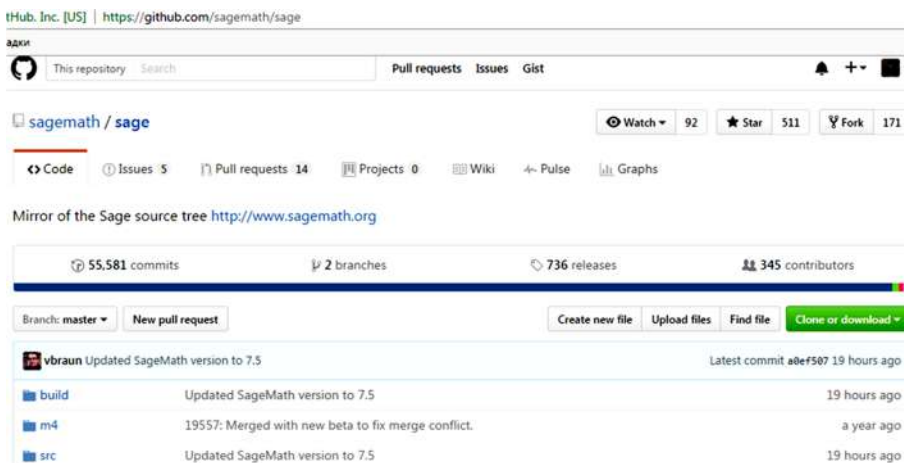


Рис. 1. Репозиторій із файлами вихідного коду Sage на веб-сервісі GitHub

Для перегляду файлів вихідного коду проектів користувачеві не обов'язково реєструватися на GitHub, однак, якщо виникає потреба у редагуванні, внесенні пропозицій чи коректив у файли проекту, тоді користувачеві необхідно пройти нескладну процедуру реєстрації і створити власний акаунт (рис. 2). Після реєстрації можна створювати власні репозиторії і приєднуватися до роботи над уже існуючими проектами користувачів, створюючи власні гілки проектів (fork). Для подальшої роботи слід встановити на свій персональний комп'ютер Git-клієнт – спеціальне програмне забезпечення для віддаленої роботи із файлами проекту. За допомогою цього програмного забезпечення на ПК користувача створюється локальна копія проекту, над яким ведеться робота. Після редагування чи додавання власних файлів до проекту за допомогою Git-клієнта локальні зміни завантажуються у віддалений репозиторій, де можуть переглядатись, коментуватись та редагуватись іншими розробниками. За умови доцільності внесених правок та їх успішної і ефективної роботи у системі, нові зміни включаються до наступних версій Sage.

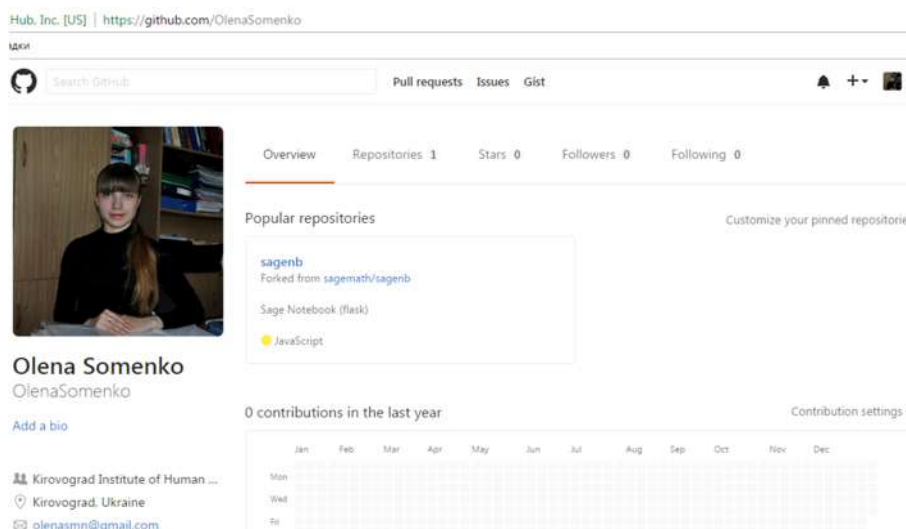


Рис. 2. Загальний вигляд домашньої сторінки користувача GitHub

Програма Poedit передбачає використання для перекладу мовних файлів програм чи шаблонів для їх перекладу. Це файли із розширенням *.po або *.pot відповідно, які містять рядки для перекладу у текстовому форматі. На відміну від них, файли *.mo представляють собою уже скомпільовану версію. Після створення перекладу у форматі *.po, необхідно його скомпільувати в файл *.mo. Програма Poedit дозволяє це зробити автоматично.

Для перекладу інтерфейсу Sage Notebook було використано файл message.pot, який знаходиться у репозиторії sagenb. Для створення нового перекладу слід виконати команду «Файл» – «Создать из POT-файла», після чого обрати потрібний файл *.pot, у нашому випадку, це message.pot. Далі натиснути кнопку

«Создать новый перевод» і обрати мову. Після вибору української мови буде автоматично створено файл uk_UA.po, який слід зберегти. Вікно програми Poedit змінить вигляд – з'явиться дві колонки тексту, зліва – рядки програми мовою оригіналу, справа – рядки перекладу, які заповнює користувач (рис. 3). Після збереження відбувається автоматична компіляція файлу у формат *.mo.

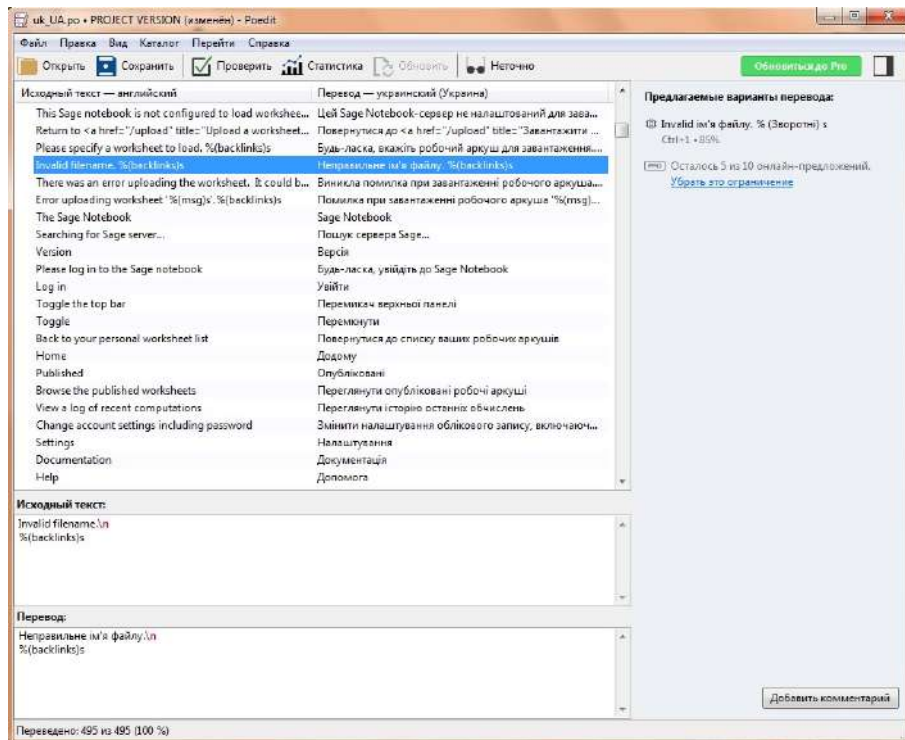


Рис. 3. Загальний вигляд вікна програми Poedit із фрагментом перекладу інтерфейсу Sage Notebook

Далі, як було описано вище, отримані файли *.po та *.mo було завантажено до проекту на ресурсі GitHub, де, після розгляду іншими розробниками, вони були внесені до основної гілки вихідного коду проекту sagenb/sagenb/translations/uk_UA/LC_MESSAGES/messages.po (рис. 4).

Таким чином, нами було розроблено українську локалізацію системи Sage, а саме, Sage Notebook, яка стала частиною загального коду і з грудня 2015 року входить до наступних версій програми.

Poedit – програма, за допомогою якої було здійснено переклад, є простою у застосуванні, зручною та дуже ефективною, а отже, отриманий досвід створення українських локалізацій програмного забезпечення може бути застосований до адаптації й інших програмних продуктів для україномовного користувача.

The screenshot shows a GitHub repository page for 'sagemath/sagenb'. The commit history shows a commit by 'kcrisman' titled 'Moved Ukrainian translation files to correct place' on Dec 1, 2015. The selected file is 'sagenb / sagenb / translations / uk_UA / LC_MESSAGES / messages.po', which is 2587 lines (2079 sloc) and 100 KB. The commit message includes the following text:

```

1 # Ukrainian (Ukraine) translations for Sage (www.sagemath.org)
2 # This file is distributed under the same license as the Sage project.
3 # Olena O. Somenko <olenasm@gmail.com>, 2015.
4 #
5 msgid ""
6 msgstr ""
7 "Project-Id-Version: sagenb\n"
8 "Report-Msgid-Bugs-To: <olenasm@gmail.com>\n"
9 "POT-Creation-Date: 2015-07-16 11:00+0300\n"
10 "PO-Revision-Date: 2015-07-20 15:29+0300\n"
11 "Last-Translator: Olena O. Somenko <olenasm@gmail.com>\n"
12 "Language-Team: Olena O. Somenko <olenasm@gmail.com>\n"
13 "MIME-Version: 1.0\n"
14 "Content-Type: text/plain; charset=UTF-8\n"
15 "Content-Transfer-Encoding: 8bit\n"
16 "Generated-By: Babel 0.9.6\n"
17 "X-Generator: Poedit 1.8.2\n"
18 "Plural-Forms: nplurals=3; plural=(n%10==1 && n%100!=11 ? 0 : n%10==2 && n"
19 "%10<=4 && (n%100>10 || n%100>20) ? 1 : 2);\n"
20 "Language: uk_UA\n"
21 "X-Poedit-SourceCharset: UTF-8\n"
22

```

Рис. 4. Гілка вихідного коду Sage Notebook із українською локалізацією програми

Проблема створення якісних україномовних інтерфейсів комп'ютерних програм, досі стоїть дуже гостро. Це пов'язано не тільки із питаннями впровадження та використання української мови у сфері інформаційних технологій, але і з недостатньою розробленістю законодавства. Така ситуація призводить до створення неякісних перекладів, спотворення змісту текстів, порушення авторських прав тощо. Тому для ефективного розв'язання вказаних проблем необхідно об'єднання зусиль фахівців із галузей філології, комп'ютерних наук та права.

Список використаних джерел

1. Bard G.V. Sage for Undergraduates / G.V. Bard. – American Mathematical Society, 2015. – 336 p.
2. The world's leading software development platform · GitHub [Electronic resource]. – 2018. – Access mode : <https://github.com>.
3. GNU General Public License [Electronic resource] // GNU Project – Free Software Foundation. – 2007. – Access mode: <https://www.gnu.org/licenses/gpl-3.0.html>.

4. Stein W. Sage for Power Users [Electronic resource] / William Stein. – 2012. – 146, [1] p. – Access mode : <https://wstein.org/books/sagebook/sagebook.pdf>.
5. Булаховський К. А. Стан і проблеми розвитку україномовної локалізації комп'ютерних програм / Кирило Булаховський // Українська мова. – 2010. – № 4. – С. 89-94.
6. Величко С. П. Підготовка майбутніх вчителів фізики в умовах глобальної інформатизації навчального процесу / С. П. Величко, Д. В. Соменко // Збірник наукових праць Уманського державного педагогічного університету імені Павла Тичини. – Умань : ПП Жовтий О. О., 2011. – Ч. 3. – С. 38-45.
7. Модло Є. О. Розробка фільтру SageMath для Moodle / Є. О. Модло, С. О. Семеріков // Новітні комп'ютерні технології. – Кривий Ріг : Видавничий центр ДВНЗ «Криворізький національний університет», 2014. – Том XII : спецвипуск «Хмарні технології в освіті». – С. 233-243.
8. Семеріков С. О. Генерування математичних завдань засобами Web-СKM SAGE / Семеріков С. О., Шокалюк С. В., Мінтій І. С., Волошаненко О. С., Кулініч Б. М. // Матеріали п'ятої науково-практичної конференції FOSS Lviv 2015. 23-26 квітня 2015 р. / Львівський національний університет імені Івана Франка, Національний педагогічний університет імені М. П. Драгоманова, Брестський державний технічний університет, Проект Linux Vacation / Eastern Europe, Українська академія друкарства, Львівська група користувачів Linux. – [Львів], [2015]. – С. 74-76.
9. Семеріков С. О. MMC Sage в моделюванні економічних процесів / С. О. Семеріков, Н. А. Хараджян // Інформаційні технології та моделювання в економіці : збірник наукових праць Другої Міжнародної науково-практичної конференції ; Черкаси, 19-21 травня 2010 р. / Редкол. : Соловйов В. М. (відп. за випуск) та ін. – Черкаси : Брама-Україна, 2010. – С. 259-261.
10. Семеріков С. О. Мобільне математичне середовище Sage: нові можливості та перспективи розвитку / С. О. Семеріков, І. О. Теплицький // Тези доповідей VII Всеукраїнської науково-практичної конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2010). 4-6 травня 2010 року. – Том 2. – Черкаси : Черкаський державний технологічний університет, 2010. – С. 71.
11. Семеріков С. О. Організація розподілених обчислень засобами MMC Sage / С. О. Семеріков, С. В. Шокалюк // Педагогічні науки: теорія, історія, інноваційні технології. – 2010. – №2 (4). – С. 338-345.
12. Семеріков С. О. Розробка фільтру Sage для СДН Moodle / С. О. Семеріков, С. В. Шокалюк, Ю. В. Плещ, І. С. Мінтій, В. В. Ткачук // Новітні комп'ютерні технології. – К. : Мінрегіон України. – 2011. – Том IX. – С. 189-194.
13. Словак К. І. Методика використання мобільних математичних середовищ у процесі навчання вищої математики студентів економічних спеціальностей : автореф. ... канд. пед. наук : 13.00.10 – інформаційно-комунікаційні технології в освіті / Словак Катерина Іванівна ; Інститут інформаційних технологій і засобів навчання НАПН України. – К, 2011. – 21 с.
14. Душкевич О. О. Використання СКМ Sage у професійній підготовці майбутніх вчителів математиків / Олена Душкевич // Наукові записки. Серія: Проблеми методики фізико-математичної і технологічної освіти. – Кіровоград: РВВ КДПУ ім. В. Винниченка, 2014. – Випуск 5. – Частина 1. – С. 24-28.
15. Триус Ю. В. Комп'ютерно-орієнтовані методичні системи навчання математичних дисциплін у ВНЗ: проблеми, стан і перспективи / Ю. В. Триус // Науковий часопис Національного педагогічного університету імені М. П. Драгоманова. Серія 2:

Комп'ютерно-орієнтовані системи навчання. – К. : Вид-во НПУ ім. М. П. Драгоманова, 2010. – Вип. 9 (16). – С. 16-29.

References (translated and transliterated)

1. Bard, G.V.: Sage for Undergraduates. American Mathematical Society, Providence (2015)
2. The world's leading software development platform · GitHub. <https://github.com> (2018)
3. GNU General Public License. GNU Project – Free Software Foundation. <https://www.gnu.org/licenses/gpl-3.0.html> (2007)
4. Stein, W.: Sage for Power Users. <https://wstein.org/books/sagebook/sagebook.pdf> (2012). Accessed 19 Nov 2017
5. Bulakhovskiy, K.A.: Stan i problemy rozvytku ukrainomovnoi lokalizatsii kompiuternykh prohram (State and problems of development of Ukrainian-language localization of the computer programs). *Ukrainska mova*. 4, 89–94 (2010)
6. Velychko, S.P., Somenko, D.V.: Pidhotovka maibutnykh vchyteliv fizyky v umovakh hlobalnoi informatyzatsii navchalnoho protsesu (Training future teachers of physics in the global informatization of educational process). *Zbirnyk naukovykh prats Umanskoho derzhavnogo pedahohichnoho universytetu imeni Pavla Tychyny*. 3, 38–45 (2011)
7. Modlo, E.O., Semerikov, S.O.: Development of SageMath filter for Moodle. *New computer technology*. 12, 233–243 (2014)
8. Semerikov, S.O., Shokaliuk, S.V., Mintii, I.S., Voloshanenko, O.S., Kulinich, B.M.: Heneruvannia matematychnykh zavdan zasobamy Web-SKM SAGE (Generation of mathematical tasks by Web-SCM SAGE). In: *Materialy piatoi naukovo-praktychnoi konferentsii FOSS Lviv 2015*. 23-26 kvitnia 2015 r., Lvivskiy natsionalnyi universytet imeni Ivana Franka, Lviv, pp. 74–76 (2015)
9. Semerikov, S.O., Kharadzhan, N.A.: MMS Sage v modeliuvanni ekonomichnykh protsesiv (MME Sage as a tool of economic processes modeling). In: Soloviov, V.M. (ed.) *Informatsiini tekhnologii ta modeliuvannia v ekonomitsi: zbirnyk naukovykh prats Druhoi Mizhnarodnoi naukovo-praktychnoi konferentsii, Cherkasy, 19-21 travnia 2010 r.*, pp. 259–261. Brama-Ukraina, Cherkasy (2010)
10. Semerikov, S.O., Teplyckyj, I.O.: Mobilne matematyчне середовище Sage: novi mozhlyvosti ta perspektyvy rozvytku (Mobile mathematical environment Sage: a new features and development prospects). In: *Tezy dopovidej VII Vseukrainskoi naukovo-praktychnoi konferencii «Informacijni tekhnologii v osviti, nauci i tehnicii» (ITONT-2010)*, 4-6 travnja 2010 roku, vol. 2, p. 71. Cherkaskyj derzhavnyj tekhnologichnyj universytet, Cherkasy (2010)
11. Semerikov, S.O., Shokaliuk, S.V.: Orhanizatsiia rozpodilenykh obchyslen zasobamy MMS Sage (The organization of distributed computing using MME Sage). *Pedahohichni nauky: teoriia, istoriia, innovatsiini tekhnologii*. 2(4), 338–345 (2010)
12. Semerikov, S.O., Shokalyuk, S.V., Plyushh, Yu.V., Mintij, I.S., Tkachuk, V.V.: Rozrobka filtru Sage dlya SDN Moodle (Sage filter development for distance learning system Moodle). *New computer technology*. 9, 189–194 (2011)
13. Slovak, K.I.: *Metodyka vykorystannia mobilnykh matematychnykh sredovyshch u protsesi navchannia vyshchoi matematyky studentiv ekonomichnykh spetsialnostei (Method of using mobile mathematical environments in the process of higher mathematics students studying in economic specialties)*. Dissertation, Instytut informatsiinykh tekhnologii i zasobiv navchannia NAPN Ukrainy (2011)

14. Dushkevych, O.O.: Vykorystannia SKM Sage u profesiinii pidhotovtsi maibutnikh vchyteliv matematyky (Use of CMS Sage in the training of future mathematical teachers). **5**(1), 24–28 (2014).
15. Tryus, Yu.V.: Kompiuterno-oriientovani metodychni systemy navchannia matematychnykh dystsyplin u VNZ: problemy, stan i perspektyvy (Computer-oriented methodological training system of mathematical sciences in universities: problems and prospects). Naukovyi chasopys Natsionalnoho pedahohichnoho universytetu imeni M. P. Drahomanova. Seriiia 2: Kompiuterno-oriientovani systemy navchannia. **9**(16), 16–29 (2010)

Адаптивний штучний інтелект у RPG-грі на ігровому рушії Unity

Олег Михайлович Гаранін,
Наталія Володимирівна Моїсеєнко^[0000-0002-3559-6081]

Криворізький державний педагогічний університет,
пр. Гагаріна, 54, м. Кривий Ріг, 50086, Україна
{olejenuc, n.v.moiseenko}@gmail.com

Анотація. *Метою* дослідження є розробити комп'ютерну RPG-гру із вбудованим адаптивним штучним інтелектом за допомогою рушія Unity. *Задачі* дослідження: розробка системи адаптації ігрових ситуацій залежно від дій гравця та прив'язка цього механізму до ігрового процесу. *Об'єкт* дослідження: комп'ютерні RPG-ігри на базі ігрових рушіїв. *Предмет* дослідження: комп'ютерні RPG-ігри із вбудованим штучним інтелектом на базі ігрових рушіїв. *Результатами* дослідження є розроблена RPG-гра із вбудованим адаптивним штучним інтелектом.

Ключові слова: адаптація, штучний інтелект, розробка комп'ютерних ігор, ігрові рушії, Unity.

Adaptive artificial intelligence in RPG-game on the Unity game engine

Oleh M. Haranin and Natalia V. Moiseienko^[0000-0002-3559-6081]

Kryvyi Rih State Pedagogical University, 54, Gagarin Ave., Kryvyi Rih, 50086, Ukraine
{olejenuc, n.v.moiseenko}@gmail.com

Abstract. *The purpose* of this study is development of RPG-game with built-in adaptive artificial intelligence on the Unity game engine. *Objectives of study:* development of the system of adaptation of playing situations depending on the actions of player and attachment of this mechanism to the playing process. *The object of research:* computer RPG-games on the game engines. *The subject of research:* computer RPG-games with built-in adaptive artificial intelligence on the game engines. *Results of the study* is developed RPG-game with built-in adaptive artificial intelligence.

Keywords: adaptation, artificial intelligence, development of computer games, game engines Unity.

1 Постановка проблеми

У наш час комп'ютерні ігри є значним сегментом ринку програмного забезпечення. В умовах жорсткої конкуренції, коли з'явилося багато великих компаній, безліч невеличких та інді-розробників, гравців весь час намагаються дивувати якимось новинками. Але що до ідеї, коли відома гра постійно може дивувати гравця? Це значно підвищить реігравельність, посилить інтерес як гравців, що вже мають досвід, так і тих, що за відгуками підуть, щоб спробувати. Одним із аспектів, що впливають на зацікавленість гравців в конкретній грі, є «відгук» гри. Певно, гравець мало повірить в історію, де усі його супротивники – бездумні ляльки для биття, а не доречні істоти із своїм мисленням, характером, які не «пруть на потяг із вилами», а обдумують стратегію, діють протилежно від самого гравця, змушуючи вже його змінювати стратегію, щоб перемагати. Таке «поглиблення» дає ще більший вплив на злиття з ігровим світом. Звісно, технології розвитку штучного інтелекту на даний момент стрімко розвиваються [1; 2; 3; 4].

Все вищезазначене визначило *мету* роботи – розробити комп'ютерну RPG-гру з убудованим штучним інтелектом за допомогою рушія Unity.

2 Обговорення результатів

Розробка гри проводиться на рушії Unity, по-перше, оскільки це один з відомих ігрових рушіїв, тож можна знайти багато корисної інформації щодо допомоги у розробці (форуми, відеоуроки, офіційні курси тощо). По-друге – гнучкість. Гру, створену на Unity (якщо приділити цьому увагу під час розробки), можна легко адаптувати під більшість платформ, що дає можливість ще збільшити аудиторію.

Розроблена рольова гра є RPG-грою з дослідженнями території та виконанням завдань (рис. 1). Для ідентифікації присутності штучного інтелекту створений простий сюжет: за університетом та життям його мешканців деякий час стежили прибульці, вивчаючи особливості поведінки. Згодом вони ініціювали атаку, керуючись отриманими знаннями. Студенти та персонал намагались відбитись, але всі намагання були марними – прибульці знали наперед дії, тож просто керували ситуацією. Так і сиділи вони у полоні, поки не з'явився невідомий абітурієнт – свіжий розум у цьому гармидері, що, заручившись підтримкою факультетських героїв, прокладе шлях до перемоги.

Розглянемо механізм розробленого адаптивного штучного інтелекту. Його версія буде складається з двох частин: аналіз типів використовуваних здібностей та загальний аналіз стилю гри.

Аналіз типів здібностей використовує систему «динамічної броні». Дана система постійно аналізує, від якого типу пошкоджень супротивники гравця переважно отримують пошкодження. Існує статичний показник захисту від типів

пошкоджень для кожного ворога. До нього додається «плаваючий» показник, що розподіляється відповідно до співвідношення отримуваних типів пошкоджень.



Рис. 1. Бій із елементами інтерфейсу (нанесені пошкодження, шкали здоров'я)

Яким чином це працює. Є рядовий супротивник – прибулець звичайний та 5 типів пошкоджень у грі: фізичний, вогняний, електричний, хімічний, природний. Умовно наш прибулець має 20 од. фізичного опору (що еквівалентно зменшенню отримуваних фізичних пошкоджень на 20 %) і не має інших. Його масив опору має такий вигляд: [20, 0, 0, 0, 0].

Розглянемо 3 ситуації:

- (1) гравець має групу бійців, що використовують лише здібності з фізичним типом пошкоджень;
- (2) гравець має групу бійців, що використовують лише здібності з вогняним типом пошкоджень;
- (3) гравець має групу бійців, що використовують комбіновані здібності з фізичним та вогняним типами пошкоджень.

Самі здібності з фізичним типом пошкоджень мають приблизно на 15 % більшу ефективність за рахунок того, що знайти захист від нього у грі найпростіше (навіть початковий комплект речей, що мають при собі бійці дає від 3 до 15 одиниць фізичного захисту).

Розрахуємо умовну продуктивність під час бою: припустимо, є умовна здібність А, що наносить 100 одиниць пошкоджень. Створимо схожі здібності, але вже з типізацією пошкоджень: АР – наносить $100+15\% = 115$ одиниць фізичних пошкоджень та АФ – наносить 100 одиниць пошкоджень від вогню.

Візьмемо за розрахунок подальшої ефективності 11 боїв – 10 базово необхідно грі, щоб саме визначитись із початковими показниками (та для того, щоб прив'язати до сюжету те, що прибульці ще не мають ніякого уявлення про дії гравця), а саме, конкретно, 10 та 11 бої. До кінця 10 бою система завершить визначення пріоритетного типу пошкоджень та на 11 бій вже буде використовувати «динамічну броню».

Розрахуємо, що буде траплятись у наведених вище ситуаціях. Допустимо, весь час у боях проти гравця участь брали групи прибульців по 4 особи, кожен з яких мав запас здоров'я рівний 1000 одиниць. На кінець 10 бою система матиме такі записи щодо типів пошкоджень відповідно:

Ситуація 1: 40000 од. фізичних пошкоджень, 0 для всіх інших;

Ситуація 2: 40000 од. вогняних пошкоджень, 0 для всіх інших;

Для ситуації 3 візьмемо такий розклад, при якому гравець наносив порівну і фізичних і вогняних пошкоджень, отже, на виході матимемо 20000 од. фізичних пошкоджень і 20000 од. вогняних пошкоджень.

Масиви у системі відповідно матимуть такий вигляд:

- (1) [40000; 0; 0; 0; 0]
- (2) [0; 40000; 0; 0; 0]
- (3) [20000; 20000; 0; 0; 0]

Далі необхідно розрахувати коефіцієнти нанесених пошкоджень. Робиться це за допомогою ділення кількості конкретних нанесених пошкоджень на суму їх всіх. Матимемо наступний результат:

- (1) [1; 0; 0; 0; 0]
- (2) [0; 1; 0; 0; 0]
- (3) [0,5; 0,5; 0; 0; 0]

Далі «вступає у бій» та сама «динамічна броня». Припустимо, що її показник дорівнює 40. Тепер ми беремо його, множимо на коефіцієнт нанесених пошкоджень та додаємо до масиву захистів нашого прибульця. Маємо наступне:

- (1) [60; 0; 0; 0; 0]
- (2) [20; 40; 0; 0; 0]
- (3) [40; 20; 0; 0; 0]

І вже у 11 бою гравцеві доведеться боротися із прибульцем, що «підготувався» до бою, орієнтуючись на «гіркий досвід» минулих супротивників гравця. Тепер можна розрахувати, яким чином це впливає на ефективність бою у тому чи іншому випадку.

Ситуація 1: До «зміни правил» гравцеві для перемоги над 4 прибульцями із 1000 од. здоров'я у кожного необхідно було нанести $4000/0,8=5000$ од. фізичних пошкоджень. Враховуємо для усіх ситуацій те, що 1 удар наносить 115 од. пошкоджень і тим, що показник броні вже було враховано вище, потрібно було не менше, ніж $5000/115=44$ атаки. Врахуємо, що за 1 раунд кожен з бійців робить 1 дію. При цьому ефективність бою буде дорівнювати $100/(44/4)=9,09$ (100

ділимо на загальну кількість атак, поділену на кількість атак за раунд, тобто $100/(AC/APR)$).

Після «зміни правил» гравцеві для перемоги над тими ж прибульцями необхідно нанести $4000/0,4=10000$ од. фізичних пошкоджень. Для цього необхідно нанести не менше, ніж $10000/115=87$ атак. Ефективність бою буде дорівнювати $100/(87/4)=4,6$.

Фактично, у цій ситуації ефективність знизилась у 1,98 разів.

Ситуація 2: Відповідно, у бою перед «зміною» необхідно було нанести $4000/1=4000$ од. вогняних пошкоджень. 1 удар наносить 100 од. пошкоджень, отже, для перемоги необхідно $4000/100=40$ атак. Ефективність бою дорівнює $100/(40/4)=10$.

Після використання «динамічної броні» необхідно нанести $4000/0,6=6667$ од. вогняних пошкоджень. Для цього потрібно не менше, ніж $6667/100=67$ атак. Ефективність бою дорівнює $100/(67/4)=5,97$. У цій ситуації ефективність бою зменшилась у 1,67 разів, адже тут було використано більш продуктивний тип пошкоджень проти конкретної цілі – прибулець ВЖЕ мав показник фізичного захисту, але не мав вогняного, тож початково ця стратегія була більш ефективною (показники ефективності 10 проти 9,09), тож і втрата ефективності далі була меншою.

А що ж ситуацією 3? Розглянемо чистий випадок:

Для початку гравцеві необхідно було нанести 2000 од. фізичних і 2000 од. вогняних пошкоджень. Відповідно, для такого результату необхідно $2000/0,8=2500$ од. фізичних і $2000/1=2000$ од. вогняних пошкоджень у чистому вигляді. У атаках це $2500/115=22$ і $2000/100=20$ відповідно, тобто, у сумі 42 атаки. Ефективність бою дорівнює 9,52.

Після адаптації прибульців необхідно нанести $2000/0,6=3334$ од. фізичних пошкоджень і $2000/0,8=2500$ од. вогняних пошкоджень. Для цього необхідно $3334/115=29$ плюс $2500/100=25$ атак, сумарно 54 атаки. Ефективність бою при цьому буде дорівнювати $100/(54/4)=7,41$.

І, власне, що ми маємо? Ефективність впала лише у 1,28.

Висновок з цього виходить сам по собі – використання комбінованих типів пошкоджень виявилось неймовірно ефективним. Окрім того, що сама ефективність у порівнянні з іншими двома ситуаціями впала значно менше, та й вийшло так, що загалом ефективність вийшла найбільшою.

Але візьмемо у розрахунок продовження ситуацій. У першому та другому випадках для підвищення ефективності необхідна лише заміна бійців у групі, взявши інших, що мають здібності з іншими типами пошкоджень. Із ситуації 3 можна зробити продовження. Наприклад, 12 бій. На даний момент масив пошкоджень буде наступним:

[22000; 22000; 0; 0; 0]

Коефіцієнти залишаються незмінними. Але в цьому випадку гравець змінить стиль бою на такий, в якому він буде переважно користуватись вогняними пошкодженнями. Припустимо, що тепер 75 % всіх пошкоджень будуть вогняними. В такому випадку треба нанести $1000/0,6=1667$ од. фізичних пошкоджень і $3000/0,8=3750$ од. вогняних пошкоджень. Сумарна кількість атак,

необхідна для перемоги, – $1667/115=15$ плюс $3750=38$, всього 53. Коефіцієнт ефективності буде дорівнювати 7,55. Це на 1,9 % збільшує ефективність у порівнянні з минулим боєм. Після цього масив пошкоджень має наступний вигляд:

[23000; 25000; 0; 0; 0]

З коефіцієнтами:

[0,48; 0,52; 0; 0; 0]

Змінюється, відповідно, і масив опору:

[39; 20; 0; 0; 0]

Якщо бути більш точним, то перші два показники дорівнюють 39,2 % та 20,8 %, але через правила показники завжди округлюються у меншу сторону, тому і маємо таку таблицю. Фактично, за 12 бій вдалось зменшити показник фізичного захисту на 1 од. (1 % відповідно). На даних етапах з даними числами це може і не надати різниці, але на пізніх стадіях гри це може зіграти важливу роль.

Додамо наступний бій до статистики, маємо:

[24000; 28000; 0; 0; 0]

Коефіцієнти:

[0,46; 0,53; 0; 0; 0]

Додамо ще два бої за таких правил, маємо наступне:

[26000; 34000; 0; 0; 0] і [0,43; 0,56; 0; 0; 0]

Фактично, вступивши у 16 бій гравець матиме супротивника із наступним масивом опору: [37; 22; 0; 0; 0]. У порівнянні з 11 боєм показник фізичного захисту знизився на 3 %, а показник захисту від вогню збільшився на 2 %, причому динаміка йде у сторону зростання захисту від вогню і зменшення фізичного захисту. Відповідно, наступить момент, коли цей показник фізичного захисту може стати меншим. Візьмемо інші перевірки: 16 бій за цією умовою – гравцеві потрібно нанести 1588 од. фізичних пошкоджень і 3847 од. пошкоджень від вогню. Сумарна кількість атак, потрібна для цього, дорівнює $1588/115=14$ плюс $3847/100=38$, тобто 52. Ефективність бою $100/(52/4)=7,69$. Фактично, через використання «слабких місць» самого супротивника і його базовій слабкості до вогняних пошкоджень ефективність за 5 боїв зросла на 3,8 %. Графічно ця картина представлена на рис. 2-4.

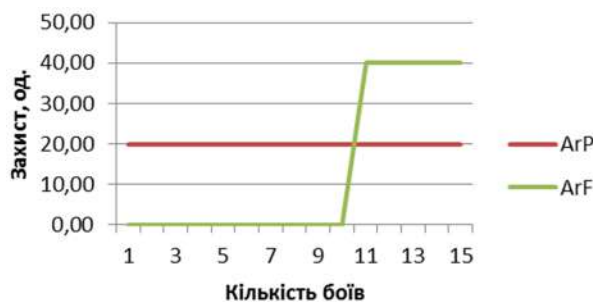


Рис. 2. Графік зміни показників захисту при ситуації 1

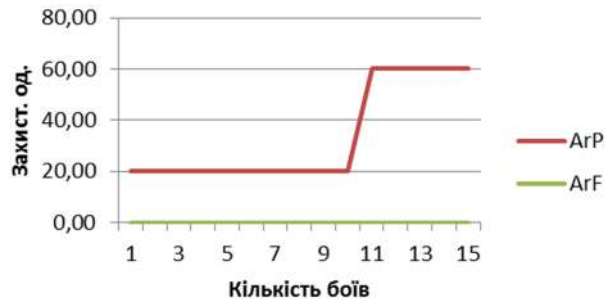


Рис. 3. Графік зміни показників захисту при ситуації 2

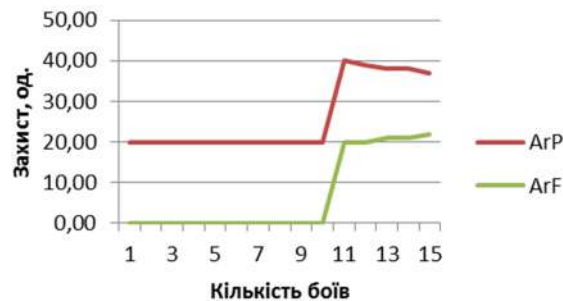


Рис. 4. Графік зміни показників захисту при ситуації 3

Другий механізм працює значно простіше: він використовує статистику нанесених пошкоджень за кожен з ходів та визначає, за яким графіком йде динаміка – зростаюча, спадна або стабільна. У першому випадку це означає, що гравець під час бою готує героїв до стрімких атак, поступово нарощуючи їх силу. Друга навпаки – стрімка атака з перемогою, або ж «жбурляння капцями» у випадку невдачі. Третя – помірний темп бою із плануванням і аналізом ситуації. Відповідно до цього, на відсотки використання певних здібностей накладається коефіцієнт, що збільшує частоту використання того чи іншого типу здібностей у супротивника. Наприклад, при спадній зростаючій динаміці вигідно буде активно атакувати у перші ходи, доки гравець ще не наростив силу героїв. Відповідно, базові характеристики (атакуючі – 40 %; захисні – 20 %; підтримка – 20 %; контроль – 20 %) отримають коефіцієнти 1,75; 0,5; 0,5; 0,5. Це дасть наступний результат: атакуючі – 70 %; захисні – 10 %; підтримка – 10 %; контроль – 10 %, що, в свою чергу, збільшує атакуючий потенціал супротивників, змушуючи гравця або грати від захисту, або змінювати стратегію на агресивну, керуючись тим, що супротивник стає значно менше використовувати захисні та підтримуючі здібності.

3 Висновки

Результатом всього є така система, що початково використовує стиль гри гравця для аналізу стилю гри супротивника, після чого, керуючись отриманими даними про найбільш уживані здібності, корегує супротивників для ефективного опору гравцеві. В подальшому ці дані можна використовувати у дослідженнях в різних галузях, як в інформаційній сфері, так й у соціальній.

References

1. Champandard, A.J.: *AI Game Development: Synthetic Creatures with Learning and Reactive Behaviors*. New Riders, Indianapolis (2003)
2. Millington, I., Funge, J.: *Artificial Intelligence for Games*. Morgan Kaufmann, Amsterdam (2009)
3. Rabin, S.: *AI Game Programming Wisdom*. Charles River Media (2002)
4. Semerikov, S.O., Teplytskyi, I.O., Yechkalo, Yu.V., Kiv, A.E.: *Computer Simulation of Neural Networks Using Spreadsheets: The Dawn of the Age of Camelot*. In: Kiv, A.E., Soloviev, V.N. (eds.) *Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018)*, Kryvyi Rih, Ukraine, October 2, 2018. *CEUR Workshop Proceedings*, vol. 2257, pp. 122–147. <http://ceur-ws.org/Vol-2257/paper14.pdf>. Accessed 30 Nov 2018

Розробка ігор на ігровому рушії Unity для дослідження елементів когнітивного мислення в ігровому процесі

Олександр Олександрович Кацко,
Наталія Володимирівна Моїсеєнко^[0000-0002-3559-6081]

Криворізький державний педагогічний університет,
пр. Гагаріна, 54, м. Кривий Ріг, 50086, Україна
{katsko.aleksander, n.v.moiseenko}@gmail.com

Анотація. *Метою* даного дослідження є розробка за допомогою ігрового рушія Unity ігор для дослідження елементів когнітивного мислення. *Задачі* дослідження: проаналізувати актуальні методи оцінювання когнітивних функцій людини, розробити алгоритми та створити програмну реалізацію оцінювання когнітивних здібностей за допомогою гри. *Об'єкт* дослідження: комп'ютерні ігри. *Предмет* дослідження: дослідження когнітивних здібностей людини в процесі гри. *Результатами* дослідження є розроблені комп'ютерні ігри для дослідження когнітивного мислення людини в процесі гри.

Ключові слова: когнітивне мислення, тестування, розробка комп'ютерних ігор, ігрові рушії, Unity.

Development computer games on the Unity game engine for research of elements of the cognitive thinking in the playing process

Oleksandr O. Katsko and Natalia V. Moiseienko^[0000-0002-3559-6081]

Kryvyi Rih State Pedagogical University, 54, Gagarin Ave., Kryvyi Rih, 50086, Ukraine
{katsko.aleksander, n.v.moiseenko}@gmail.com

Abstract. *The purpose* of this study is development of computer games for research of elements of the cognitive thinking on the Unity game engine. *Objectives of study:* to analyze the actual methods of evaluation of cognitive functions of man, to work out algorithms and to create programmatic realization of evaluation of cognitive capabilities by means of game. *The object of research* are computer games. *The subject of research:* research of elements of the

cognitive capabilities of man in the playing process. *Results of the study* is developed computer games for research of cognitive thinking of man in playing process.

Keywords: cognitive thinking, testing, development of computer games, game engines, Unity.

1 Постановка проблеми

Сучасний світ стає все складніше, його напружений ритм вимагає від нас додаткових розумових зусиль. Не дивно, що у кожній галузі і сфері нашого життя потрібні люди, які володіють потенціалом для успішного подолання зростаючих труднощів. Результати когнітивних вимірів, як і раніше, є одними з найбільш ефективних засобів для розкриття талантів, здібностей або недоліків людини, прогнозування при підборі співробітників тощо. Високий рівень кореляції з якістю і рівнем ефективності дозволяє робити висновки про такі навички особистості, як мислення, міркування, логіка, вирішення проблем і здатність до навчання. Ці поняття (які часто називають «g-фактором») дають довгостроковий і точний опис людини і вимірюються переважно невербальними методами.

Оцінку когнітивних здібностей або загального рівня розумових здібностей (GMA) найкраще проводити спільно з оцінкою особистості, щоб отримати точний психологічний портрет людини. Існують різні типи оцінок розумових здібностей. Використання тієї чи іншої оцінки залежить від поставленої задачі. Адаже методи оцінки розділяються на комплексні, коли оцінюються дві або більше когнітивних здібностей, та конкретні, коли оцінюється лише одна когнітивна здібність. Отже, такі тести можуть оцінювати загальні розумові здібності, логічне мислення, аналітичні здібності, здатність до абстрактного мислення, математичні здібності, вербальні здібності, зорову уяву, прийняття рішень і розважливості, швидкість обробки інформації, швидкість навчання і пізнавальні здібності тощо.

У даний час в якості засобів оцінки розроблено багато тестів і проб. Різні методики дозволяють перевірити стан всіх сфер вищої нервової діяльності. Одним із засобів оцінювання когнітивних функцій людини є комп'ютерні ігри. Аналіз сучасного стану проблеми дослідження когнітивного мислення за допомогою комп'ютерних засобів приводить до висновку що це актуальна тема, адже на сьогодні для дослідження когнітивного мислення комп'ютерні засоби використовуються здебільшого у вигляді тестів.

2 Обговорення результатів

У процесі виконання роботи було створено дві комп'ютерні гри «The Maze» (рис. 1) та «Мемогу» (рис. 2, 3) за допомогою ігрового рушія Unity. Вони слугують інструментами дослідження когнітивного мислення людини: в «The

«Maze» вимірюються показники якості обробки інформації та стратегічного мислення, а в «Меморі» – показники пам'яті.

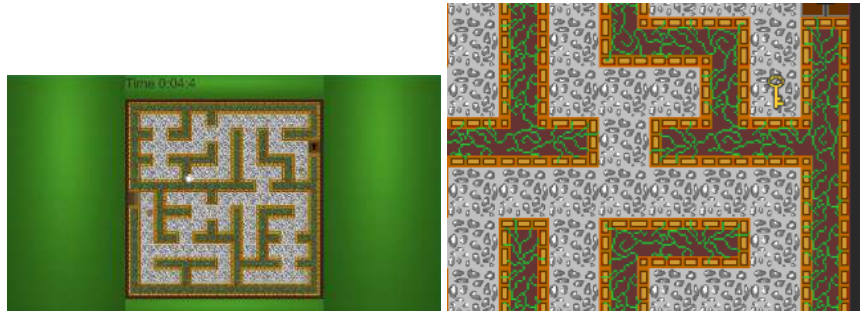


Рис. 1. Інтерфейс гри «The Maze»



Рис. 2. Інтерфейс гри «Memory»

Було обрано та реалізовано метод дослідження когнітивного мислення людини. Завдяки цьому методу вдалося відтворити когнітивні функції у математичній формі. За допомогою відповідних математичних операцій визначені показники когнітивного мислення, які потрібні були для подальшого дослідження.

Метод дослідження когнітивного мислення було запропоновано в статті А. Ю. Ківа та ін. [1]. Було запропоновано модель простору мислення, в якій розумові процеси розглядаються як накопичення кроків мислення або елементів мислення. Диференціальні рівняння сформульовані для процесів мислення, відповідних конкретних видів професійної діяльності. На основі математичної моделі процесів мислення були розроблені нові комп'ютерні тести для тестування творчого мислення оператора.



Рис. 3. Реєстрація ефективних ходів (ES) в грі «Memory»

Короткий опис моделі простору мислення (TS) було зроблено в роботі [2]. Простір мислення містить дискретні елементи мислення, і кожен з них відповідає заданому етапу мислення людини в процесі його переходу до вирішення проблеми. Кроки мислення можна розділити в різних групах на різновиди моделі. Зокрема, це можуть бути три групи: ефективні кроки (ES), неправильні кроки (WS) і проміжні етапи (IS). В цьому випадку можуть бути записані три диференціальних рівняння, що описують «кінетику» кожного типу кроків. Такі рівняння відомі в природничих науках, і багато їх типів добре вивчені математично.

Під час проходження ігор людиною програмно замірюються ефективні ходи, помилкові ходи, час. Людина в цей самий час просто грає в комп'ютерну гру та отримує задоволення і гадки не має, що всі показники в ході гри записуються. Це є дуже важливим моментом, адже, якщо людина дізнається, що її тестують, вона навмисно починає змінювати модель своєї поведінки, рухів, реакції, уваги. Для досягнення поставленої мети потрібно було пропонувати людям просто пограти в гру, щоб вони нічого не здогадались.

У кожній з цих комп'ютерних ігор є алгоритм, згідно якого вимірюються всі потрібні показники. Так, для гри «Memory» реєстрація цих показників здійснюється наступним чином:

1. Після того, як гравець починає гру і переходить від головного меню до безпосередньої гри, починається відлік часу, який триватиме до самого завершення гри.
2. Другий показник, який вимірюється під час гри, – це кількість хибних ходів. Таким вважається хід, коли попередня і наступна карта не співпали. До кількості хибних ходів в такому випадку додається два.

3. Наступним показником, який вимірюється, є кількість ефективних ходів. Хід в цій грі вважається ефективним, якщо знайдено попарно однакові карти, але зараховується як дві. Це пояснюється тим, що при перегортанні карти є шанс, що наступною картою буде така сама. Отже, для кожного випадку успішного проходження гри реєструється однакова кількість ефективних ходів, а саме 18 – стільки, скільки загалом перегорнутих карт. Наочно побачити, як реєструються ефективні ходи в грі «Методу», можна на рис. 3.

Для гри «The Maze» реєстрація показників для дослідження здійснюється дещо складнішим способом. В основу реєстрації ефективних та хибних ходів ліг алгоритм пошуку найкоротшого шляху. Після побудови ігрового рівня, за допомогою алгоритму вираховується найкоротший шлях до ключа та від ключа до дверей. Реєстрація показників здійснюється наступним чином:

1. Після того, як гравець починає гру і переходить від головного меню до безпосередньої гри, починається відлік часу, який триватиме до самого завершення гри.
2. Наступним показником, який вимірюється, є кількість ефективних ходів. Для зручності підрахунку ефективних та хибних ходів ігрове поле було програмно поділено на матрицеподібну сітку. Хід в цій грі вважається ефективним, якщо головний герой рухається по лінії найкоротшого шляху до ключа, а потім до дверей.

3 Висновки

1. Шляхом аналізу отриманих в ході математичного розрахунку показників когнітивного мислення було порівняно показники різних вікових груп та виділено загальні тенденції щодо когнітивного мислення кожної з них.
2. Було знайдено показники якості сприймання інформації та критичного мислення, що є дуже важливим для порівняльного аналізу людей певних вікових груп. Порівняння показників конкретної людини з визначеними показниками відповідної вікової групи дає можливість зробити висновки щодо цієї людини.
3. Також розроблені ігри можна використовувати в якості тренажера для розвитку когнітивних здібностей.

References

1. Kiv, A.E., Orischenko, V.G., Tavalika, L.D., Holmes, S.: Computer testing of operator's creative thinking. *Computer Modelling & New Technologies*. 4(2), 107–109 (2000)
2. Kiv, A.E., Orischenko, V.G., Polozovskaya, I.A., Zakharchenko, I.G. Computer Modelling of the Learning Organization. In: Kidd, P.T., Karwowski, W. (eds.) *Advances in Agil Manufacturing*, pp. 553–556. IOS Press, Amsterdam (1994)

Розробка графічної оболонки для програми розрахунків фізичних характеристик твердих тіл

Владислав Сергійович Кузнецов,
Наталія Володимирівна Моїсеєнко^[0000-0002-3559-6081]

Криворізький державний педагогічний університет,
пр. Гагаріна, 54, м. Кривий Ріг, 50086, Україна
{ayron1705, n.v.moiseenko}@gmail.com

Анотація. *Метою* даного дослідження є розробка графічної оболонки для програми розрахунків фізичних характеристик твердих тіл. *Задачі* дослідження: проаналізувати програмне забезпечення для розрахунків фізичних характеристик твердих тіл, що існує; розробити алгоритм роботи та його програмну реалізацію графічної оболонки для програми розрахунків фізичних характеристик твердих тіл. *Об'єкт* дослідження: програми розрахунків фізичних характеристик твердих тіл. *Предмет* дослідження: графічна оболонка для програми розрахунків фізичних характеристик твердих тіл. *Результатом* роботи є розроблена графічна оболонка для програми розрахунків фізичних характеристик твердих тіл.

Ключові слова: тверде тіло, фізичні характеристики, програма для розрахунків, графічна оболонка.

Development graphic shell for the program calculations of physical properties of solids

Vladyslav S. Kuznietsov and Natalia V. Moiseienko^[0000-0002-3559-6081]

Kryvyi Rih State Pedagogical University, 54, Gagarin Ave., Kryvyi Rih, 50086, Ukraine
{ayron1705, n.v.moiseenko}@gmail.com

Abstract. *The purpose* of this study is development of graphic shell for the program calculations of physical properties of solids. *Objectives of study:* to analyze software for the calculations of physical properties of solids, that exists; to work out the algorithm and programmatic realization of graphic shell for the program of calculations of physical properties of solids. *The object of research* are programs of calculations of physical properties of solids. *The subject of research:* graphic shell for the program of calculations of physical properties of

solids. *Results of the study* is developed graphic shell for the program of calculations of physical properties of solids.

Keywords: solid state, physical properties, the program for calculations, graphic shell.

Пошуки методів мінімізації електронних схем призвели до відкриття низки надзвичайних фізичних явищ, які показали, що індивідуальні молекули можуть виступати у ролі провідників, діодів, транзисторів, логічних та запам'ятовуючих пристроїв. Основними перевагами використання молекул у ролі прототипів елементів електронних схем є їх досить малий розмір, що складає всього декілька нанометрів, легкий та дешевий спосіб синтезу молекулярних структур, а також виявлення зовсім нових вольт-амперних характеристик, що не мають аналогів у напівпровідниковій електроніці. Тому поряд з актуальністю практичного застосування як окремих молекул, так і молекулярних комплексів для створення, передачі, прийому і обробки інформації, побудови так званих молекулярних комп'ютерів, проблема дослідження молекулярних наноструктур є самостійною фундаментальною проблемою фізики конденсованого стану, оскільки властивості таких структур істотно відрізняються від класичних напівпровідникових об'єктів.

Обчислення повної енергії та моделювання молекулярної динаміки з використанням теорії функціоналу електронної густини представляють надійний інструмент в фізиці твердого тіла, фізиці матеріалів, хімічній фізиці і фізичній хімії. Множина різних застосунків для систем типу молекул, об'єму матеріалів та поверхонь довели потужність цих методів в аналізі та прогнозуванні рівноважних та нерівноважних властивостей. *Ab initio* моделювання молекулярної динаміки дозволяє аналізувати рух атомів та точно розраховувати такі термодинамічні властивості як вільна енергія, константи дифузії та температура плавлення матеріалів.

Для розрахунків електронної структури матеріалів нині широко застосовуються неемпіричні, так звані *ab initio* методи. За їх допомогою проводяться розрахунки розподілу електронної густини, повної енергії системи, щільності станів, енергетичних і структурних параметрів твердих тіл.

Методи *ab initio* часто використовуються в поєднанні з класичним методом молекулярної динаміки, який протягом тривалого часу використовується в теорії твердого тіла для дослідження атомних конфігурацій і реальної структури речовини. Цей метод дозволяє досліджувати атомні системи з великою кількістю частинок і спостерігати поведінку системи протягом великих часових інтервалів.

У роботі створено графічну оболонку для пакету програм *fhimd*, розробленого для дослідження властивостей великих систем. Пакет програм *fhimd* надає можливість обчислення статичної повної енергії або *ab initio* моделювання молекулярної системи.

Розрахунки базуються на ітеративному методі отримання параметрів основного стану електронів. Псевдопотенціали у формі Клейнмана та Біландера використовуються для опису потенціалу ядер та електронів остову. Обмін та

кореляція описуються в наближенні локальної електронної густини або в різних апроксимаціях «універсального градієнта». Рівняння руху ядер інтегруються за стандартною схемою молекулярної динаміки. Крім того, ефективна оптимізація структури може бути виконана за спрощеною схемою варіаційним методом.

Для введення початкових даних для стартової утіліти fhstart потрібні два вхідних файли. Файл inp.mod містить головним чином керуючі параметри для запуску, наприклад, часовий крок для мінімізації електронних та атомних схем, критерії збіжності, максимальна кількість кроків та ін.

Вигляд файлу вхідних даних inp.mod:

```
-1 100 1000000    : nbeg iprint timequeue
100 2           : nomore nomore_init
6.0 0.2         : delt gamma
4.0 0.2 0.0001  : delt2 gamma2 eps_chg_dlt
400 2           : delt_ion nOrder
0.0 1.0         : pfft_store mesh_accuracy
2 2             : idyn i_edyn
0 .false.       : i_xc t_postc
.F. 0.001 .F. 0.002 : trane ampre tranp amprp
.false. .false. .false. 1800 : tfor tdyn tsdp nstepe
.false.         : tdipol
0.0001 0.0005 0.2 : epsel epsfor epsekinc
0.01 0.001 3     : force_eps(1) force_eps(2) max_no_force
3               : init_basis
```

Файл start.inp описує геометрію суперкомірки та конфігурацію ядер. він також містить інформацію, яка відноситься до молекулярно-динамічного моделювання, оптимізації структури або обчислення основного електронного стану.

Файл start.inp містить всю інформацію про структуру: геометрію суперкомірки та позиції ядер.

Розроблена графічна оболонка є зручним інтерфейсом між програмою розрахунків фізичних характеристик твердих тіл [1; 2] fhimd та користувачем (рис. 1).

Для програмної реалізації графічної оболонки було розроблено систему класів мовою C# в середовищі Microsoft Visual Studio.

Оскільки програма fhimd потребує введення даних шляхом запису їх у текстові файли у визначеному порядку, то головною задачею оболонки є надання зручного способу введення вхідних даних для програми розрахунків характеристик твердих тіл fhimd і автоматичного запису їх в задані файли. Програма має два режими: створення вхідних файлів та редагування вже існуючих.

У процесі введення даних відбувається перевірка на коректність та запобігання введення неактуальних даних. Наприклад, якщо не вибраний режим молекулярної динаміки, то не потрібно вносити дані, які використовуються для цього режиму, як то pros, nseed, nthm, Q та ін. (рис. 2, 3).

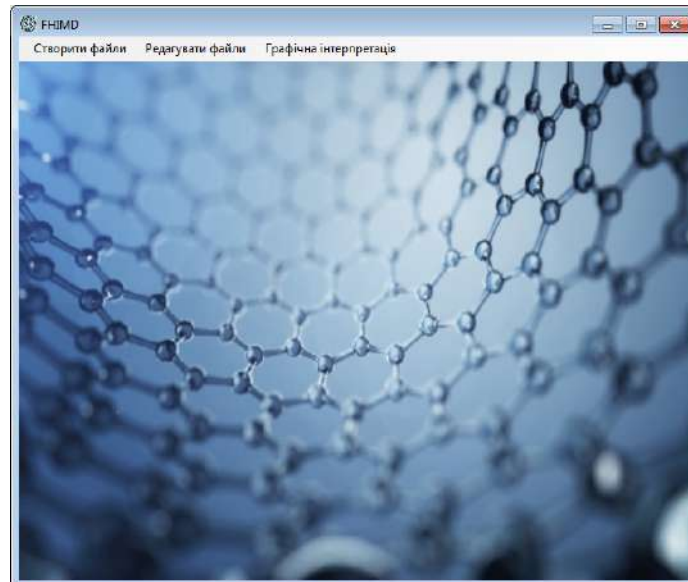


Рис. 1. Головне вікно програми fhimd

Параметр	Значення	Опис
trane	<input type="radio"/> true <input checked="" type="radio"/> false	збурення початкових хвильових функцій значенням ampre
ampre	0,001	амплітуда довільного збурення додається до хвильової функції
tranp	<input type="radio"/> true <input checked="" type="radio"/> false	чи можна переміщати іони
amprp	0,002	амплітуда довільного збурення додається до іонних позицій
tfor	<input type="radio"/> true <input checked="" type="radio"/> false	якщо встановлено в .true. іонні позиції релаксують
tdyn	<input type="radio"/> true <input checked="" type="radio"/> false	якщо встановлено в .true. виконується МД моделювання
tsdp	<input type="radio"/> true <input checked="" type="radio"/> false	схема оптимізації структури: T-модифік. схема покрокового спуску, F-спрощена схема динаміки
nstepe	1800	максимальна кількість електронних ітерацій, для того щоб зійшлися сили
tdipol	<input type="radio"/> true <input checked="" type="radio"/> false	якщо встановлено в .true. обчислюється корекція поверхневого диполю
epsel	0,0001	середня зміна енергії менш ніж epsel для останніх трьох ітерацій
epsfor	0,0005	іонні сили менше ніж epsfor; є активним, тільки якщо tfor та tford - Т.
epsekinc	0,2	середня зміна хвильових функцій для останніх трьох ітерацій менш ніж epsekinc
force_eps(1)	0,001	максимально припустима відносна зміна в локальних силах
force_eps(2)	0,001	максимально припустима відносна зміна в глобальних силах перед зміщенням іонів
max_no_force	3	макс. число електронних ітерацій для якого на іонному кроці не обчисл. локальні сили
init_basis	3	тип базису, що використовується в ініціалізації

Зберегти

Рис. 2. Форма для створення вхідного файлу inp.mod

start.inp

Загальні | Параметри | Атоми

npres кількість процесорів (PE)

minpres мінімальне число PE

ngprx число PE в групі

nsp кількість сортів атомів

nel_exc кількість надлишкових електронів

n_empty кількість незанихтих станів

ibrav 0 тип комірки 0-замовч 1 проста куб., 2 гранецентр. 3 об'ємноцентр 8 ромб

pginde індекс точкової групи: 0-центр та симетрії, 1-немає симетрії

celldm параметри решітки суперкомірки; celldm(1) константа решітки (бор)

nkpt кількість k-точок

xk(1..3) координати k-точок

wkpt вага k-точок

i_facs(1..3) фактор згортки k-точки: 1 1 1 відповідає відсутності згортки

t_kpoint_rel true false true - вектори зворотної решітки, false - Декартові координати

Зберегти

Рис. 3. Форма для створення вхідного файлу start.inp

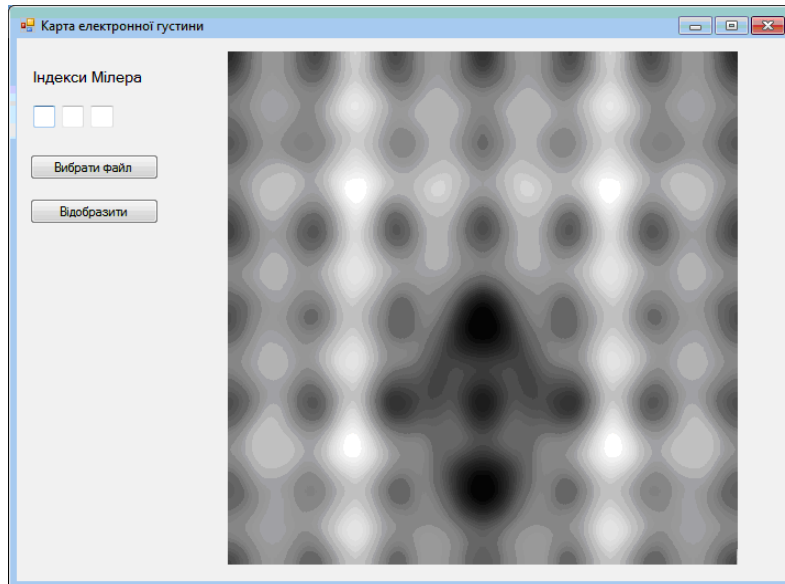


Рис. 4. Форма для графічного відображення карти електронної густини

Після натиснення кнопки «Зберегти» відкривається діалогове вікно, в якому можна вибрати папку та ім'я для зберігання файлу.

У режимі редагування відкривається діалогове вікно для вибору файлу, який треба відкрити, та дані завантажуються в відповідні форми.

Також оболонка містить інструмент для побудови та перегляду карт електронної густини. Оскільки електронна густина обчислюється для багатьох точок і зберігається в окремому файлі, необхідно мати можливість продивлятися карти електронної густини як переріз в заданій індексами Міллера площині (рис. 4).

Пункт меню «Графічна інтерпретація» надає можливість обрати потрібний файл електронної густини, задати індекси Міллера та побудувати карту електронної густини.

References

1. Car, R., Parrinello, M.: Unified Approach for Molecular Dynamics and Density-Functional Theory. *Phys. Rev. Lett.* **55**(22), 2471–2474 (1985). doi:10.1103/PhysRevLett.55.2471
2. Car, R., Parrinello, M.: The unified approach to density functional and molecular dynamics in real space. *Solid State Communications.* **62**(6), 403–405 (1987). doi: 10.1016/0038-1098(87)91043-X

Розробка адаптивного навчального програмного забезпечення з теми «Дробові числа» для учнів 5 класу

Маргарита Євгеніївна Петрова¹, Михайло Михайлович Мінтій¹,
Сергій Олексійович Семеріков¹^[0000-0003-0789-0272],
Наталія Павлівна Волкова²^[0000-0003-1258-7251]

¹ Криворізький державний педагогічний університет,
пр. Гагаріна, 54, м. Кривий Ріг, 50086, Україна
{reetta.me14, mikhail.mintii9, semerikov}@gmail.com

² Університет імені Альфреда Нобеля,
вул. Січеславська Набережна, 18, м. Дніпро, 49000, Україна
npvolkova@yahoo.com

Анотація. *Об'єкт дослідження* – розробка веб-орієнтованого програмного забезпечення для підтримки навчання математики у закладах загальної середньої освіти. *Предмет дослідження* – розробка адаптивного навчального програмного забезпечення з теми «Дробові числа» для учнів 5 класу закладів загальної середньої освіти. *Мета дослідження* – теоретично обґрунтувати, розробити та експериментально перевірити адаптивне навчальне програмне забезпечення з теми «Дробові числа» для учнів 5 класу закладів загальної середньої освіти. *Завдання дослідження:* 1) проаналізувати стан проблеми розробки та використання адаптивного програмного забезпечення для навчання математики; 2) обґрунтувати вибір засобів розробки адаптивного навчального програмного забезпечення для учнів середньої школи; 3) розробити та експериментально перевірити адаптивне навчальне програмне забезпечення з теми «Дробові числа» для учнів 5 класу закладів загальної середньої освіти. Результати експериментального навчання з використанням розробленого програмного забезпечення показали, що виконана розробка не урахує зміну індивідуальних особливостей учня (у тому числі психологічних) у процесі навчання. Це визначає наступні шляхи його розвитку, спрямованого на підвищення рівня адаптивності: індивідуалізація навчальних впливів; формалізація та реалізація моделі учня; урахування процесуальних характеристик навчання у моделі учня.

Ключові слова: ІКТ у навчанні математики, дробові числа, адаптивне програмне забезпечення.

Development of adaptive educational software on the topic of “Fractional Numbers” for students in grade 5

Marharyta Ye. Petrova¹, Mykhailo M. Mintii¹, Serhiy O. Semerikov¹[0000-0003-0789-0272]
and Nataliia P. Volkova²[0000-0003-1258-7251]

¹ Kryvyi Rih State Pedagogical University, 54, Gagarin Ave., Kryvyi Rih, 50086, Ukraine
{reetta.me14, mikhail.mintii9, semerikov}@gmail.com

² Alfred Nobel University, 18, Naberezhna Sicheslavska St., Dnipro, 49000, Ukraine
npvolkova@yahoo.com

Abstract. The *object of the research* is the development of Web-based software to support mathematics education in general secondary education institutions. The *subject of the research* is the development of adaptive educational software on the subject of “Fractional numbers” for students in grade 5 of general secondary education institutions. The *purpose of the study* is to theoretically substantiate, develop, and experimentally test adaptive learning software on the topic “Fractional numbers” for students in grade 5 of general secondary education institutions. The *objectives of the study*: 1) to analyze the state of the problem of developing and using adaptive software for teaching mathematics; 2) to substantiate the choice of development tools for adaptive educational software for secondary school students; 3) to develop and experimentally test adaptive educational software on the subject of “Fractional numbers” for students in grade 5 of general secondary education institutions. The *results of experimental training* using the developed software have shown that the development is not taking into account the change in the individual characteristics of the student (including psychological) in the learning process. This determines the following ways of its development, aimed at improving the level of adaptability: individualization of learning effects; formalization and implementation of the student model; taking into account the procedural characteristics of training in the student model.

Keywords: ICT in teaching mathematics, fractional numbers, adaptive software.

1 Вступ

Ключова реформа Міністерства освіти і науки України – «Нова українська школа», головна мета якої – створити школу, у якій буде приємно навчатись і яка даватиме учням не тільки знання, як це відбувається зараз, а й вміння застосовувати їх у житті [20]. Серед 10 ключових компетентностей нової української школи важливе місце займає математична компетентність, що включає в себе культуру логічного і алгоритмічного мислення, уміння застосовувати математичні (числові та геометричні) методи для вирішення

прикладних завдань у різних сферах діяльності, здатність до розуміння і використання простих математичних моделей та уміння будувати такі моделі для вирішення проблем [13, с. 11].

Внеском математичної компетентності:

- у формування компетентності спілкування державною мовою є уміння, що виробляється в процесі навчання математики, – лаконічно та зрозуміло формулювати думку, аргументувати, доводити правильність тверджень;
- у компетентність спілкування іноземними мовами – зіставляти математичний термін чи буквене позначення з його походженням з іноземної мови;
- в основні компетентності у природничих науках і технологіях – моделювати процеси, що відбуваються в навколишньому світі;
- в інформаційно-цифрову компетентність – діяти за алгоритмом та складати алгоритми;
- у компетентність уміння вчитися – доводити правильність певного судження та власної думки;
- у компетентність ініціативність і підприємливість – здійснювати раціональний вибір;
- у соціальну та громадянську компетентності – робити висновки з отриманих результатів розв’язування задач соціального змісту;
- в обізнаність та самовираження у сфері культури – естетично зображувати фігури, графіки, рисунки;
- в екологічну грамотність і здорове життя – ощадливо користуватися природними ресурсами [21, с. 15].

Зміст ключових компетентностей нової української школи було сформульовано [20] з урахування Рекомендацій Європейського Парламенту та Ради Європи щодо формування ключових компетентностей освіти впродовж життя [24], згідно яких знання математики – це можливість розвивати та застосовувати математичне мислення для розв’язання ряду задач в повсякденних ситуаціях. «При виробленні здібності до кількісного мислення сам процес та діяльність важливі так само, як і знання. ... Необхідні знання з математики включають глибокі знання чисел, мір та структур, основних математичних дій та їх демонстрація, розуміння математичних термінів та понять і знання задач, які можна розв’язати за допомогою математики. Особа повинна вміти застосовувати основні математичні принципи та прийоми у побуті, на роботі та вдома» [24].

У порадику для вчителя нової української школи визначається два основних напрями підтримки учнів у навчальному середовищі:

1. *модифікації* як зміни до навчального змісту, що стосуються або його скорочення, або зміни концептуальної складності навчального завдання;
2. *адаптації*, що, на відміну від модифікацій, змінюють характер подання навчального матеріалу, не змінюючи зміст чи концептуальну складність навчального завдання.

Зокрема, можуть використовуватись такі види адаптації, як зміни в навчальних

підходах (використання навчальних завдань різного рівня складності; збільшення часу на виконання навчального завдання, зміна темпу занять, чергування видів діяльності тощо); адаптація навчальних матеріалів (адаптація навчальних посібників, наочних та інших матеріалів тощо) [21, с. 154-155].

Щоденна діяльність з математики учня нової української школи передбачає застосування ротаційної моделі «Щоденні 3» (математика) – щоденні діяльності, виконуючи які, діти навчаються бути самостійними під час математики, в той час як учитель має можливість працювати з учнями індивідуально та у малих групах [21, с. 203]. Модель включає три компоненти (1 – математика самостійно; 2 – математика разом; 3 – математичне письмо) діяльності вчителя із залучення дітей до математичних ігор, а також спонукає брати участь у діяльності і вирішенні проблем, використовуючи інструменти для лічби чи ігрові матеріали. Дана діяльність може також передбачати використання комп'ютерів, калькуляторів або планшетів, як засобів для самостійної роботи [21, с. 99].

Комп'ютерна підтримка математичної діяльності засобами комп'ютерних технологій надає можливість за допомогою засобів комп'ютерної візуалізації [4], управління досліджуваними об'єктами самостійно обирати методи та форми організації навчання з метою його індивідуалізації, підвищення мотивації та інтересу до навчання. Уведення до програм ігрових ситуацій надає учням можливість, поставивши цікаве завдання, пропонувати будь-які рішення проблеми, проявити оригінальність розв'язку тощо.

Отже, розробка адаптивного навчального програмного забезпечення з теми «Дробові числа» для учнів 5 класу закладів загальної середньої освіти є актуальною проблемою.

2 Апарат дослідження

Мета дослідження – теоретично обґрунтувати, розробити та експериментально перевірити адаптивне навчальне програмне забезпечення з теми «Дробові числа» для учнів 5 класу закладів загальної середньої освіти.

Завдання дослідження:

1. Проаналізувати стан проблеми розробки та використання адаптивного програмного забезпечення для навчання математики.
2. Обґрунтувати вибір засобів розробки адаптивного навчального програмного забезпечення для учнів середньої школи.
3. Розробити та експериментально перевірити адаптивне навчальне програмне забезпечення з теми «Дробові числа» для учнів 5 класу закладів загальної середньої освіти.

Об'єкт дослідження – розробка веб-орієнтованого програмного забезпечення для підтримки навчання математики у закладах загальної середньої освіти.

Предмет дослідження – розробка адаптивного навчального програмного забезпечення з теми «Дробові числа» для учнів 5 класу закладів загальної середньої освіти.

Методи дослідження: аналіз – для виявлення теоретичних основ дослідження; проектування та тестування – для розробки програмного забезпечення.

Практичне значення одержаних результатів. Матеріали проведеного дослідження стануть у нагоді для навчання дітей математики у закладах загальної середньої освіти та полегшення роботи вчителів.

3 Теоретичні основи розробки та використання адаптивного програмного забезпечення для навчання математики

3.1 Поняття про адаптивне програмне забезпечення

Автоматичне пристосування системи навчання до зміни зовнішніх умов та збереження здатності до ефективного досягнення дидактичних цілей при зміні характеристик учня називається адаптивністю. Адаптивним системам властиві, зокрема, спрямованість програми на досягнення певної мети, одержання у процесі роботи інформації про зовнішні умови та застосування її для зміни власної поведінки [18].

Адаптивною навчальною програмою називається навчальна програма, в якій послідовність подання і характер викладення навчального матеріалу залежать від історії навчання (даних про час виконання завдань і допущені помилки) та індивідуальних особливостей учня, у тому числі психологічних. Реалізація адаптивної навчальної програми неможлива без розв'язання задач діагностики (рівня знань, психологічних особливостей учня, тощо) та оптимізації (зокрема, вибору наступного навчального впливу) [17].

Задача адаптації навчального процесу розв'язується за допомогою адаптивних автоматизованих навчальних систем та експертних навчальних систем, що генерують такі регулятивні впливи (способи подання матеріалу, приклади, підказки), сприймання яких учнем приводить до стабілізації або підвищення поточних оцінок успішності засвоєння ним змісту навчального матеріалу. Використовуючи такі системи, учитель у процесі навчання здійснює накопичення і опрацювання даних, необхідних для визначення ефективного регулятивного впливу, після чого подання навчального матеріалу адаптується до індивідуальних особливостей учня [14, с. 69].

Теоретичною основою побудови адаптивних автоматизованих навчальних систем є програмоване навчання, яке Г. О. Балл визначав як тип навчання, що здійснюється у відповідності до навчальної програми, яка контролює як обсяг знань, умінь і навичок, якими повинні оволодіти учні, так й спосіб організації їх навчальної діяльності [11, с. 420]. Це досягається шляхом поділу навчального матеріалу на окремі порції та інтенсивного обміну відомостями між учнем та навчальною програмою, що здійснюється переважно у формі «питання-відповідь».

У програмованому навчанні використовуються лінійні навчальні програми з фіксованою послідовністю порцій навчального матеріалу та розгалужені

навчальні програми, у яких послідовність порцій навчального матеріалу залежить від відповідей учня на завдання. Слід зазначити, що програмоване навчання може бути реалізованим не лише за допомогою програмних засобів, а й за допомогою програмованих посібників у традиційній формі – спеціально оформлених і складених книг [11].

Г. О. Балл відмічає, що застосування програмованого навчання у багатьох випадках не виправдало сподівань на різке підвищення ефективності освіти, але застосування комп'ютерів для реалізації програмованого навчання дозволить уникнути багатьох притаманних йому обмежень та недоліків [11, с. 422-423].

Таким чином, розроблюване програмне забезпечення повинно надавати можливості:

- вивчення дробових чисел;
- розподілу уроків;
- виконання багатьох дій над дробовими числами;
- виконання вправ на запам'ятовування;
- розв'язання задач.

Виходячи із поставлених задач та сформульованих вимог до програмного засобу, було виділені такі етапи його розробки:

1. Ознайомлення з методикою проведення заняття з математики у школі в 5 класі.
2. Вивчення існуючих програмних продуктів, виявлення їх переваг та недоліків.
3. Проектування власного програмного продукту.
4. Експериментальна перевірка (тестування) розробленого програмного забезпечення.

3.2 Програмоване навчання теми «Дробові числа» у 5 класі

У навчальній програмі з математики для 5-9 класів загальноосвітніх навчальних закладів [19] вказується, що у 5-6 класах відбувається поступове розширення множини натуральних чисел до множини раціональних чисел шляхом послідовного введення дробів (звичайних і десяткових). У 5 класі тема «Дробові числа і дії з ними» (60 год) передбачає опанування наступних підтем:

1. Звичайні дроби. Правильні та неправильні дроби. Звичайні дроби і ділення натуральних чисел. Мішані числа.
2. Порівняння звичайних дробів з однаковими знаменниками.
3. Додавання і віднімання звичайних дробів з однаковими знаменниками.
4. Десятковий дріб. Запис десяткових дробів. Порівняння десяткових дробів. Округлення десяткових дробів.
5. Арифметичні дії з десятковими дробами.
6. Відсотки.
7. Середнє арифметичне. Середнє значення величини.

Програмні вимоги до математичної компетентності учнів із дробових чисел формулюються у такий спосіб:

- *наводить приклади*: звичайних і десяткових дробів;
- *розрізняє*: звичайні і десяткові дроби; правильні і неправильні дроби;
- *пояснює, що таке*: чисельник і знаменник дроби; мішане число;
- *читає і записує*: звичайні та десяткові дроби; мішані числа;
- *формулює означення*: правильного і неправильного дроби; відсотка; середнього арифметичного;
- *розв'язує вправи, що передбачають*: порівняння, додавання і віднімання звичайних дробів з однаковими знаменниками; порівняння, округлення, додавання, множення і ділення десяткових дробів; перетворення мішаного числа у неправильний дріб; перетворення неправильного дроби в мішане число або натуральне число; знаходження відсотка від числа та числа за його відсотком; знаходження середнього арифметичного кількох чисел, середнього значення величини;
- *розв'язує сюжетні задачі з реальними даними щодо*: використання природних ресурсів рідного краю; безпеки руху; знаходження периметрів та площ земельних ділянок, підлоги класної кімнати, об'єму об'єктів, що мають форму прямокутного паралелепіпеда; розрахунку сімейного бюджету, можливості здійснення масштабних покупок; розрахунків, пов'язаних із календарем і одиницею тощо [19, с. 17].

Наприкінці 1960-х рр. під керівництвом Г. Ф. Рамлоу (Harold F. Rahmlow) була розроблена серія з 21 посібника для професійно спрямованого програмованого навчання математики [5]:


1. Symbols
2. Representing Numbers by Letters
3. Equivalent Forms (Reduction of Fractions)
4. Ratios and Fractions
- 5A. Addition of Fractions
- 5B. Subtraction of Fractions
6. Multiplication of Fractions
7. Division of Fractions
8. Concepts of Decimals and Fractions
9. Addition and Subtraction of Decimals
10. Multiplication of Decimals
11. Division of Decimals
12. Conversion of Fractions into Decimals
13. Equivalent Forms of $A=BC$
14. Solutions of $A=BC$
15. Percentage
16. Commutative Law
17. Reciprocals
18. Scientific Notation
19. Proportions
20. Concepts of Number Bases

Співставлення змісту навчальної програми [19] та [5] показує, що дробові

числа утворюють важливу змістову лінію (рис. 1, зліва).

Так, при програмованому навчанні раціональних та дробових чисел за американським посібником програмні вимоги конкретизуються до наступних:

1. The student should be able to demonstrate his recognition of fractions of the form a/b where a and b are letters or positive integers less than 100.
2. The student should be able to demonstrate his knowledge of the terms numerator and denominator.
3. The student should be able to demonstrate how shaded areas of plane figures can be represented by fractions.

(e.g. The shaded area of the figure  represents $1/4$ of figure.

4. The student should be able to demonstrate his knowledge of the relationship between a ratio and a fraction.

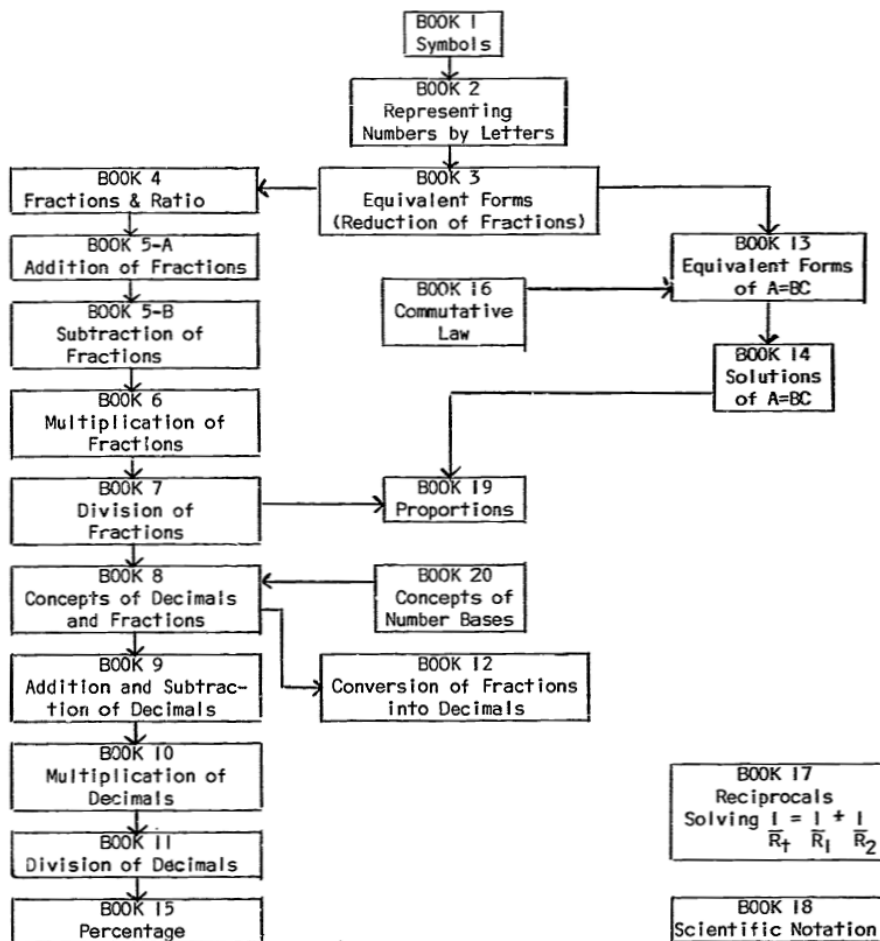


Рис. 1. Структура професійно спрямованого навчання математики (за [5])

Адаптивність навчання у відповідному посібнику [6] забезпечується формуванням індивідуальної для кожного учня навчальної траєкторії шляхом вибору відповідей на запитання, згідно яких виконуються відповідні гіпертекстові переходи. На рис. 2 показано перші фрейми із [6].

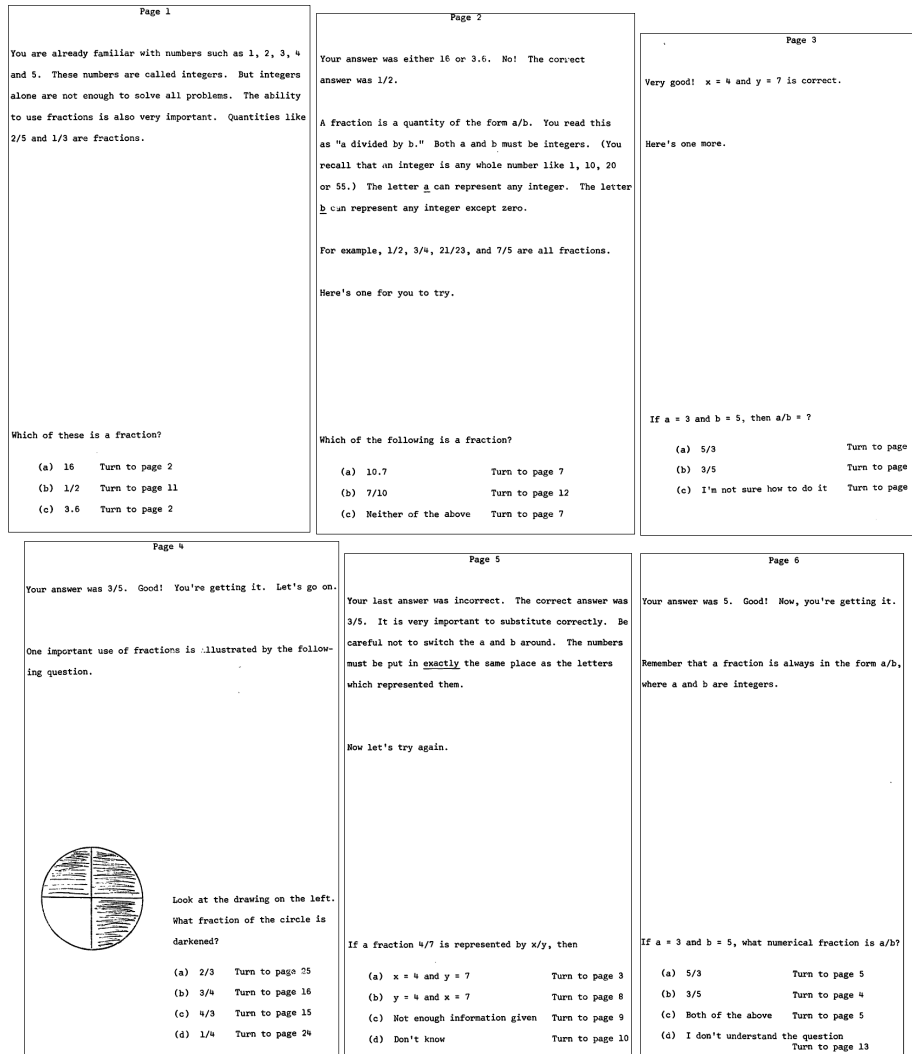


Рис. 2. Початок програмованого посібника [6]

Г. Ф. Рамлоу та Ю. Дж. Керр (Eugene G. Kerr) у статті 1969 року [7] вказують, що розроблена серія посібників призначена для комп'ютерно-орієнтованого навчання математики (Computer Assisted Instruction – CAI). На рис. 3 наведено послідовність гіперзв'язків між фреймами (R – правильна відповідь, W – неправильна), спрямовану на досягнення навчальної цілі: «Учень повинен бути

зданен додавати два чи більше числових дроби у формі a/b , де $0 < (a, b) < 100$ (наприклад, $\frac{1}{2} + \frac{1}{4} = \frac{3}{4}$)».

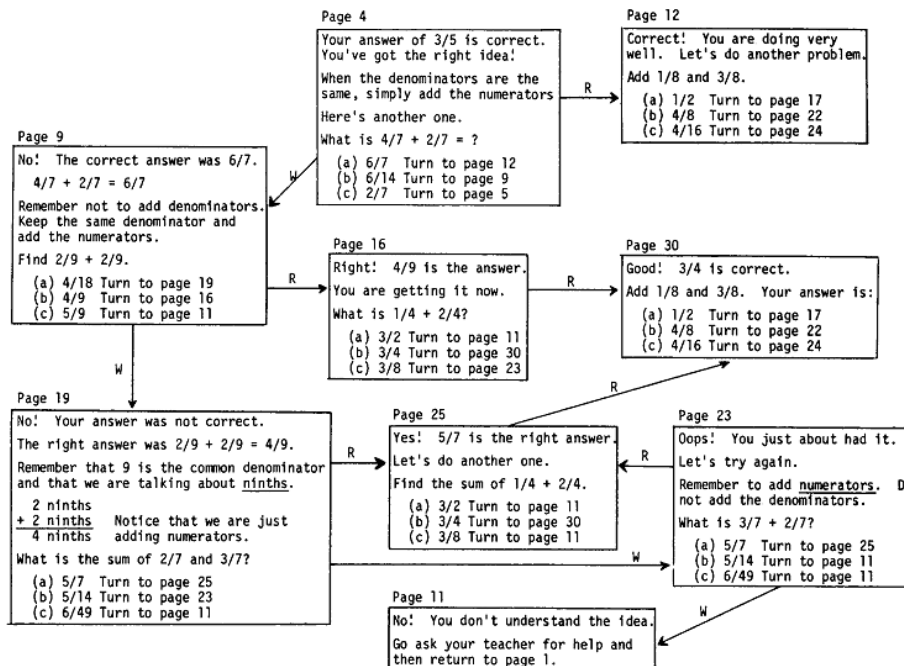


Рис. 3. Гіперзв'язки між фреймами при навчанні додавання дробів

Перевагою запропонованого підходу автори [7] вважають високий ступінь адаптивності також й до апаратного забезпечення, пропонуючи застосовувати мову високого рівня «... для надання адаптивності до будь-якого комп'ютера, що має телекомунікаційні можливості. Це означає, що будь-яка школа має доступ як до свої власних комп'ютерів, так й до регіональних, і навіть до комерційних систем спільного використання, що надаватимуть ці програми як послуги». Таким чином, ще майже 50 років тому було визначено доцільність реалізації адаптивного програмного забезпечення для навчання теми «Дробові числа» як мережного програмного засобу з віддаленим доступом.

3.3 Огляд існуючих адаптивних програмних засобів навчання математики

У мережі Інтернет можна знайти багато різноманітних адаптивних програм, за допомогою яких можна вивчити математику. Всі вони мають свої переваги та недоліки, відрізняються функціональністю, інтерфейсом та змістом навчання.

Розглянемо більш детально найбільш популярні програмні засоби навчання математики з теми «Дробові числа».

«Уздовж однієї лінії» – програма для вивчення чисел і дій над ними в якій учневі потрібно закріпити позначки, що відображають частини дробу (рис. 4).

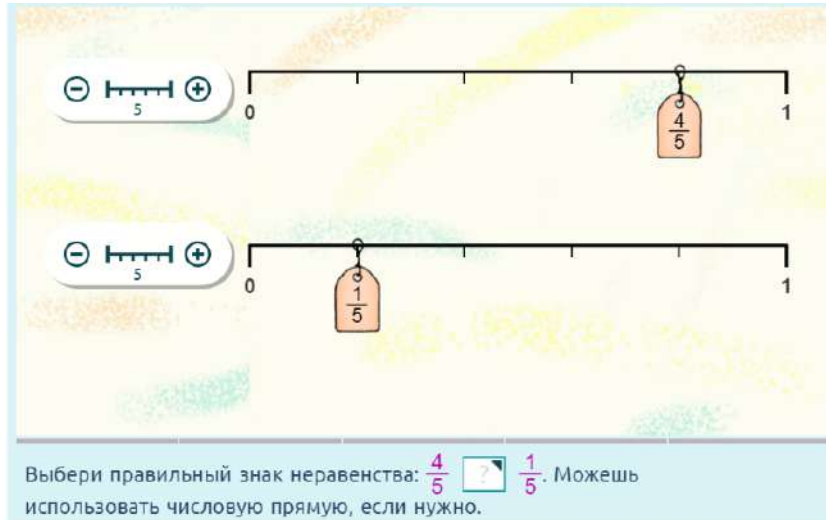


Рис. 4. Головне вікно програми «Уздовж однієї лінії»

«Тягни авто» – щоб пройти рівень і перемогти, учень повинен із дробів вибирати найбільший. Якщо він вибрав дріб правильно, то його авто рушить праворуч, якщо неправильно – ліворуч. Ця гра допомагає дитині швидко прийняти рішення, розвиває вміння порівняння дробових чисел (рис. 5).



Рис. 5. Головне вікно програми «Тягни авто»

Розглянемо ще один додаток – Learning.ua (рис. 6). На головній сторінці можна обрати рівень навчання. Вибравши 5 клас, з'явився короткий опис «Завдання з математики для 5 класу»: *П'ятий клас — дуже відповідальний рік для всіх учнів. Програма цього року значно ускладнюється у порівнянні з четвертим класом. Одна тема швидко змінює іншу, а завдання потребують від учнів підвищеної уваги та кмітливості. Батьки та діти повинні бути підготовлені до цього та відповідально ставитися до кожної теми, яка розглядається на уроці математики.*



Завдання з математики для 5 класу

П'ятий клас — дуже відповідальний рік для всіх учнів. Програма цього року значно ускладнюється у порівнянні з четвертим класом. Одна тема швидко змінює іншу, а завдання потребують від учнів підвищеної уваги та кмітливості. Батьки та діти повинні бути підготовлені до цього та ві

[Допомога онлайн](#)

Рис. 6. Головна сторінка Learning.ua

У рівні повторюються арифметичні дії з натуральними числами, а також їхні властивості, які були вивчені у четвертому класі. Програма ускладнюється появою дій з дробами та відсотками.

Окрім цього, у п'ятому класі вивчається числова пряма, на базі якої закріплюються такі важливі моменти, як лінія, промінь, точка та відрізок. Вивчаються такі поняття, як координатний промінь та бісектриса кута. У курсі математики за п'ятий клас починають у повній мірі розглядати поняття площі та об'єму різних геометричних фігур. Вивчаються формули, за допомогою яких можна визначити периметр, площу та об'єм різних фігур [15].

Learning.ua надає багато додатків до теми «Дробові числа». Натиснувши на будь-яке посилання, користувач переходить до нового вікна, де зображено завдання. З початку завдання починається відлік часу. Чим швидше користувач відповідає на питання, тим більше балів він отримає. Приклади виконання завдань подані на рис. 7, 8 та 9.

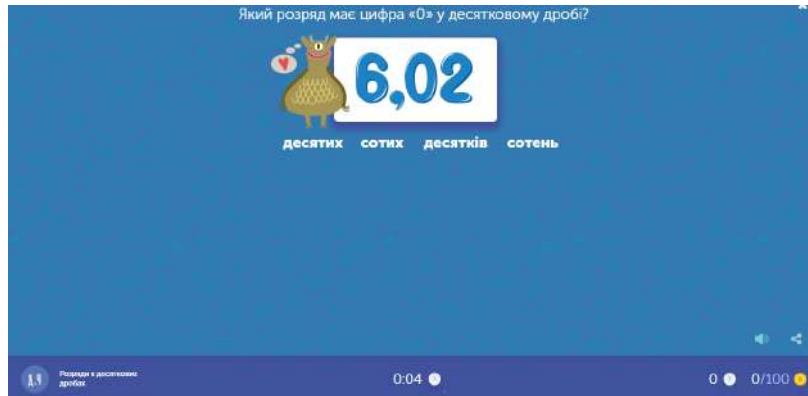


Рис. 7. Приклад виконання завдання

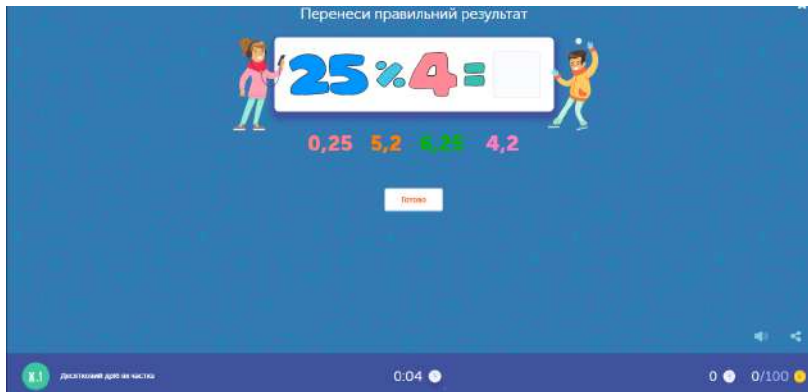


Рис. 8. Десятковий дріб як частка



Рис. 9. Який результат найбільший/найменший?

4 Проектування та розробка адаптивного навчального програмного забезпечення з теми «Дробові числа»

4.1 Обґрунтування вибору інструментів розробки

Для розробки програмного забезпечення були використані наступні засоби: OpenServer, Sublime Text 3, Adobe Photoshop CS6, Bootstrap 3 та середовища розробки мовами HTML, CSS 3, PHP і JavaScript.

Вибір засобів розробки є одним з важливих етапів при створенні програмного засобу, тому розглянемо їх більш детально.

OpenServer – портативна серверна платформа і програмна середовище для веб-розробників, що має багатий набір серверного програмного забезпечення, зручний багатофункціональний інтерфейс, великий спектр можливостей із адміністрування та налаштування компонентів. Платформа широко використовується з метою налагодження, розробки і тестування веб-проектів [12].

Sublime Text 3 – зручний багатоплатформовий текстовий редактор, що підтримує плагіни мовою програмування Python. Sublime Text підтримує велику кількість мов програмування і має можливість підсвічування синтаксису для C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL і XML. Sublime Text може бути оснащений менеджером пакетів, який дозволяє користувачеві знаходити, встановлювати, оновлювати і видаляти пакети без перезавантаження програми. Менеджер підтримує встановлені пакети в актуальному стані, завантажуючи нові версії з репозиторіїв. Крім того, він надає команди для активації і деактивації встановлених пакетів [8].

У розробленому програмному забезпеченні використовується багато малюнків, які потрібні для того, щоб наочно пояснити будь-яку тему з уроків. Для їх створення використовувалась програма **Adobe Photoshop CS6** – продукт для обробки зображень з розширеною функціональністю, що підтримує всі функції класичного Photoshop, а також надає можливості створення і редагування тривимірних зображень. Adobe Photoshop Extended дозволяє малювати на тривимірних об'єктах, створювати тривимірну анімацію, перетворювати двовимірні об'єкти на тривимірні. Система дозволяє додавати ефекти розмиття, нахилу і зсуву, підвищувати різкість однієї точки фокусу, а також застосовувати різні типи розмиття до декількох фокусних точок. Інструмент кадрування підтримує апаратне прискорення і має сучасний інтерфейс, що дозволяє виконувати редагування на полотні і переглядати результати в реальному часі. Крім того, доступне фонове автоматичне збереження змін файлів для відновлення у разі непередбаченого збою [1].

Bootstrap 3 – вільний набір інструментів для створення сайтів та веб-додатків. Включає в себе HTML і CSS-шаблони для оформлення типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсу, включаючи JavaScript-розширення. Bootstrap використовує сучасні напрацювання в області CSS і HTML, тому необхідно бути уважним при підтримці старих браузерів [2].

Програмне забезпечення було створено із використанням наступних мов:

HTML (HyperText Markup Language – мова розмітки гіпертекстових документів) – стандартна мова розмітки веб-сторінок в Інтернет [9].

CSS 3 (Cascading Style Sheets – каскадні таблиці стилів) – спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів. Таблицю стилів CSS можна убудувати в HTML-сторінку (внутрішня таблиця стилів), або ж її можна створити в окремому файлі, і вже потім приєднати посилання на нього до потрібної HTML-сторінки (зовнішня таблиця стилів) [3].

PHP – мова програмування загального призначення з відкритим вихідним кодом. PHP сконструйований спеціально для веб-розробки, а його код може убудовуватись безпосередньо у HTML. Програми мовою PHP виконуються на сервері і генерують HTML, який надсилається клієнту [25].

JavaScript – це «безпечна» мова програмування загального призначення, можливості якої залежать від оточення, в якому запущений JavaScript. У браузері JavaScript вміє робити все, що відноситься до маніпуляції зі сторінкою, взаємодії з відвідувачем і, в якійсь мірі, з сервером:

- створювати нові теги HTML, видаляти, змінювати стилі елементів, приховувати, показувати елементи тощо;
- реагувати на дії користувача, опрацьовувати події від миші, переміщення курсору, натискання на клавіші тощо;
- надсилати запити на сервер і завантажувати дані без перезавантаження сторінки (ця технологія називається AJAX);
- отримувати та встановлювати cookie, запитувати дані, виводити повідомлення [16].

4.2 Розробка архітектури програмного забезпечення

Проектоване клієнт-серверне програмне забезпечення використовує 3 основні компоненти: базу даних, веб-браузер та веб-сервер (рис. 10).

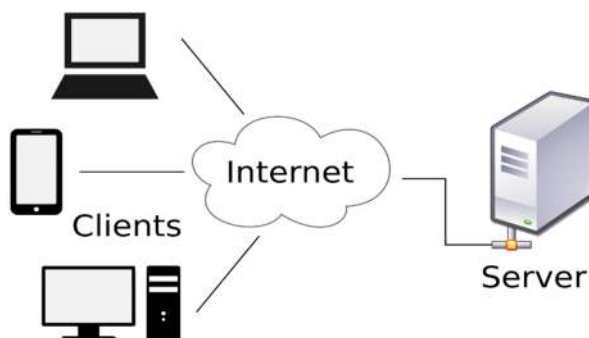


Рис. 10. Модель «клієнт-сервер»

Архітектура клієнт-сервер являє собою розподілену структуру програми, яка розділяє завдання або робочі навантаження між постачальниками ресурсу або сервісу (серверами) та ініціаторами запитів (клієнтами) [10].

У даному програмному забезпеченні браузер знаходиться на першому рівні архітектури. За допомогою браузера здійснюються будь-які операції, які потребує користувач. На другому рівні архітектури знаходиться логіка веб-серверу, тобто те, як сервер буде опрацьовувати прийняті від користувача дані. На третьому рівні знаходиться система управління базами даних, яка буде зберігати дані для опрацювання сервером.



Рис. 11. Діаграма станів процесу авторизації/реєстрації користувача



Рис. 12. Діаграма процесу навчання

Розглянемо складові архітектури системи. Для того, щоб почати навчання, користувач спочатку повинен пройти реєстрацію: йому потрібно ввести свій логін, який він буде використовувати при вході до власного кабінету, та пароль. Якщо користувач введе логін, який вже існує в базі, то спливе повідомлення про те, щоб користувач ввів інший логін. Якщо ніяких помилок немає, то користувач

переходить до свого кабінету, де може почати навчання, обравши певний урок (рис. 11).

Для того, щоб розпочати навчання, користувач має натиснути на кнопку «Розпочати навчання», після чого обрати, який із п'ятнадцяти уроків він хоче вивчити. У процесі навчання користувачеві будуть надані теорія з даної теми та практика. До практики входить 10 тестових завдань, завдяки яким програма визначає рівень складності, після цього, будуть подані завдання, на які користувачеві потрібно дати правильні відповіді (рис. 12).

Визначення рівня знань виконується за кількістю вірних відповідей (рис. 13).

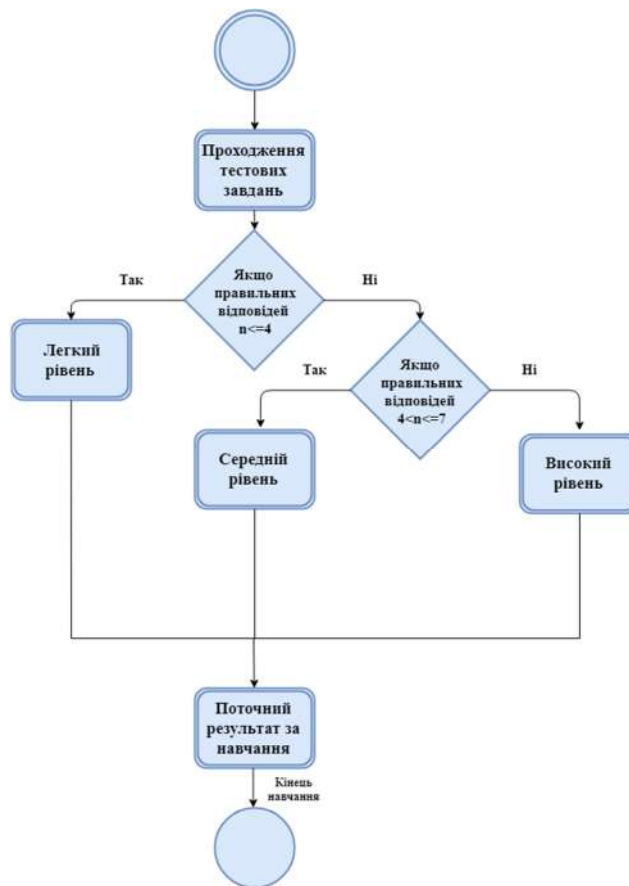


Рис. 13. Діаграма визначення рівня знань

Щоб подивитися назви уроків, користувачеві потрібно буде натиснути на кнопку із меню «Список уроків». Також користувач може подивитися інформацію про програмне забезпечення, натиснувши на кнопку «Про додаток».

4.3 Програмна реалізація проекту

Одним з головних завдань проектування було створення зручної сторінки для відповіді на тестові завдання для того, щоб учень міг легко і зручно дати відповідь на запитання. Виходячи з усього цього, було прийнято рішення, що для розробки цього додатка, оптимальними засобами є каскадні таблиці CSS 3, мова розмітки HTML 5, PHP, а також AJAX, за допомогою якого з метою скорочення часу роботи сторінки перевантажуються не повністю, а лише частково.

Після входу до додатку учневі буде представлено меню з трьох пунктів «Розпочати навчання», «Список уроків» та «Про додаток». Код елемента меню подано нижче:

```
<html lang="UTF-8" >
<body>
<form action="index.php" method="POST">
  <div id='ajax_reload'>
    <div class = "okno">
      <div class = "play" id="example-1">
        <h2>Розпочати навчання</h2>
      </div>
      <div class = "play" id="option">
        <h2>Список уроків</h2>
      </div>
      <div class = "play" id="ogame">
        <h2>Про додаток</h2>
      </div>
    </div>
  </div>

  <div class="container2">
    
    <? echo "Ім'я: ".$_SESSION['login']. " "; ?>
    <input class="dws-submit2" name="submit" type="submit"
value="Вихід">
  </div>
</form>
</body>
  <script type="text/javascript" src="/js/game.js"></script>
</html>
```

Наступним кроком було створення списку уроків, за допомогою яких можна перейти до навчання. Створено AJAX запит, який після натискання на кнопку «Розпочати навчання» з div блоком id=example-1 звертається до файлу example.php. Код елемента AJAX запити:

```
$(document).ready(function() {
```

```
// вішаємо на клік по елементу
$('#example-1').click(function(){
    // завантаження HTML коду з файлу example.html
    // $('#ajax_reload').load('/ajax/example.html');
    $.post('/ajax/example.php', {}, function (answer)
    {
        $('#ajax_reload').html(answer);
    });
})
```

Потім з'явиться результат запити з оновленням div блоку з id=example-1:

```
<div class = "okno">
  <div class = "menu">
    <h1> Уроки </h1>
  </div>
  <div id="example-1">
    <div class ="less1" id="example_2"> Урок 1
    </div>
    <div class='less1' id="example_3"> Урок 2
    </div>
    <div class='less1' id="example_4"> Урок 3<br>
    </div>
    <div class='less1' id="example_5"> Урок 4
    </div>
    <div class='less1' id="example_6"> Урок 5
    </div>
    <div class='less1' id="example_7"> Урок 6
    </div>
    <div class='less1' id="example_8"> Урок 7
    </div>
    <div class='less1' id="example_9"> Урок 8
    </div>
    <div class='less1' id="example_10"> Урок 9
    </div>
    <div class='less1' id="example_11"> Урок 10
    </div>
    <div class='less1' id="example_12"> Урок 11
    </div>
    <div class='less1' id="example_13"> Урок 12
    </div>
    <div class='less1' id="example_14"> Урок 13
    </div>
    <div class='less1' id="example_15"> Урок 14
    </div>
    <div class='less1' id="example_16"> Урок 15
```



```

        </div>
    </div>
</div>
<div id='men' class="dws-submit3">Меню</div>

```

У кожного уроку створений свій div блок з певним id. Розглянемо AJAX запит переходу до сторінки навчання уроку:

```

$(document).on('click','#example_2', function(){
    $.post('/ajax/1/example1.php', {}, function (answer){
        $('#ajax_reload').html(answer);
    });
});

```

При натисканні на кнопку «Урок 1», що має div блок з id="example_2", завантажиться HTML-код з файлу example1.php. Розглянемо детальніше зміст коду з файлу example1.php:

```

<script>
$('.fraction').each(function(key, value) {
    $this = $(this)
    var split = $this.html().split("/")
    if( split.length == 2 ){
        $this.html('<span class="top">'+split[0]+ '</span><span
class="bottom">'+split[1]+'</span>')
    }
});
</script>

```

Використавши JavaScript, було створено функцію 'fraction'. Так як дроби записують за допомогою двох натуральних чисел і риски дроби, треба записати число над рисою дроби, що є чисельником, саму риску дроби та число під рисою дроби, що є знаменником:

```
<span class="fraction">1/8</span>
```

Використовуючи тег , який призначений для визначення рядкових елементів документа, додаємо атрибут class з ім'ям селектора class="fraction". Першим число записується чисельник дроби, риска дроби та знаменник. У результаті вийде дробове число.

Код елемента з «Урок 1»:

```

<p> <span class="colortext">Приклад 1. </span>
    В саду Барвінку зростає 24 дерева, з них 7 дерев –
яблуні. Яку частину всіх дерев становлять яблуні?<br>
    Оскільки все зростає 24 дерева, то одна яблуня становить
    <span class="fraction">1/24</span>
    всіх дерев, а 7 яблунь –

```

```

<span class="fraction">7/24</span>
всіх дерев.<br><br>
<span class="blacktext">Відповідь:
<span class="fraction">7/24</span></span>.<br>
<span class="colortext">Приклад 2. </span>
В саду Барвінку зростає 24 дерева, з них
<span class="fraction">5/8</span>
складають вишні. Скільки вишневих дерев росте в саду?<br>
1)  $24 : 8 = 3$  (дерева) - становить
<span class="fraction">1/8</span>
всіх дерев.<br>
2)  $3 * 5 = 15$  (дерев) - зростає вишень.<br>
<span class="blacktext">Відповідь: 15 вишень.</span></p>

```

Крім теоретичної частини, в процесі навчання є і практичні завдання. Ці завдання подані у вигляді тестів і завдань, в яких потрібно ввести відповідь з клавіатури.

Нижче наведено лістинг коду тестового завдання:

```

<span class="zadanue"> В саду росло 56 дерев, з них 23 - яблуні.
Яку частину дерев становили яблуні??<br><br> </span>
<form id="forma">
<div class="row">
<p><div class="col-6">
<input class='r1' type="radio" id="test1" name="q1" value="3">
<label for="test1">
<span class="fraction">56/23</span> (дерев)
</label></div>
<div class="col-6">
<input class='r2' type="radio" id="test2" name="q1" value="2">
<label for="test2">
<span class="fraction">23/23</span> (дерев)
</label></div>
</p><br>
<p> <div class="col-6">
<input class='r1' type="radio" id="test3" name="q1" value="1">
<label for="test3">
<span class="fraction">23/56</span> (дерев)
</label></div>
<div class="col-6">
<input class='r2' type="radio" id="test4" name="q1" value="4">
<label for="test4">
23 (дерев)
</label></div></p>
<input class="test" type="button" onclick="send();"
value="Відповісти">
<div id="result"></div>

```

У тезі `` записується завдання, на яке треба дати вірну відповідь. Далі, використовуючи `type="radio"`, задаються варіанти відповіді. При натисканні кнопки форми "Відповісти" спрацьовує подія `onclick` і йде виклик функції з ім'ям `send()`, яка перевіряє, чи правильний обрали варіант відповіді. Код функції `send()` для відправки AJAX-запиту:

```
function send() {
    $.ajax({
        type: "POST",
        url: '/ajax/1/test1.php',
        data: $("#forma").serialize(),
        success: function(html) {
            $("#result").empty();
            $("#result").append(html);
        }
    });
}
```

У даній функції, відправляється POST-запит до файлу обробника `test1.php` і в результаті отримуємо відповідь. Фрагмент лістингу коду наведено нижче:

```
<?php
include $_SERVER['DOCUMENT_ROOT'].'/db/db.php';
if(!empty($_POST)){
$q1 = $_POST['q1'];
$q2 = $_POST['q2'];
$q3 = $_POST['q3'];
$q4 = $_POST['q4'];
$q5 = $_POST['q5'];
$q6 = $_POST['q6'];
$q7 = $_POST['q7'];
$q8 = $_POST['q8'];
$q9 = $_POST['q9'];
$q10 = $_POST['q10'];
if($q1 == 1 or $q2 == 1 or $q3 == 1 or $q4 == 1 or $q5 == 1 or
$q6 == 1 or $q7 == 1 or $q8 == 1 or $q9 == 1 or $q10 == 1) {
echo "<div class = 'result'>Вірна відповідь!<br> Рухайся
далі<br> +1 бал";
$result = $dbh -> query ("UPDATE test1 SET t1 = t1 + 1");
}
else
echo "<div class = 'result'>На жаль, відповідь не є
вірною!<br>Спробуй далі!<br>";
}
else echo "<div class = 'result'><p>Ти не відповів на
питання!</p>";
?>
```

За допомогою тестових завдань визначається рівень знань із даної теми. Рівень визначається за кількістю правильних відповідей. Обрано три рівні: низький, середній і високий. За цими рівнями формуються такі завдання, в яких потрібно ввести відповіді з клавіатури.

Елемент коду завдань:

```
<?php
include $_SERVER['DOCUMENT_ROOT'].'/db/db.php';
foreach ($dbh->query( "SELECT t1 FROM test1") as $data)
if($data['t1'] <= 4) {
echo '<span class="zadanue">Три рибалки зловили 168 риб. Щукін
зловив <span class="fraction">5/14</span> усіх риб, Окунев -
<span class="fraction">8/21</span> усіх риб, а Карасьов - решту.
Скільки риб зловив Карасьов?<br><br></span>';
}
if($data['t1'] > 4 and $data['t1'] <= 7) {
echo "<span class='zadanue'>Ілля Муромець заготував на зиму для
свого коня 3 4 т 9 ц вівса. За грудень кінь з'їв <span
class='fraction'>3/7</span> усього запасу вівса, а за січень -
<span class='fraction'>9/14</span> решти. Скільки центнерів
вівса кінь з'їв за січень?<br><br></span>";
}
if($data['t1'] > 7 and $data['t1'] <= 10) {
echo '<span class="zadanue">Знайдіть число, <span
class="fraction">2/3</span> якого дорівнюють <span
class="fraction">3/7</span> числа 210.<br><br></span>';
}
}
?>
```

Відповідь, яку вводить учень з клавіатури при натисканні на кнопку «Перевірити», звіряється із даними, які знаходяться в базі даних.

Код елемента із перевірки правильної відповіді:

```
if($data['t1'] <= 4) {
    if($data['chusl'] == ($_POST['chusl']) && $data['znam'] ==
($_POST['znam'])) {
        $result = $dbh -> query ("UPDATE test1 SET t2 = t2 + 1");
        echo "<div class = 'result'>Вірна відповідь!<br> Рухайся
далі<br> +1 бал";
    } else {
        echo "<div class = 'result'>Помиляєшся!<br>Вірна
відповідь:<br><span class='fraction'>", $data['otv'], "</span>";
    }
}
```

Наприкінці кожного уроку виводиться кінцевий результат та кількість правильних відповідей. Після цього потрібно повернутися в меню і розпочати наступний урок.

5 Тестування програмного забезпечення

Для того, щоб розпочати роботу із розробленим програмним засобом, необхідно ініціалізувати клієнтську (завантажити будь-який веб-браузер) та серверну (запустити програмне середовище Open Server) частини.

При натисканні на іконці Open Server відкривається меню, у якому після вибору пункту «Запустити» активуються всі інші пункти меню. Після цього необхідно перейти до пункту «Мої сайти» і обрати Math.loc.

Після вибору користувач попадає на сайт, де буде проходити навчання математики з теми «Дробові числа». Для початку навчання користувачу потрібно зареєструватися, після чого увійти у власний кабінет (рис. 14).

Після входу з'явиться головне меню та вікно, де вказане ім'я користувача, який навчатиметься. У головному меню користувач може обрати один із трьох заданих пунктів: «Розпочати навчання», «Список уроків», «Про додаток». За потреби користувач може залишити сторінку, натиснувши кнопку «Вихід» (рис. 15).

Обравши пункт із меню «Розпочати навчання», користувач отримає посилання на уроки, за якими він зможе перейти і почати навчання. Тема «Дробові числа» розділена на два розділи. У першому розділі, який складається з 5 уроків, розглядаються звичайні дроби. У другому розділі, який складається з 10 уроків, розглядаються десяткові дроби. Отже, користувачеві буде надано 15 уроків (рис. 16).

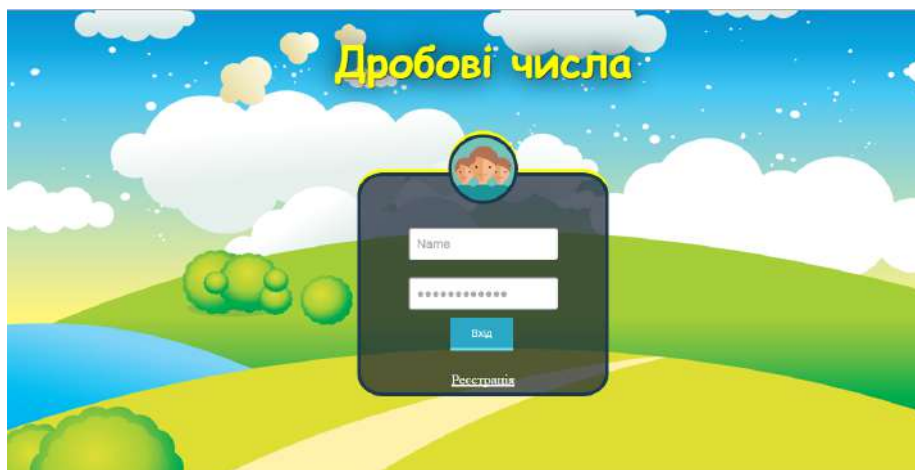


Рис. 14. Головна сторінка програмного засобу «Дробові числа»

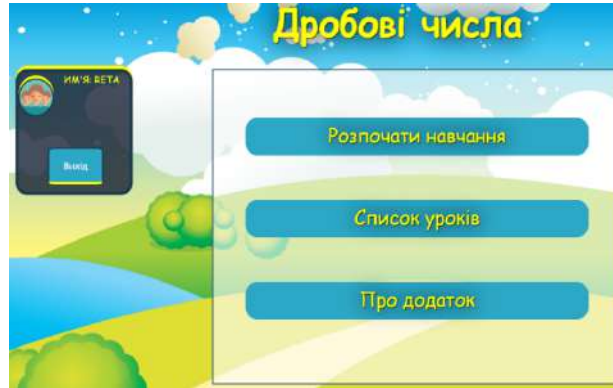


Рис. 15. Меню додатку

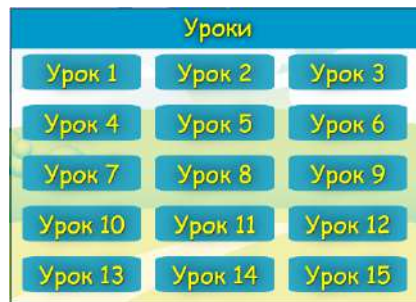


Рис. 16. Список уроків

Натиснувши на кнопку «Урок 1», учень перейде до початку уроку (рис. 17).



Рис. 17. Перша сторінка уроку 1 «Уявлення про звичайні дроби»

Прочитавши теорію на першій сторінці, учень натискає кнопку «Далі», щоб перейти до наступної сторінки. Наприкінці теорії будуть наведені деякі приклади задач із розв'язками, щоб учень краще розумів дану тему. Після завершення опрацювання теорії та прикладів розпочинається практична робота, в якій потрібно попрацювати над тестовими завданнями (рис. 18).

Завдання 1

В саду росло 56 дерев, з них 23 - яблуні. Яку частину дерев становили яблуні??

$\frac{56}{23}$ (дерев) $\frac{23}{23}$ (дерев)

$\frac{23}{56}$ (дерев) 23 (дерев)

Відповісти

Рис. 18. Тестове завдання уроку 1

У тестовому завданні представлено одне питання та чотири відповіді, з яких лише одна відповідь є правильною. При натисканні на неправильну відповідь з'являється повідомлення про те, що відповідь є невірною (рис. 19).

Завдання 1

В саду росло 56 дерев, з них 23 - яблуні. Яку частину дерев становили яблуні??

$\frac{56}{23}$ (дерев) $\frac{23}{23}$ (дерев)

$\frac{23}{56}$ (дерев) 23 (дерев)

На жаль, відповідь не є вірною!
Спробуй далі!

Відповісти

Рис. 19. Повідомлення про невірну відповідь

Якщо відповідь є правильною, то з'являється повідомлення, що учневі нараховується один бал (рис. 20).

Після того, як учень пройде десять тестових завдань, йому виведеться кількість вірних відповідей, щоб він зрозумів свій рівень засвоєння даної теми. Потім почнуться завдання, в яких потрібно дати одну правильну відповідь на питання (рис. 21).



Рис. 20. Повідомлення про вірну відповідь



Рис. 21. Практичне завдання

6 Висновки

У процесі дослідження проблеми розробки веб-орієнтованого програмного забезпечення для підтримки навчання математики у закладах загальної середньої освіти були розв'язані завдання аналізу стану проблеми розробки та використання адаптивного програмного забезпечення для навчання математики, обґрунтування вибору засобів розробки адаптивного навчального програмного забезпечення для учнів середньої школи, розробки та експериментальної перевірки адаптивного навчального програмного забезпечення з теми «Дробові числа» для учнів 5 класу закладів загальної середньої освіти.

Розв'язання поставлених задач надало можливість зробити наступні висновки:

1. Теоретичною основою побудови адаптивних навчальних програм є програмоване навчання – тип навчання, що здійснюється у відповідності до навчальної програми, яка контролює як обсяг знань, умінь і навичок, якими повинні оволодіти учні, так і спосіб організації їх навчальної діяльності. Це

досягається шляхом поділу навчального матеріалу теми «Дробові числа» на окремі порції та інтенсивного обміну відомостями між учнем та навчальною програмою, що здійснюється переважно у формі «питання-відповідь».

2. Вивчення зарубіжного досвіду програмованого навчання теми «Дробові числа» із використанням засобів ІКТ надає можливість стверджувати, що доцільним є розробка програмного забезпечення, що надається учневі як послуга через телекомунікаційні мережі. На сучасному етапі розвитку ІКТ таким є клієнт-серверне програмне забезпечення із веб-доступом, тому провідними засобами розробки були визначені OpenServer, Sublime Text 3, Adobe Photoshop CS6, Bootstrap 3 та середовища розробки мовами HTML, CSS 3, PHP і JavaScript.
3. Розроблене навчальне програмне забезпечення з теми «Дробові числа» для учнів 5 класу закладів загальної середньої освіти містить компоненти входу (для індивідуалізації процесу навчання), попереднього тестування (для адаптації навчального матеріалу до рівня знань користувача) та багаторівневого навчання (15 уроків, що відповідають повному змісту теми «Дробові числа»).

Результати експериментального навчання з використанням розробленого програмного забезпечення показали, що виконана розробка не урахує зміну індивідуальних особливостей учня (у тому числі психологічних) у процесі навчання. Це визначає наступні шляхи його розвитку, спрямованого на підвищення рівня адаптивності:

- індивідуалізація навчальних впливів;
- формалізація та реалізація моделі учня;
- урахування процесуальних характеристик навчання у моделі учня.

Список використаних джерел

1. Adobe Photoshop CS6 Extended (License) [Електронний ресурс] / Арсенал. – 2018. – Режим доступу : http://www.arssoft.ru/mycatalog/view_post.php?id=144.
2. Bootstrap · The most popular HTML, CSS, and JS library in the world. [Електронний ресурс] / Mark Otto, Jacob Thornton, and Bootstrap contributors. – Режим доступу : <http://getbootstrap.com/>.
3. HTML & CSS [Електронний ресурс] / World Wide Web Consortium (W3C). – 2016. – Режим доступу : <https://www.w3.org/standards/webdesign/htmlcss>.
4. Modlo Ye. O. Xcos on Web as a promising learning tool for Bachelor's of Electromechanics modeling of technical objects [Electronic resource] / Yevhenii O. Modlo, Serhiy O. Semerikov // Cloud Technologies in Education : Proceedings of the 5th Workshop on Cloud Technologies in Education (CTE 2017). Kryvyi Rih, Ukraine, April 28, 2017 / Edited by : Serhiy O. Semerikov, Mariya P. Shyshkina. – P. 34-41. – (CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2168). – Access mode : <http://ceur-ws.org/Vol-2168/paper6.pdf>.
5. Rahmlow H. F. A Series of Programmed Instruction Books for Learning Occupationally Oriented Basic Mathematics. Report No. 16. Final Report / Harold F. Rahmlow ; Washington State Coordinating Council for Occupational Education, Olympia ; Washington

- State Univ., Pullman. Coll. of Education. – Washington : U.S. Department of Health, Education and Welfare, 1968. – 20 p.
6. Rahmlow H. F. Occupational Mathematics; Ratios and Fractions. Report No. 16-D. Final Report / Harold F. Rahmlow, Karl Ostheller, Clarence Potratz, Leonard T. Winchell, Arthur Snoey ; Washington State Coordinating Council for Occupational Education, Olympia ; Washington State Univ., Pullman. Coll. of Education. – Washington : U.S. Department of Health, Education and Welfare, 1968. – 68 p.
 7. Rahmlow H. F. The modular development of computer-assisted instruction programs / Harold F. Rahmlow, Eugene G. Kerr // Performance Improvement. – 1969. – Vol. 8. – Iss. 2. – P. 20-23. – <https://doi.org/10.1002/pfi.4180080204>.
 8. Sublime Text - A sophisticated text editor for code, markup and prose [Електронний ресурс]. – Режим доступу : <https://www.sublimetext.com/>.
 9. W3C Invites Implementations of WebVTT: The Web Video Text Tracks Format [Електронний ресурс]. – May 11, 2018. – Режим доступу : <https://www.w3.org/html/>
 10. Архітектура клієнт-сервер [Електронний ресурс] / Павлоградський коледж Державного ВНЗ "НГУ". – 2017. – Режим доступу : <http://inter.ptngu.com/kompyuterni-merezhi/arhitektura-kliiyent-server>.
 11. Балл Г. А. Программированное обучение / Г. А. Балл // Компьютерная технология обучения : словарь-справочник / под ред. Гриценко В. И., Довгялло А. М., Савельева А. Я. – К. : Наукова думка, 1992. – [Кн. 1] : А-М. – 1992. – С. 420-423.
 12. Встречайте: Open Server! [Електронний ресурс] / ospanel.io. – 2018. – Режим доступу : <https://ospanel.io>.
 13. Гриневич Л. М. Нова українська школа: концептуальні засади реформування середньої школи [Електронний ресурс] / Гриневич Лілія, Елькін Олександр, Калашнікова Світлана, Коберник Іванна, Ковтунець Володимир, Макаренко Оксана, Малахова Олена, Нанаєва Тетяна, Усатенко Галина, Хобзей Павло, Шиян Роман ; загальна редакція : Гриценко Михайло. – 2016. – 34 с. – Режим доступу : <https://mon.gov.ua/storage/app/media/zagalna%20serednya/nova-ukrainska-shkola-compressed.pdf>.
 14. Довгялло А. М. Адаптации задача / А. М. Довгялло // Компьютерная технология обучения : словарь-справочник / под ред. Гриценко В. И., Довгялло А. М., Савельева А. Я. – К. : Наукова думка, 1992. – [Кн. 1] : А-М. – 1992. – С. 68-69.
 15. Завдання з математики для 5 класу [Електронний ресурс] / КідУнівер. – 2018. – Режим доступу : <https://learning.ua/matematyka/piatyi-klas>.
 16. Кантор И. Введение в JavaScript [Електронний ресурс] / Илья Кантор. – 2018. – Режим доступу : <https://learn.javascript.ru/intro>.
 17. Кунат Э. Н. Адаптивная обучающая программа / Э. Н. Кунат, Е. М. Сеница // Компьютерная технология обучения : словарь-справочник / под ред. Гриценко В. И., Довгялло А. М., Савельева А. Я. – К. : Наукова думка, 1992. – [Кн. 1] : А-М. – 1992. – С. 70.
 18. Лобанов Ю. И. Адаптивность системы обучения / Ю. И. Лобанов // Компьютерная технология обучения : словарь-справочник / под ред. Гриценко В. И., Довгялло А. М., Савельева А. Я. – К. : Наукова думка, 1992. – [Кн. 1] : А-М. – 1992. – С. 71.
 19. Математика. 5–9 класи : навчальна програма для загальноосвітніх навчальних закладів [затверджена Наказом Міністерства освіти і науки України від 07.06.2017 № 804] [Електронний ресурс] / Міністерство освіти і науки України. – [2017]. – 40 с. – Режим доступу : <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-5-9-klas/onovlennya-12-2017/5-programa-z-matematiki.docx>.

20. нова українська школа | Міністерство освіти і науки України [Електронний ресурс]. – Режим доступу : <https://mon.gov.ua/ua/tag/nova-ukrainska-shkola>.
21. Нова українська школа : poradnyk dlya vchytelya [Електронний ресурс] / Під заг. ред. Бібік Н. М. – К. : Видавничий дім «Плеяди», 2017. – 206 с. – Режим доступу : <http://nus.org.ua/wp-content/uploads/2017/11/NUSH-poradnyk-dlya-vchytelya.pdf>.
22. Петюшкин А. В. HTML в Web-дизайне / Алексей Петюшкин. – СПб. : БВХ-Петербург, 2004. – 400 с.
23. Портал знань – Знання повинні бути доступними | Портал знань, портал знаний, дистанційне навчання [Електронний ресурс] / Портал Знань. – 2018. – Режим доступу : <http://www.znannya.org/>.
24. Рекомендація Європейського Парламенту та Ради (ЄС) "Про основні компетенції для навчання протягом усього життя" [Електронний ресурс] : Рекомендації, Міжнародний документ № 2006/962/ЄС / Європейський Союз. – 18.12.2006. – Режим доступу : http://zakon0.rada.gov.ua/laws/show/994_975.
25. What is PHP? [Electronic resource] / The PHP Group. – 2018. – Access mode : <http://php.net/manual/en/intro-what-is.php>.

References (translated and transliterated)

1. Adobe Photoshop CS6 Extended (License). http://www.arssoft.ru/mycatalog/view_post.php?id=144 (2017). Accessed 12 Jun 2017
2. Bootstrap · The most popular HTML, CSS, and JS library in the world. <http://getbootstrap.com> (2017). Accessed 12 Jun 2017
3. HTML & CSS. World Wide Web Consortium (W3C). <https://www.w3.org/standards/webdesign/htmlcss> (2016). Accessed 23 Jun 2017
4. Modlo, Ye.O., Semerikov, S.O.: Xcos on Web as a promising learning tool for Bachelor's of Electromechanics modeling of technical objects. In: Semerikov, S.O., Shyshkina, M.P. (eds.) Proceedings of the 5th Workshop on Cloud Technologies in Education (CTE 2017), Kryvyi Rih, Ukraine, April 28, 2017. CEUR Workshop Proceedings, vol. 2168, pp. 34–41. <http://ceur-ws.org/Vol-2168/paper6.pdf> (2018). Accessed 15 Sep 2018
5. Rahmlow, H.F.: A Series of Programmed Instruction Books for Learning Occupationally Oriented Basic Mathematics. Report No. 16. Final Report. U.S. Department of Health, Education and Welfare, Washington (1968)
6. Rahmlow, H.F.: Occupational Mathematics; Ratios and Fractions. Report No. 16-D. Final Report. U.S. Department of Health, Education and Welfare, Washington (1968)
7. Rahmlow, H.F., Kerr, E.G.: The modular development of computer-assisted instruction programs. Performance Improvement. **8**(2), 20–23 (1969). doi:10.1002/pfi.4180080204
8. Sublime Text - A sophisticated text editor for code, markup and prose. <https://www.sublimetext.com> (2018). Accessed 25 Oct 2018
9. W3C Invites Implementations of WebVTT: The Web Video Text Tracks Format. <https://www.w3.org/html> (2018). Accessed 25 Oct 2018
10. Архитектура клиент-сервер (Client-server architecture). <http://inter.ptngu.com/kompyuterni-merezhi/arhitektura-kliient-server> (2017). Accessed 21 Mar 2018
11. Ball, G.A.: Программированное обучение (Programmed learning). In: Gritcenko, V.I., Dovgiallo, A.M., Savelev, A.Ia. (eds.) Kompiuternaia tekhnologiia obucheniia: slovar-spravochnik, pp. 420–423. Naukova dumka, Kyiv (1992)
12. Vstrechaite: Open Server! (Meet the: Open Server!). <https://ospanel.io> (2018). Accessed 21 Mar 2018

13. Hrynevych, L.M., Elkin, O., Kalashnikova, S., Kobernyk, I., Kovtunets, V., Makarenko, O., Malakhova, O., Nanaieva, T., Usatenko, H., Khobzei, P., Shyian, R.: Nova ukrainska shkola: kontseptualni zasady reformuvannia serednoi shkoly. <https://mon.gov.ua/storage/app/media/zagalna%20serednya/nova-ukrainska-shkola-compressed.pdf> (2016). Accessed 31 May 2018
14. Dovgiallo, A.M.: Adaptatsii zadacha (Adaptation problem). In: Gritcenko, V.I., Dovgiallo, A.M., Savelev, A.Ia. (eds.) *Kompiuternaia tekhnologiia obuchenii: slovar-spravochnik*, pp. 68–69. Naukova dumka, Kyiv (1992)
15. *Zavdannia z matematyky dlia 5 klasu* (Problems in mathematics for grade 5). <https://learning.ua/matematyka/piaty-klas> (2018). Accessed 30 Apr 2018
16. Kantor, I.: *Vvedenie v JavaScript* (Introduction to JavaScript). <https://learn.javascript.ru/intro> (2018). Accessed 10 May 2018
17. Kunat, E.N., Sinitca, E.M.: *Adaptivnaia obuchaiushchaia programma* (Adaptive training program). In: Gritcenko, V.I., Dovgiallo, A.M., Savelev, A.Ia. (eds.) *Kompiuternaia tekhnologiia obuchenii: slovar-spravochnik*, p. 70. Naukova dumka, Kyiv (1992)
18. Lobanov, Iu.I.: *Adaptivnost sistemy obuchenii* (Learning system adaptivity). In: Gritcenko, V.I., Dovgiallo, A.M., Savelev, A.Ia. (eds.) *Kompiuternaia tekhnologiia obuchenii: slovar-spravochnik*, p. 71. Naukova dumka, Kyiv (1992)
19. *Matematyka. 5–9 klasy: navchalna prohrama dlia zahalnoosvitnikh navchalnykh zakladiv* (Maths. Grades 5–9: A curriculum for general education institutions). <https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-5-9-klas/onovlennya-12-2017/5-programa-z-matematiki.docx> (2017). Accessed 31 Dec 2017
20. nova ukrainska shkola | Ministerstvo osvity i nauky Ukrainy (new Ukrainian school | Ministry of Education and Science of Ukraine). <https://mon.gov.ua/ua/tag/nova-ukrainska-shkola> (2018). Accessed 25 Oct 2018
21. Bibik, N.M. (ed.): *Nova ukrainska shkola: poradnyk dlia vchytelia* (New Ukrainian school: teacher's guide). Vydavnychiy dim «Pleiady», Kyiv (2017)
22. Petiushkin, A.V.: *HTML v Web-dizaine* (HTML in Web-design). BVKh-Peterburg, Sankt-Peterburg (2004)
23. Portal znan – Znannia povynni buty dostupnymi | Portal znan, portal znanyi, dystantsiine navchannia (Knowledge Portal - Knowledge should be available | Portal of knowledge, well-known portal, distance learning). <http://www.znannya.org> (2018). Accessed 20 Nov 2018
24. Rekomendatsiia Yevropeiskoho Parlamentu ta Rady (IeS) Pro osnovni kompetentsii dlia navchannia protiahom usoho zhyttia (Recommendation of the European Parliament and Council (EC) On core competences for lifelong learning). http://zakon0.rada.gov.ua/laws/show/994_975 (2006). Accessed 21 Mar 2017
25. What is PHP? The PHP Group. <http://php.net/manual/en/intro-what-is.php> (2018). Accessed 1 Apr 2018

Проектування програмних засобів доповненої реальності навчального призначення

Олександр Володимирович Сироватський¹,
Сергій Олексійович Семеріков¹[0000-0003-0789-0272],
Євгеній Олександрович Модло²[0000-0003-2037-1557],
Юлія Володимирівна Єчкало³[0000-0002-0164-8365],
Сніжана Олександрівна Зелінська¹[0000-0002-3071-5192]

¹ Криворізький державний педагогічний університет,
пр. Гагаріна, 54, м. Кривий Ріг, 50086, Україна

² Криворізький металургійний інститут Національної металургійної академії України,
вул. Степана Тільги, 5, м. Кривий Ріг, 50006, Україна

³ Криворізький національний університет,
вул. Віталія Матусевича, 11, м. Кривий Ріг, 50027, Україна
{sovhero, semerikov, eugenemodlo, uliaechk}@gmail.com,
zvit-zss@ukr.net

Анотація. У процесі дослідження проблеми професійної підготовки майбутніх учителів інформатики до використання технологій доповненої реальності в освіті розв'язані завдання: 1) виконано історико-технологічний аналіз досвіду застосування засобів доповненої реальності для розробки інтерактивних навчальних матеріалів; 2) схарактеризовано програмне забезпечення для проектування засобів доповненої реальності навчального призначення та визначено технологічні вимоги для факультативу «Розробка програмних засобів віртуальної та доповненої реальності»; 3) розроблено окремі складові навчально-методичного комплексу із проектування систем віртуальної та доповненої реальності для майбутніх учителів інформатики.

Ключові слова: доповнена реальність, віртуальна реальність, ІКТ, інтерактивні навчальні матеріали.

Augmented reality software design for educational purposes

Oleksandr V. Syrovatskyi¹, Serhiy O. Semerikov²[0000-0003-0789-0272],
Yevhenii O. Modlo²[0000-0003-2037-1557], Yuliia V. Yechkalo³[0000-0002-0164-8365]
and Snizhana O. Zelinska¹[0000-0002-3071-5192]

¹ Kryvyi Rih State Pedagogical University, 54, Gagarin Ave., Kryvyi Rih, 50086, Ukraine

² Kryvyi Rih Metallurgical Institute of the National Metallurgical Academy of Ukraine,
5, Stephana Tilhy St., Kryvyi Rih, 50006, Ukraine

³ Kryvyi Rih National University, 11, Vitalii Matushevych St., Kryvyi Rih, 50027, Ukraine
{sovhero, semerikov, eugenemodlo, uliaechk}@gmail.com,
zvit-zss@ukr.net

Abstract. In the process of researching the problem of training future informatics teachers to use augmented reality technologies in education, the tasks were solved: 1) a historical and technological analysis of the experience of using augmented reality tools for developing interactive teaching materials was performed; 2) the software for the design of augmented reality tools for educational purposes is characterized and the technological requirements for the optional course “Development of virtual and augmented reality software” are defined; 3) separate components of an educational and methodical complex for designing virtual and augmented reality systems for future informatics teachers have been developed.

Keywords: augmented reality, virtual reality, ICT, interactive learning materials.

1 Вступ

Доповнена реальність (augmented reality, також відома як mixed reality) – сьогодні досить популярна технологія, що має широкі бізнесові, розважальні та освітні застосування. І. С. Мінтій та В. М. Соловійов у [30] наголошують, що в Україні технологія доповненої реальності практично відсутня в освітньому просторі підготовки майбутніх учителів, незважаючи на виявлений її значний потенціал:

- у професійній підготовці майбутніх інженерів (Н. В. Рашевська, В. М. Соловійов [36]), зокрема – інженерів гірничого профілю (С. О. Зелінська та ін. [53]);
- у позашкільній навчально-дослідницькій діяльності учнів (В. Б. Шаповалов, С. Б. Шаповалов, А. І. Атамась, Ж. І. Білик та ін. [42; 43]);

- у спільному розвитку компетентностей з фізики та іноземної мови на бінарних уроках у ЗЗСО (В. Л. Бузько, А. В. Бонк, О. В. Мерзликін, І. Ю. Тополова та ін. [6; 28]);
- для підготовки до лабораторних занять з природничо-математичних дисциплін у середній та вищій школі (Т. В. Грунтова, Ю. В. Єчкало, А. М. Стрюк, П. П. Нечипуренко, Т. В. Старова, Т. В. Селіванова, А. О. Томіліна та ін. [23; 34]);
- для адаптації студентів-іноземців до навчання на підготовчих відділеннях вітчизняних ЗВО (Н. О. Зінонос, О. В. Віхрова та ін. [54]);
- для навчання глобального читання дошкільників з розладами спектру аутизму (Т. Г. Коломоєць та ін. [24]) тощо.

Технологія доповненої реальності стає важливим компонентом мобільно орієнтованого середовища навчання ЗВО через доступність та повсюдність мобільних Інтернет-пристроїв, використовуваних учасниками навчального процесу [31; 32]. Один з основоположників доповненої реальності А. Е. Сазерленд розглядав її як комбінування модельного та реального світу на певному ідеальному (остаточному) дисплеї (ultimate display), з'єданого із комп'ютером для полегшення знайомства із поняттями, які неможливо реалізувати у фізичному світі [47, с. 507-508]. Тому розробнику засобів доповненої реальності необхідно бути компетентним у комп'ютерній графіці, комп'ютерному моделюванні та застосовувати відповідні поставленим задачам засоби розробки.

2 Апарат дослідження

Проблема дослідження породжена відсутністю у вітчизняному освітньому просторі адаптованих навчальних матеріалів із навчання майбутніх учителів застосування систем доповненої реальності для розробки інтерактивних навчальних матеріалів, що викликало необхідність звернення до зарубіжного досвіду – масових відкритих онлайн-курсів з розробки засобів віртуальної та доповненої реальності.

Мета дослідження – розробити адаптований навчально-методичний комплекс із проектування систем віртуальної та доповненої реальності для майбутніх учителів інформатики.

Для досягнення мети дослідження були поставлені такі *завдання*:

1. Виконати історико-технологічний аналіз досвіду застосування засобів доповненої реальності для розробки інтерактивних навчальних матеріалів.
2. Схарактеризувати програмне забезпечення для проектування засобів доповненої реальності навчального призначення та визначити технологічні вимоги для факультативу «Розробка програмних засобів віртуальної та доповненої реальності».

3. Розробити окремі складові навчально-методичного комплексу із проектування систем віртуальної та доповненої реальності для майбутніх учителів інформатики.

Об'єкт дослідження – професійна підготовка майбутніх учителів інформатики до використання інноваційних технологій в освіті.

Предмет дослідження – адаптований навчально-методичний комплекс із проектування систем віртуальної та доповненої реальності для майбутніх учителів інформатики.

Методи дослідження:

- *аналіз* джерел та програмного забезпечення з метою визначення стану розв'язання проблеми дослідження та добору засобів розробки систем віртуальної та доповненої реальності;
- *синтез* технологічних вимог до факультативу «Розробка програмних засобів віртуальної та доповненої реальності»;
- *методи програмної інженерії* (проектування, розробка, тестування);
- *методи педагогічного проектування* для досягнення загальної мети дослідження.

Наукова новизна результатів дослідження полягає в тому, що розроблено окремі компоненти методики навчання майбутніх учителів інформатики застосування систем доповненої реальності для розробки інтерактивних навчальних матеріалів.

Практичне значення одержаних результатів полягає в тому, що:

1. Розроблено складові факультативного курсу «Розробка програмних засобів віртуальної та доповненої реальності» для майбутніх учителів інформатики.
2. Результати дослідження можуть бути використані на лабораторних заняттях із дисциплін циклу професійної підготовки вчителя інформатики, що передбачають опанування комп'ютерної графіки.

3 Доповнена реальність: історія та напрями застосування

Як зазначають автори [31], використання мобільних Інтернет-пристроїв розширює межі традиційного інформаційно-освітнього середовища університету до мобільно орієнтованого – відкритої багатовимірної педагогічної системи, що включає психолого-педагогічні умови, мобільні інформаційно-комунікаційні технології і засоби навчання, наукових досліджень та управління освітою, і забезпечує взаємодію, співпрацю, розвиток особистості викладачів і студентів у процесі вирішення освітніх та наукових завдань у будь-який час та у будь-якому місці. Одним із шляхів підвищення ефективності мобільно орієнтованого навчального середовища є застосування технології доповненої реальності, що надає можливість об'єднання реальних та віртуальних засобів навчання за допомогою мобільних Інтернет-пристроїв. Мобільні Інтернет-пристрої

реалізують концепцію мобільного навчання – навчання незалежно від часу та місця.

Концепція віртуальної та доповненої реальності розвивається з 1960-х рр. і вважається дуже перспективним, потужним і корисним інструментом, особливо в освіті. На відміну від віртуальної реальності, доповнена не створює повністю віртуальне середовище, а поєднує віртуальні елементи з реальним світом: до реального оточення користувача додаються віртуальні об'єкти, що змінюються унаслідок його дій.

Віртуальна реальність – середовище, у якому за допомогою комп'ютера моделюється фізична присутність людини у певному місці реального або уявного світу.

Доповнена реальність – це:

- a. штучне середовище, створене шляхом об'єднання об'єктів реального світу та даних, згенерованих комп'ютером;
- b. синтетичне середовище – різновид віртуального середовища (віртуальної реальності), у якому об'єкти фізичного (реального) доповнюються (або підтримуються) комп'ютерно генерованими сенсорними даними (звук, відео, графіка, позиція);
- c. поєднання фізичних та цифрових просторів у семантично пов'язаних контекстах, для яких об'єкти асоціацій розташовані у реальному світі [8].

Піонером віртуальної реальності можна вважати Мортон Леонард Хейліга (Morton Leonard Heilig), який у 1957 році [19] сконструював Сенсораму (Sensorama Simulator [41], рис. 1) – перший багатосенсорний (мультимодальний) пристрій з ефектом присутності (запах, вібрація, звуки, відео), а у 1960 році запатентував Телесферу (Telesphere Mask, рис. 2) – наголовний дисплей (стереоскопічне відео, стереозвук, широкий кут огляду).



Рис. 1. Сенсорам М. Л. Хейліга



Рис. 2. Телесфера М. Л. Хейліга

У статті [19] 1955 р. «Кіно майбутнього» М. Л. Хейліг наводить основні модальності, за яких відбувається сприйняття людиною об'єкту як «джерела багатьох форм енергії», провідними з яких є промені світла, молекули запаху, смакові рецептори звукові хвилі, тепло та тиск (рис. 3).

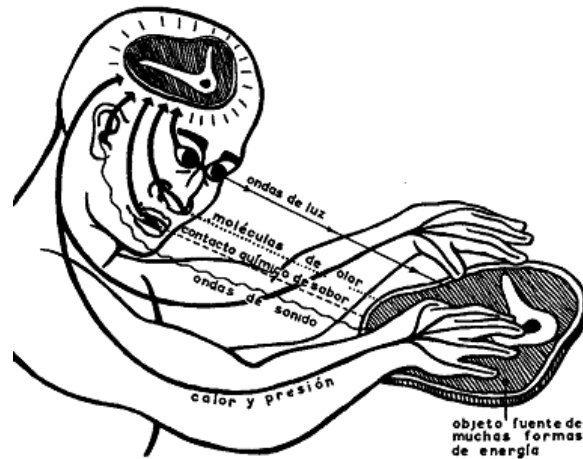


Рис. 3. Модальності сприйняття об'єкту (за [19, с. 280])

На рис. 4 подано «кінематографічне занурення майбутнього», за якого технології стають посередником у сприйнятті людиною об'єкту.

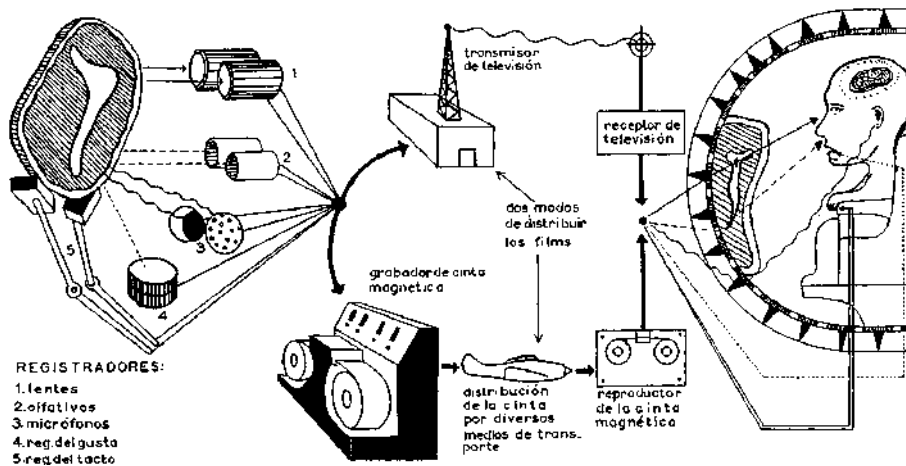
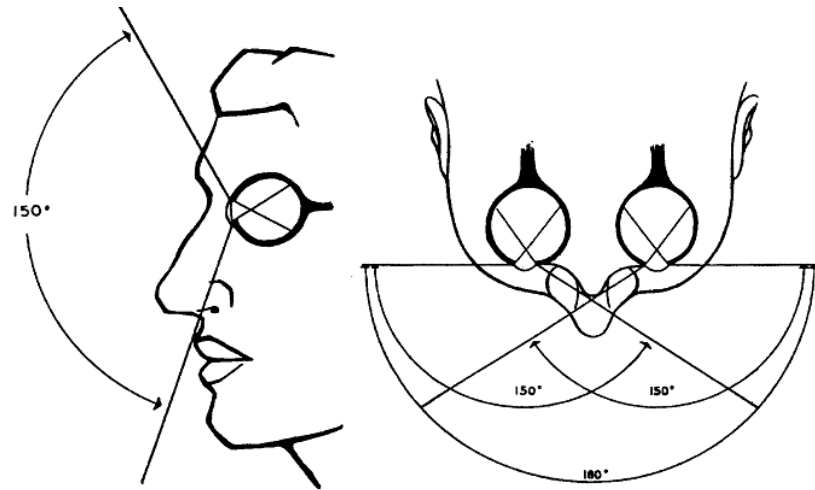


Рис. 4. Кінематографічне занурення (за [19, с. 281])

М. Л. Хейлігом було встановлено технологічні вимоги не лише для поля зору (рис. 5), а й порядок, за якого доцільно подавати різні модальності об'єкту («піраміда сприйняття», рис. 6).



ANGULO DE LA VISION VERTICAL. FILMANDO EN 3-D CINEMASFERA

Рис. 5. Вертикальні (ліворуч) та горизонтальні (праворуч) кути огляду у кінематографічному зануренні М. Л. Хейліга (за [19, с. 284])

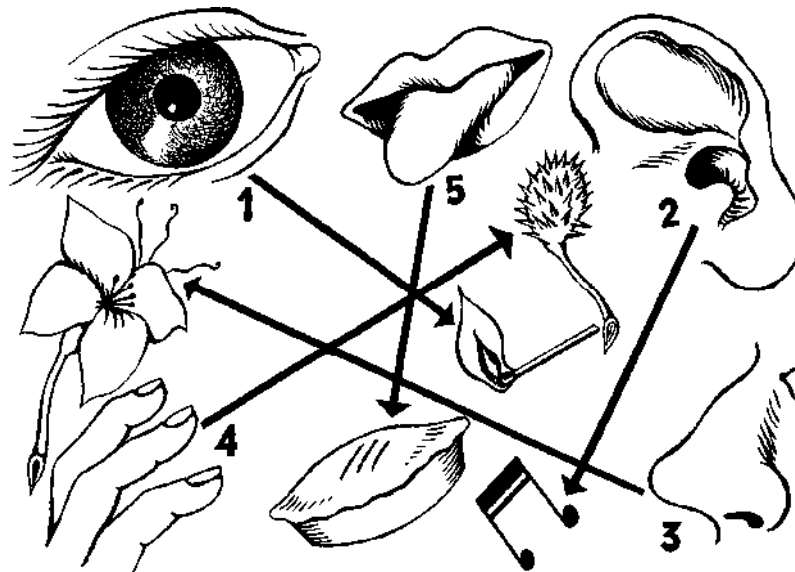


Рис. 6. «Піраміда сприйняття» – впорядкування модальностей за їх внеском у загальну перцепцію (за [19, с. 285])

Крім порядку сприйняття, М. Л. Хейлін встановив співвідношення різних каналів сприйняття: зір – 70 %, слух – 20 %, запах – 5 %, дотик – 4 %, смак – 1 %.

«Батько» сучасних інтерфейсів користувача А. Е. Сазерленд (Ivan Edward Sutherland) у піонерській роботі 1968 року вказує, що це вимагає створення

віртуальних інструментів або компонентів, керованих користувачем, для виконання певних дослідів, проведення експерименту тощо [46]. Так, у 1965 році А. Е. Сазерленд увів концепцію ідеального (остаточного) дисплею (ultimate display), з'єднаного із комп'ютером для полегшення знайомства із поняттями, які неможливо реалізувати у фізичному світі: «Ідеальним дисплеєм буде, звичайно, кімната всередині якої комп'ютер може контролювати існування матерії. Крісло, зображене в такій кімнаті буде достатньо зручними, щоб на ньому сидіти. Наручники, зображені в такій кімнаті, будуть стримувати, а куля, зображена в такій кімнаті, буде фатальною. За відповідного програмування такий дисплей міг буквально стати Країною Чудес, до якої подорожувала Аліса» [47, с. 507-508]. Опис дисплею, наведений А. Е. Сазерлендом, включає як візуальні, так й кінестетичні стимули. Останнє стимулювало Ф. П. Брукса (Frederick Phillips "Fred" Brooks Jr.) розпочати у 1967 році в Університеті Північної Кароліни проект GROPE для дослідження використання кінестетичної взаємодії як засобу, що допомагає біохімікам «відчувати» взаємодію між протеїновими молекулами (рис. 7). Свій варіант дисплею Ф. П. Брукс назвав гаптичним – таким, що надає такі, відчуття як дотик, температура, тиск тощо, опосередковані шкірою, м'язами, сухожиллями або суглобами [5, с. 177-178].

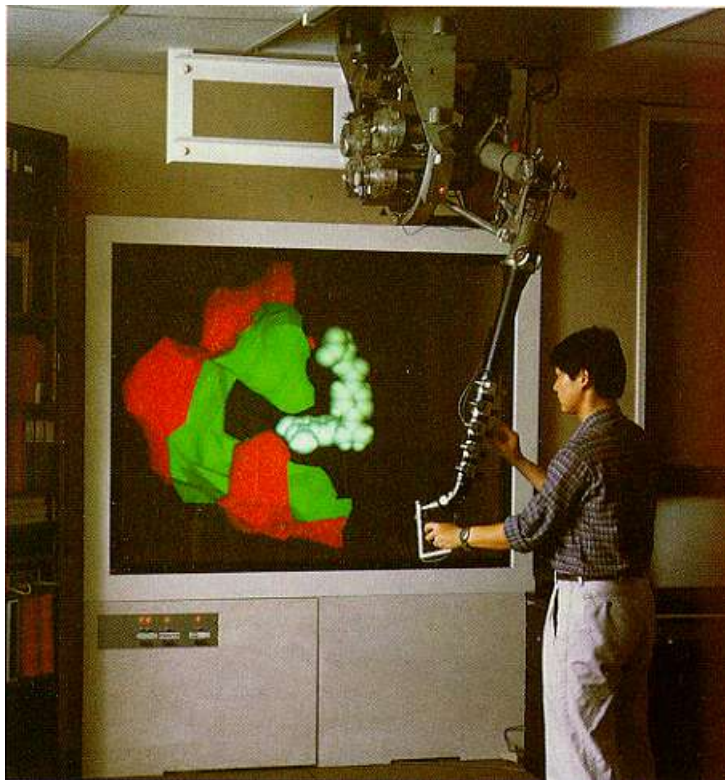


Рис. 7. Гаптичний дисплей GROPE-III

З 1965 року по 1968 рік А. Е. Сазерленд працював у Гарвардському університеті на посаді асоційованого професора електричної інженерії, де за допомогою свого студента Р. Ф. Спрула (Robert Fletcher "Bob" Sproull) у 1968 році створив перший варіант ідеального дисплею – шолом доповненої реальності (head-mounted display) – поєднання фізичних та цифрових просторів у семантично пов'язаних контекстах, для яких об'єкти асоціації розташовані у реальному світі. На відміну від віртуальної реальності, доповнена не створює повністю віртуальне середовище, а поєднує віртуальні елементи з реальним світом: до реального оточення користувача додаються віртуальні об'єкти, що змінюються унаслідок його дій. А. Е. Сазерленд у роботі [46] вказує, що це вимагає створення віртуальних інструментів або компонентів, керованих користувачем, для виконання певних дослідів, проведення експерименту тощо. Розроблений ним шолом віртуальної та доповненої реальності має влучну назву «Дамоклів меч» (The Sword of Damocles) – через велику вагу та розміри механізм був стаціонарно змонтований над користувачем (рис. 8). Середовище доповненої реальності створювалось шляхом накладання простих комп'ютерних моделей на зображення реального світу (рис. 9).

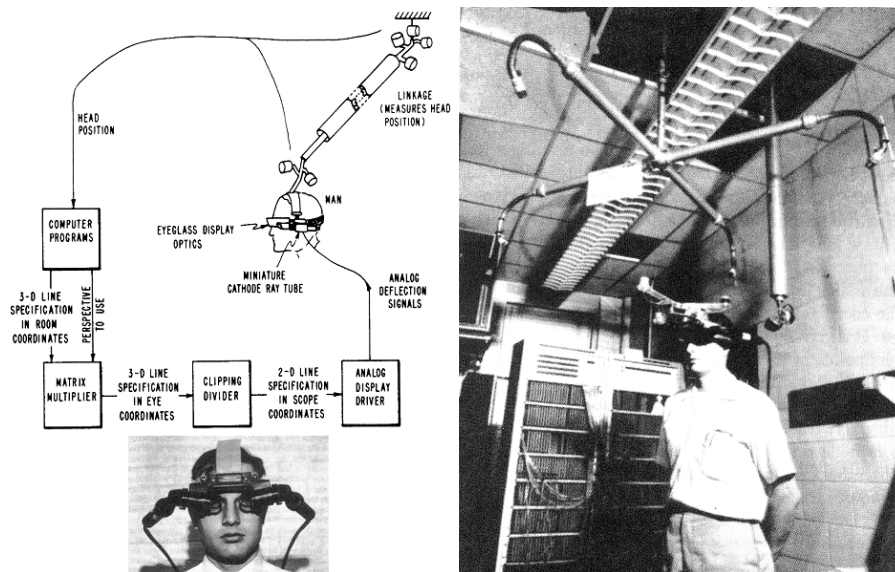


Рис. 8. Загальна схема роботи та зовнішній вигляд «Дамоклового мечу» А. Е. Сазерленда та Р. Ф. Спрула [46, с. 296-298]

Інший із студентів А. Е. Сазерленда, Д. Коен (Danny Cohen), був першим, хто розробив візуальні симулятори польоту (рис. 10) та радара. Робота над авіасимулятором привела до розробки алгоритму Коена-Сазерленда для тривимірної відсікання ліній [9; 44].



Рис. 9. Комп'ютерна модель кубу, що доповнює оточуюче середовище (кадр, знятий через камеру «Дамоклового меч»)»

У 1968 році А. Е. Сазерленд став співзасновником та віце-президентом Evans and Sutherland Computer Corporation, об'єднавшись із Д. Евансом (David Evans) для створення центру досліджень комп'ютерної графіки при Університеті Юти, де з 1968 по 1974 рік А. Е. Сазерленд працював професором комп'ютерних наук. Компанія проводила піонерську роботу в галузі апаратного забезпечення для прискорення тривимірної графіки реального часу та створення принтерних мов. Серед колишніх працівників Evans and Sutherland Computer Corporation – майбутні засновники Adobe Дж. Е. Варнок (John Edward Warnock) та Silicon Graphics Дж. Г. Кларк (James Henry Clark).

«Дослідження – це весело! Як і в командному спорті, полювання на нові знання дає мету, товариство, розмову, змагання та визнання. Пошук нових знань приносить радість новизни, краси, простоти, розуміння, а іноді – й користі. ... Ми всі повинні пам'ятати, що дослідження – це людські зусилля, яким загрожують технічні та емоційні ризики і розчарування. Скоротіть важку роботу, викориніть розчарування, надихніть, надайте підтримку та визнайте досягнення, щоб досягти лояльності та результатів» [48]. Серед його студентів того часу були: винахідник першої об'єктно-орієнтованої мови програмування Smalltalk А. К. Кей (Alan Curtis Kay), автор методу тонування поверхонь А. Гуро (Henri Gouraud), розробник перших практичних методів екранного згладжування Ф. Кроу (Franklin C. Crow), відкривач текстур та В-сплайнів Е. Е. Кетмелл (Edwin Earl Catmull), розробник першої реалістичної 3D-анімації людського обличчя Ф. А. Парк (Frederic Ira Parke).

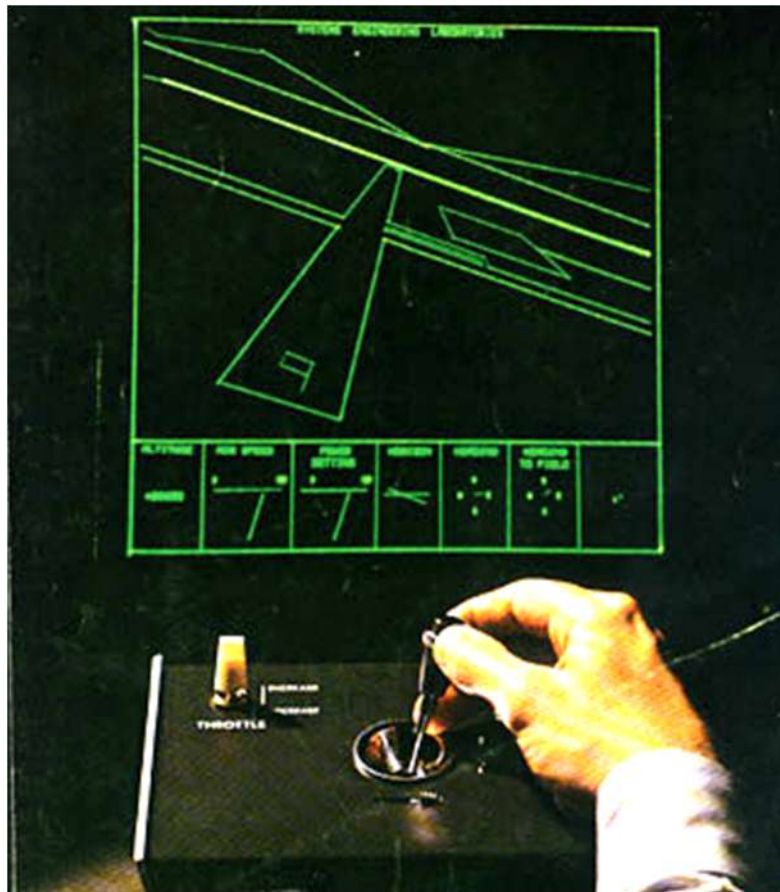


Рис. 10. Приземлення у симуляторі польоту Д. Коена (за [17])

У 1975 році М. Крюгер (Myron W. Krueger) створив «лабораторію штучної реальності» Videoplacе [25, с. 125]. Головна ідея проекту, що розроблявся з 1969 року – створення штучної реальності, яка оточувала користувачів і відповідала на їхні рухи та дії, не обтяжуючись використанням окулярів або рукавичок. Засоби: проектори, відеокамери, апаратні засоби спеціального призначення та екранні силуети користувачів для їх подання в інтерактивному середовищі. Користувачі в окремих кімнатах лабораторії могли взаємодіяти між собою за допомогою цієї технології. Переміщення користувачів, записаних на відео, представлялись силуетами у середовищі штучної реальності (рис. 11).

Стосовно використання адаптивних середовищ у навчанні М. Крюгер зазначав наступне: «Адаптивне середовище має величезний потенціал для освіти. Вся наша система освіти будується на припущенні, що тридцять дітей будуть сидіти в одній кімнаті по шість годин на день і вчитися. Це явище ніколи не спостерігалось в природі, і це є виключенням у класі, де вчителі стикаються з природним бажанням дітей бути активними. Адаптивне середовища пропонує

навчальну ситуацію, в якій заохочується фізична активність. ... Хоча приклади того, як можна використовувати середовище для викладання традиційних предметів, легко отримати, їх значення полягає не тільки в їх здатності автоматизувати традиційне навчання. Більш того, вони можуть революціонізувати те, чого ми навчаємо, а також те, як ми вчимо. Оскільки середовище може визначати цікаві відносини і змінювати їх складними способами, повинна бути можливість створювати взаємодії, які збагачують концептуальний досвід дитини. Це дало б дитині більш потужні інтелектуальні структури для організації конкретної інформації, яку він отримає пізніше. Метою було б набуття дитиною досвіду, а не годування її фактами» [26, с. 432].

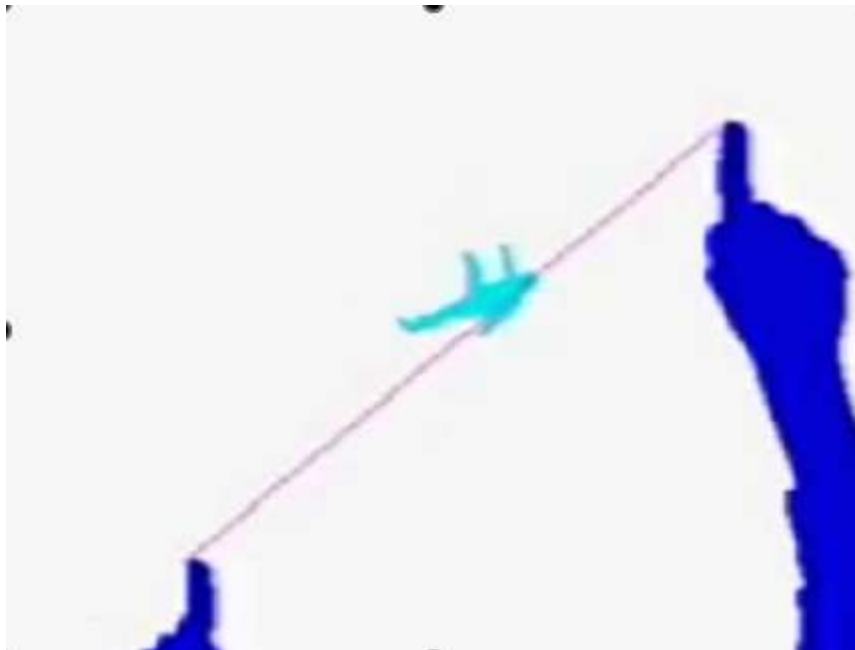


Рис. 11. «Людина на нитці» (з проекту Videoplace)

У 1985 році С. С. Фішером (Scott S. Fisher) було реалізовано ефект телеприсутності у проекті VIEW – Virtual Environment Workstation (проект NASA із повним сенсорним зануренням). Засоби: наголовний дисплей із стереоскопічним зображеннями, 3D аудіо, система розпізнавання мовлення, dataglove – провідна рукавичка, яку користувач застосовує для захоплення віртуальних об'єктів у кіберпросторі (рис. 12). Ця багатосенсорна взаємодія з кібернетичними пристроями створила потужну ілюзію входження в оцифрований ландшафт. Використовуючи концепцію М. Л. Хейліга, С. С. Фішер зробив значний прогрес у тому, що він назвав «телеприсутністю» – проєкцією себе у віртуальний світ (рис. 13).



Рис. 12. Ранній прототип VIEW

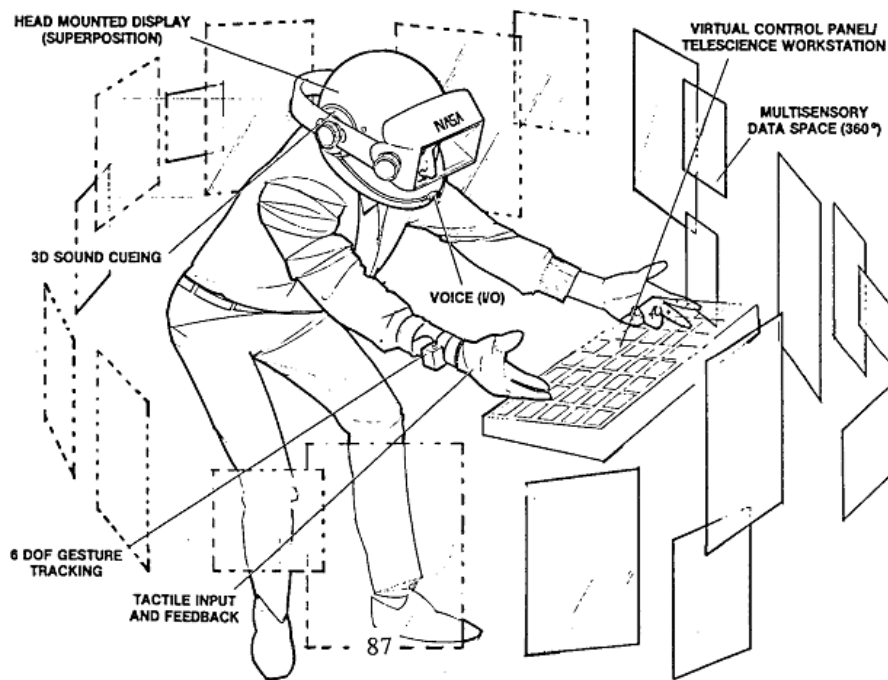


Рис. 13. Схема реалізації телеприсутності (за [15, с. 87])

Недостатня мобільність технології доповненої реальності стала основною перешкодою її широкого поширення – більше 30 років дослідження у цій галузі не виходили за межі окремих лабораторій. Теоретичним підсумком цього етапу стала робота П. Мілгрема та Ф. Кішіно: автори описують простір між реальним

та віртуальним світом (називаючи його комбінованою реальністю), у якому доповнена реальність є більш близькою до реального (немодельованого) світу, а доповнена віртуальність – до віртуального (повністю модельованого) світу [29]. Дана класифікація отримала назву таксономії Мілгрема-Кішіно (рис. 14).

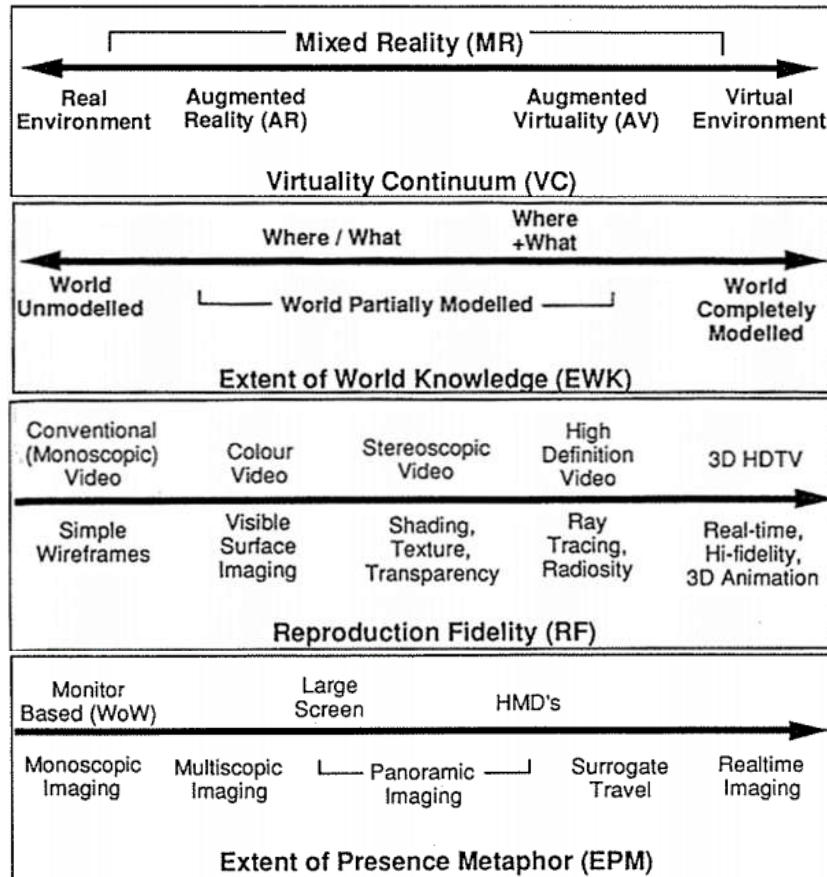


Рис. 14. Класифікація видів синтетичної реальності (за [29])

4 Стратегії використання засобів доповненої реальності у професійній підготовці майбутніх фахівців

Етапним у розвитку віртуальної та доповненої реальності став ARQuake Б. Томаса (Bruce H. Thomas) [49] (2000 рік) – перший проект, що «вийшов на вулицю» [35]: з появою мобільних пристроїв у 1990-х рр. виникли технологічні передумови для використання технології доповненої реальності поза межами спеціалізованих лабораторій – у мобільному просторі Інтернет-користувача. На основі технології доповненої реальності були створені мобільні програмні

засоби, призначені для вивчення різних дисциплін (соціально-гуманітарних, фундаментальних та фахових). За допомогою таких засобів надаються відомості про навчальні об'єкти та їхні характеристики. У ряді проектів, реалізованих у Північній Америці та Європі, мобільні пристрої використовувалися для візуалізації віртуальних об'єктів доповненої реальності. Так, за допомогою програмного забезпечення для мобільних пристроїв майбутні інженери могли бачити, де розташовуються опори мостів при їх візуальному огляді під різними кутами [8; 27; 37].

Мобільні пристрої надають можливість розширити межі традиційного (аудиторного) навчання шляхом перенесення його в мобільне навчальне середовище. Відштовхнувшись від ідеї аудіогіда в музеях, що розповідає відвідувачам про найцікавіші експонати або картини, розробники створили мобільні програмні засоби, призначені для вивчення різних дисциплін, як соціально-гуманітарних (зокрема, історії), так і фундаментальних (зокрема, хімії) та фахових. Такі засоби перетворюються на своєрідного екскурсовода, надаючи відомості про найважливіші об'єкти, їх композиції, конструкції і значення. Інші засоби використовуються при вивченні ботаніки, надаючи відомості про рослини у міру знайомства з ними у середовищі природного проживання. Мобільні Інтернет-пристрої, по суті, надають буквальне значення вислову про те, що світ – це навчальний клас. У декількох проектах, реалізованих в Північній Америці і Європі, мобільні пристрої використовувалися для занурення у доповнену реальність. Спираючись на дані геолокації, програми виявляли процеси і структури, що мають місце у фізичному світі, але невидимі неозброєним оком. Наприклад, за допомогою програмного забезпечення для планшетних комп'ютерів майбутні інженери могли «бачити», де розташовуються опори мостів при їх візуальному огляді під різними кутами.

М. Т. Рестіво та іншими авторами [38] було розглянуто можливості застосування технології доповненої реальності у навчанні розділу «Електрика» курсу фізики. Дослідники вказують, що, незважаючи на широке поширення дослідницького підходу у навчанні, студенти не завжди у змозі виконати експеримент аудиторно через брак часу або матеріалів. Виконання експериментальної роботи у позанавчальний час несе додаткові ризики, особливо при роботі з небезпечними матеріалами. Використання сучасних технологій надає безпечний спосіб виконання експериментів як під керівництвом викладача, так і самостійно. Онлайн-експерименти засобами доповненої реальності (рис. 15) та сенсорних пристроїв візуалізують для користувачів реальні дослідження і спрямовані на надання студентам можливості спостерігати й описувати роботу реальних систем при зміні їхніх параметрів та часткову заміну матеріальних ресурсів та експериментальних установок об'єктами доповненої реальності.

За посиланням https://remotelab.fe.up.pt/#ar_exp можна знайти приклади використання доповненої реальності на лабораторних роботах із фізики. В якості маркерів пропонується використання картонних аркушів із роздрукованими на них створеними у 2D-середовищі або згенерованими простими зображенням (зокрема, вони можуть бути подібні до QR-кодів).

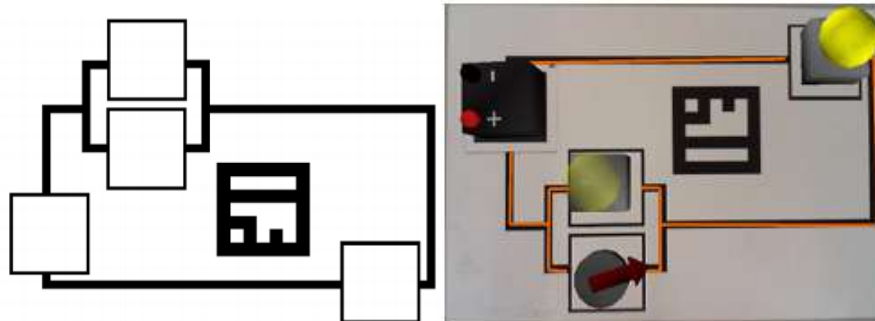


Рис. 15. Електрична схема у доповненій реальності на <http://onlinelab.fe.up.pt/>

Робота з об'єктами доповненої реальності полягає, наприклад, у тому, щоб, використовуючи маркери як компоненти конструктора, зрозуміти принципи складання схеми електричного кола, змодельовати різні ситуації, після чого зібрати коло з реальних компонентів. У системі доповненої реальності користувач може керувати лабораторною установкою шляхом зміни положення перемикача, комбінування різних елементів, зміни положення джерела струму, його полярності тощо простим розкладанням, перекладанням та обертанням маркерів. Простота складання електричних кіл та швидка зміна конфігурації схеми дозволяє відразу ж проаналізувати результати роботи у кожній конкретній конфігурації. Такі поняття, як напрям струму, напрям обертання ротора двигуна, відкритий/закритий контур, паралельне/послідовне з'єднання легко спостерігаються і перевіряються.

Будь-який засіб доповненої реальності може бути навчальним об'єктом, якщо він є керованим та сприяє взаємодії користувача з реальними об'єктами з метою вивчення їх властивостей у процесі експериментального дослідження. Якщо ці вимоги виконуються, когнітивні й емоційні переживання можуть забезпечити нове розуміння того, що студенти вивчають. Застосування засобів доповненої реальності:

- надає можливість підвищити реалістичність дослідження;
- забезпечує емоційний та пізнавальний досвід, що сприяє залученню студентів до систематичного навчання;
- надає коректні відомості про систему у процесі експериментування з нею;
- створює нові способи подання реальних об'єктів у процесі навчання [38, с. 69-70].

Т. Різов та Є. Різова [39], розглядаючи використання доповненої реальності у навчанні інженерної графіки, вводять поняття «підготовленої» та «непідготовленої» сцени (віртуального простору моделювання). Якщо програмний засіб доповненої реальності планується використовувати у «непідготовленій» сцені (як правило, поза межами аудиторії), то для визначення та відстеження її стану необхідні додаткові апаратні засоби – гіроскопи, GPS-навігатори, компаси тощо. Для аудиторного застосування доцільно

«підготувати» сцену – у цьому випадку визначення положення й відстеження здійснюється за допомогою відповідних надійних чорно-білих маркерів характерної форми (квадрат або коло), що конкретизується архітектурою програмного забезпечення для їх виявлення та відстеження.

Т. П. Коделл та Д. В. Майзел [7], характеризуючи технологію доповненої реальності, вказують на простоту відображення в ній віртуальних об'єктів у порівнянні із віртуальною реальністю. Розробка об'єкту для системи доповненої реальності виконується у такий спосіб:

- (1) у 3D-середовищі створюється візуальна модель компоненту доповненої реальності;
- (2) у 2D-середовищі створюється простий маркер, що може бути швидко розпізнаний системою доповненої реальності;
- (3) у програмному засобі для підтримки доповненої реальності маркер пов'язується із 3D-моделлю.

При розпізнаванні маркера системою доповненої реальності на екрані пристрою із програмним засобом для підтримки доповненої реальності на зображення розпізнаного маркера накладається відповідна йому 3D-модель. Цей процес реалізується за схемою, поданою на рис. 16.



Рис. 16. Схема реалізації доповненої реальності (за [23])

Використання доповненої реальності на лекційних, практичних та лабораторних заняттях полегшує розуміння студентами креслень, технічної документації та інструкцій з експлуатації. Викладачі, які використовують технологію доповненої реальності на лабораторних роботах із фізики, можуть краще пояснити студентам будову внутрішніх елементів приладів та установок, що забезпечує ефективність навчання майбутніх фахівців. Наприклад, методичні рекомендації (рис. 17) та лабораторні установки (рис. 18) можуть бути середовищем для роботи з доповненою реальністю. Лабораторні стенди або вимірювальні прилади використовують у якості маркерів, зокрема для доповнення їх інструкціями з використання. Доповнена реальність дає сучасне вирішення завдання заохочення студентів до дослідницької діяльності та мотивує студентів до експериментування.

Х. Мартін-Гутьєррес, Е. Гуінтерс та Д. Перес-Лопес [27] зазначають, що доповнена реальність може бути використана для спільної роботи студентів. Особливої актуальності це набуває у процесі виконання лабораторних робіт із потенційно небезпечним обладнанням, що вимагає постійного контролю

діяльності студентів. Реальним лабораторним роботам передують роботи у доповненій реальності шляхом розміщення маркерів на лабораторних установках. Використовуючи маркери, студенти зможуть за допомогою мобільного пристрою візуалізувати інструкції або навчальні матеріали, необхідні для правильного використання та налаштування обладнання.

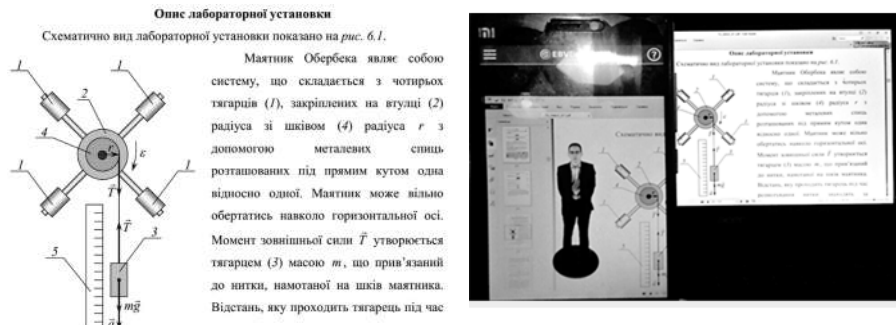


Рис. 17. Використання доповненої реальності у методичних рекомендаціях до лабораторних робіт з фізики (за [23])

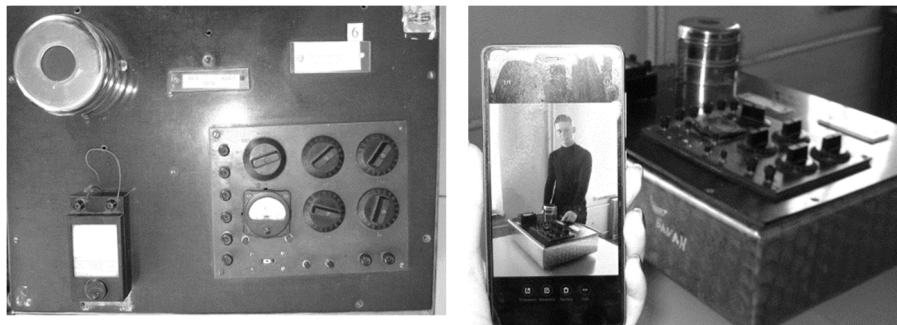


Рис. 18. Використання доповненої реальності на лабораторних роботах з фізики (за [23])

Ж.-М. Сьотат, О. Хьюг, Н. Гуаєль [8, с. 32], розглядаючи застосування доповненої реальності для активізації навчання, виділяють основні напрями її використання:

- середовища моделювання, у яких поєднуються можливості викладання, навчання, комунікації з ігровими елементами;
- підтримка наукових досліджень та експериментального підходу;
- перевірка моделі на адекватність;
- набуття технічних навичок.

Д. Вейдліх (Dieter Weidlich), С. Шерер (Sandra Scherer) та М. Вабнер (Markus Wabner) у [51] описують досвід покращення процесу розробки деталей машин з

використанням систем віртуальної та доповненої реальності Хемницького технічного університету, в якому розроблено нові методи візуалізації для вивчення результату моделювання методом скінченних елементів шляхом занурення у середовище через мобільні пристрої доповненої реальності. Основною метою розробки програмного забезпечення була візуалізація напрямку та градієнту напружень 3D-гліфами. Метод скінченних елементів є чисельним методом інженерного аналізу, що використовується для багатьох типів задач, таких як визначення навантажень та зрушень у механічних об'єктах, або теплопередачу та потокової динаміки.

Визначення механічних навантажень є основою для аналізу поведінки сил у галузі механічної інженерії. Гліф є способом графічного кодування числової інформації. Гліф є графічним елементом, що в змозі передавати велику кількість атрибутів даних шляхом їх відображення (форму, колір, орієнтацію, позицію тощо). Використання гліфів надає можливість відображення багатовимірних тензорів, що відображають власні вектори та значення тензорів, використовуючи форму, розмір, орієнтацію та характеристику поверхні геометричних примітивів, таких як куби та еліпси (рис. 19).

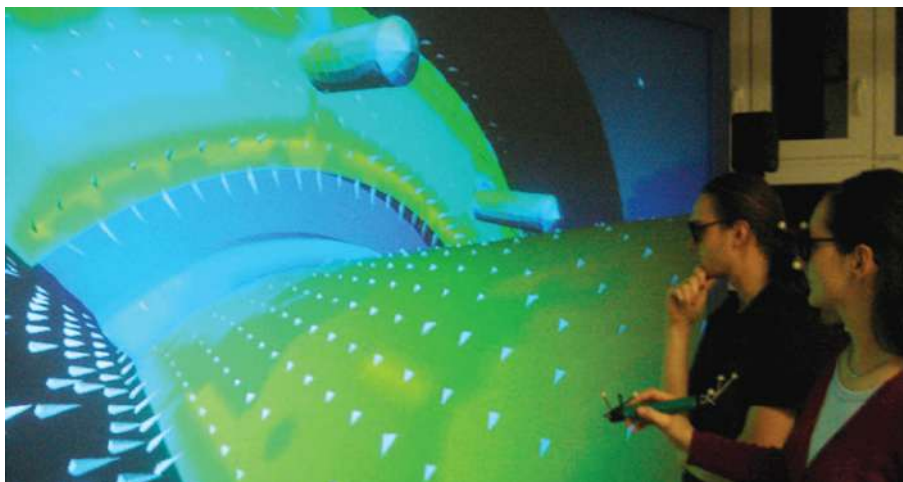


Рис. 19. Візуалізація результатів скінченно-елементного аналізу: вершини гліфів, спрямовані назовні, відображають навантаження розтягування, а гліфи, спрямовані усередину – навантаження стиску

Для представлення багатовимірних даних можуть бути використані різні геометричні примітиви, такі як кубоїди, тетраедри, сфери та лінії. Це потребує попереднього вивчення відповідності обраного гліфу до даних, що відображаються. Перевірка показала, що тетраедр добре підходить для візуалізації напрямку та градієнту навантажень, тому що вершина тетраедру вказує точний напрям. Окрім напрямку навантаження, 3D-гліфи можуть виявляти характер навантаження (стиск або розтягування).

Розроблене авторами [51] програмне забезпечення надає можливість переключатися між результатами структурного та термічного аналізу та порівнювати їх з реальним фізичним об'єктом. На рис. 20 показано накладання скінченно-елементної моделі на реальну систему: чим більш «гарячим» є колір, тим більші навантаження. Чорно-білий маркер в руці користувача необхідний для позиціонування результатів аналізу.

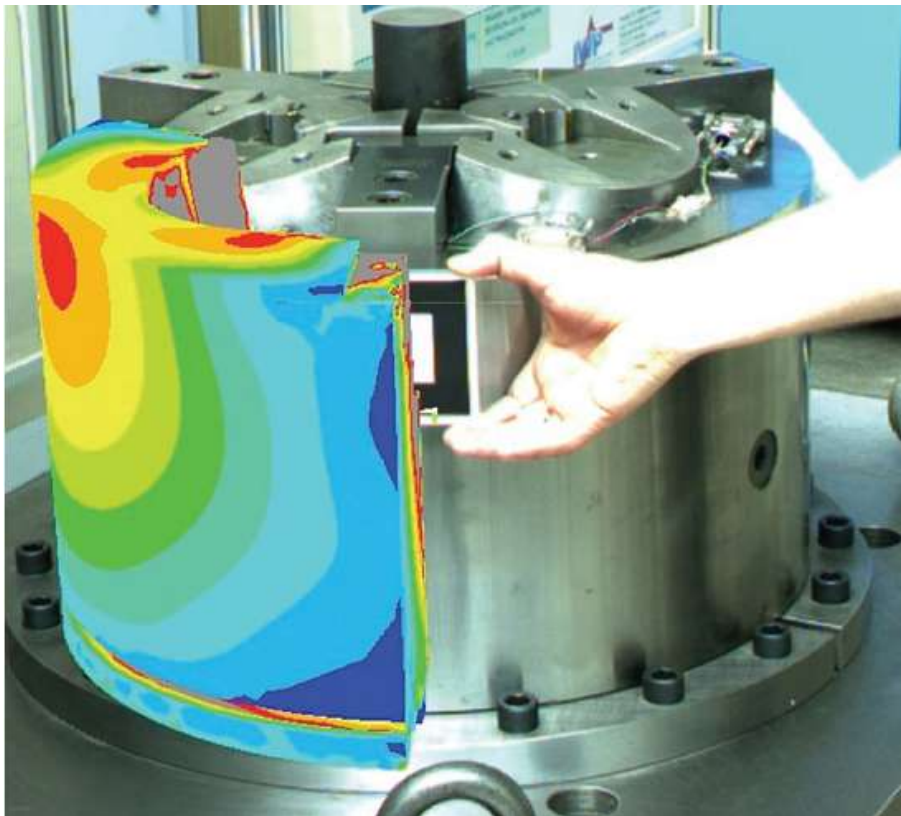


Рис. 20. Відображення результатів моделювання на реальний об'єкт

М. Фіорентіно (Michele Fiorentino), Дж. Монно (Giuseppe Monno) та А. Е. Ува (Antonio E. Uva) у статті [14] виділяють 6 основних способів використання доповненої реальності в інженерній діяльності, для кожного з яких окреслюються такі аспекти, як апаратна конфігурація, спосіб доповнення, рівень інтерактивності TUI/GUI (TUI – Tangible User Interface, матеріальний інтерфейс користувача; GUI – Graphical User Interface, графічний інтерфейс користувача), область застосування, підтримка фізичної співпраці та віддалене співробітництво.

1. Доповнений користувач

Користувач носить окуляри доповненої реальності, підключені до носимого комп'ютеру. Проглядні дисплеї дозволяють користувачеві бути в курсі справжнього промислового середовища. Ця конфігурація забезпечує максимальну мобільність користувача, дозволяючи йому працювати у великій робочій області з вільними руками. База інженерних даних доступна через бездротову мережу. Взаємодія досягається переважно через TUI без графічного інтерфейсу або з обмеженим GUI. Передбачувані застосування для цієї конфігурації: перевірка, навчання та ін. Недоліки можуть включати роздільну здатність дисплея, обмежену область перегляду та надійність оптичного відстежування у агресивному виробничому середовищі (наприклад, пил, електричні шуми, погане освітлення тощо).

У іншій конфігурації користувач тримає кишенькову (подібну до ліхтаря) камеру та носимий комп'ютер, підключений до мережі. Користувач може вільно переміщатися в промисловому середовищі та проводити телеконференції з іншими (віддаленими) користувачами. Різниця в порівнянні з попередньою конфігурацією – це мобільність точки зору. Користувач може перемістити камеру в промислове середовище, досягаючи потенційно будь-якого місця у межах бездротового покриття. Локальне відстеження забезпечується маркерами (в майбутньому можуть бути активні маркери RFID) і транслюється у системи. Цей спосіб використання особливо важливий для технічного обслуговування, де віддалені спеціалісти можуть керувати та допомагати користувачеві. Користувач завантажує свою індивідуальну візуалізацію моделі та передає її дистанційно. Головною перевагою цієї конфігурації є максимальна мобільність точки зору. Це також може призвести до нестабільної точки зору через той факт, що користувач повинен тримати камеру. Взаємодія TUI та графічного інтерфейсу також досить обмежена.

2. Мобільне вікно

Користувач тримає планшетний ПК із камерою на задній стороні. Планшетні дисплеї дозволяють користувачеві повністю усвідомлювати справжнє виробниче середовище. Ця конфігурація дозволяє користувачеві забезпечити хорошу мобільність, дозволяючи йому працювати у великій робочій області, але вимагає, щоб принаймні одна рука тримала планшет. База інженерних даних доступна через бездротову мережу. Взаємодія досягається головним чином за допомогою графічного інтерфейсу з планшетом. Передбачувані програми для цього способу: перегляд проекту, перевірка тощо. Недоліки можуть включати вагу планшетного пристрою та обмеження, пов'язані з використанням однієї руки.

3. Доповнений настільний комп'ютер

Користувач працює на настільній робочій станції з камерою, що вказує на вільну ділянку на столі, яка буде доповненою робочою поверхнею, яка обмежена робочим столом користувача, а взаємодія моделі досягається шляхом переміщення TUI (довповнені креслення) та традиційного графічного інтерфейсу

настільного комп'ютера за допомогою миші та клавіатури. За традиційного використання ТUI – це лише підтримка звичайного графічного інтерфейсу користувача, тому даний спосіб пропонується для всіх задач опрацювання інженерних даних, які передбачають активне використання клавіатури для введення числових чи текстових даних: наприклад, проектування деталей, інжиніринг, чисельний аналіз тощо. Головною перевагою цього способу є схожість з традиційним робочим середовищем, що надає легкий доступ навіть для пересічного користувача, для якого матеріальний інтерфейс до 3D-моделей є простішим та інтуїтивнішим. Обмежувальним фактором є те, що даний спосіб повинен бути реалізований у офісному середовищі.

4. Доповнена майстерня

Цей спосіб подібний до попереднього щодо налаштування апаратного забезпечення, але призначений для виробничого середовища, а не офісного приміщення. Користувач знаходиться на робочому місці на виробничій лінії, де немає клавіатури чи миші. Користувач може послуговуватися сенсорним екраном на промисловому моніторі та матеріальними доповненими кресленням. Також можуть бути використані промислові пульти. Основними перевагами є: обидва руки користувача є вільними, є можливість відображення високоякісного рендерінгу 3D-моделі та інженерних даних, зручне робоче середовище, подібне до традиційного. Ідеальним застосуванням може бути перевірка якості чи керована збірка.

5. Доповнений стіл для спільної роботи

Цей спосіб найкращим чином підтримує спільну робочу область. Він складається зі столу, що виконує функцію спільної розширеної зони та великого екрана. Екран може бути вертикальним або горизонтальним, і в остаточному підсумку мати стереографічне або голографічне відображення. Всі користувачі можуть отримувати доступ до доповненої спільної області зі своїми маркерами, і вони можуть анотувати модель, використовуючи власний комп'ютер для точного графічного введення. Віддалені користувачі можуть приєднатися до групи та працювати із засобами для віртуальних семінарів. Система забезпечить синхронізацію основних цифрових даних, включаючи анотації, чат та історію. Цей спосіб застосовується на етапі маркетинг та перегляду дизайну: спільне робоче середовище може містити віртуальні моделі САПР, реальні попередні виробничі макети, онлайніві інженерні дані та результати моделювання для спільного обговорення. Головними перевагами цього способу є висока підтримка співпраці, співіснування реальних та віртуальних продуктів та соціальний зв'язок реальних зустрічей.

6. Доповнена презентація

Цей спосіб використовує доповідач, який має намір представити певне рішення великій аудиторії. Основним пристроєм візуалізації є великий екран. Управління даними здійснюється, головним чином, за допомогою ТUI у вигляді цифрового

креслення або макета, розміщеного перед доповідачем. Присутні можуть отримати доступ до тих самих цифрових даних з персональних пристроїв візуалізації та можуть додавати анотації, які оновлюються в режимі реального часу для всіх учасників дискусії.

Характеристики кожного способу, запропоновані авторами [14], узагальнені в таблиці 1.

Таблиця 1. Основні способи використання доповненої реальності в інженерній діяльності

Спосіб	Точка зору	Рівень TUI	Рівень GUI	Рівень співпраці	Підтримка віддаленої спільної роботи
Доповнений користувач	мобільна	високий	низький	низький	середній
Мобільне вікно	мобільна	низький	високий	середній	середній
Доповнений настільний комп'ютер	фіксована	середній	високий	низький	високий
Доповнена майстерня	фіксована	середній	низький	низький	низький
Доповнений стіл для спільної роботи	фіксована	високий	середній	високий	середній
Доповнена презентація	фіксована	високий	низький	середній	середній

У процесі набуття теоретичних знань з конструкції електричних машин доцільним є використання мобільних засобів доповненої реальності, які пропонує SIKE Software. Зокрема, їх навчальна система-тренажер з технологією доповненої реальності надає можливість сформувати комплекс знань про будову електродвигунів різних типів та набути навички ідентифікації складових деталей електродвигунів та безпечного, правильного і швидкого порядку збирання і розбирання електродвигунів (рис. 21).



Рис. 21. Навчальна система-тренажер SIKE Software

У підготовці бакалаврів електромеханіки система може бути застосована при проведенні практичних занять, виконанні лабораторних та самостійних робіт, проведенні іспитів з дисципліни тощо, а у процесі виробничого навчання – для теоретичної інтерактивної підготовки працівників, що беруть участь в процесах ремонту, монтажу, демонтажу промислового електрообладнання. Програма надає доступ до 3D-моделей з високим ступенем точності, що повторюють будову реального обладнання. Кожна деталь конструкції має назву та опис, а порядок технологічних операцій відповідає реальному процесу і розроблений спільно з діючими експертами провідних промислових підприємств.

Дана система у якості маркера використовує QR-код, розміщений на спеціальній картці. Інший підхід – використання маркеру сцени, що містить реальний об'єкт, або універсального маркеру.

5 Програмне забезпечення для проектування засобів доповненої реальності навчального призначення

Для розробки засобів доповненої реальності існує багато засобів, найбільш популярними з яких станом на 2018 рік згідно [40] є наступні.

Wikitude SDK [52] є основним продуктом однойменної компанії з 2008 року. SDK включає розпізнавання та відстеження зображень, рендеринг 3D-моделей, відео-накладання, геоінформаційні послуги [33]. У 2017 році Wikitude запустив технологію SLAM (Simultaneous Localization And Mapping – одночасна локалізація та картографування), яка дозволяє розпізнавати і відстежувати об'єкти, у тому числі без маркеру. Для доповненої реальності на основі місцезнаходження, положення об'єктів на екрані мобільного пристрою розраховується за допомогою геопозиції користувача (за допомогою GPS або Wi-Fi), напрямку, у якому користувач рухається (за допомогою компаса) та швидкості (за допомогою акселерометра).

Крос-платформенна SDK доступна для операційних систем Android, iOS та Windows, оптимізована також для кількох окулярів доповненої реальності (Epson Moverio, Vuzix M100, ODG R-7).

Підтримувані засоби розробки: Native API, JavaScript API, Unity3D, Xamarin, Titanium, Cordova.

Версія SDK для використання у некомерційних цілях накладає на зображення водяний знак – в усьому іншому вона еквівалентна комерційній версії вартістю 1990 євро.

ARKit [3] – відносно новий (з 2017 року) засіб від Apple. SDK Apple використовує апаратну програму iPhone / iPad, датчики руху, камеру для активації засобів доповненої реальності.

Підтримувані платформи: iOS 11/12.

ARKit підтримує розпізнавання двовимірних зображень (тригери доповненої реальності з плакатами, знаками, зображеннями) та двовимірне відстеження зображень, тобто можливість убудовувати об'єкти доповненої реальності. SDK також дозволяє розробляти програми, які розпізнають просторові та 3D-об'єкти,

а також розмішувати віртуальні об'єкти у реальному просторі. Поточна версія ARKit 2 надає можливість розробляти багатокористувацькі ігри з доповненою реальністю.

ARKit вільно поширюється для некомерційного використання.

ARCore [2] – новий (березень 2018 року) засіб від Google, своєрідна відповідь на ARKit.

Підтримувані платформи: Android 7.0 та вище, iOS 11 та вище.

ARCore поставляється з трьома основними можливості об'єднання віртуальних і реальних світів:

- відстеження руху – для відстеження положення телефону щодо оточення;
- «розуміння навколишнього середовища» надає можливість телефону визначати розмір та розташування горизонтальних поверхонь;
- оцінка освітленості надає можливість телефону оцінити реальні умови освітлення.

ARCore вільно поширюється.

Vuforia [50] – одна з найпопулярніших платформ для розробки доповненої реальності. SDK реалізує наступні функціональні можливості: розпізнавання різних типів візуальних об'єктів (коробка, циліндр, площина), розпізнавання тексту і оточення, VuMark (комбінація зображення і QR-коду).

За допомогою Vuforia Object Scanner можна сканувати і створювати об'єкти-маркери. Процес розпізнавання може бути реалізований з використанням бази даних (локальне або хмарне сховище).

Можливість реєстрації зображень дозволяє розробникам розташовувати і орієнтувати віртуальні об'єкти, такі, як 3D-моделі і медіаконтент, у зв'язці з реальними образами при перегляді через камери мобільних пристроїв. Віртуальний об'єкт орієнтується на реальному образі так, щоб точка зору спостерігача співвідносилась до них однаковим чином для досягнення головного ефекту – відчуття, що віртуальний об'єкт є частиною реального світу.

Vuforia підтримує різні 2D- і 3D-типи маркерів, включаючи безмаркерні Image Target, тривимірні мішені Multi-Target, а також реперні маркери, які виділяють в сцені об'єкти для їх розпізнавання. Додаткові функції включають виявлення перешкод з використанням так званих «Віртуальних кнопок» («Virtual Buttons»), детектування цілей і можливість програмно створювати і реконфігурувати цілі в рамках самоодифікованого коду.

Підтримувані платформи: Android, iOS, UWP і Unity Editor.

Vuforia надає інтерфейси програмування мовами C++, Java, Objective-C і .NET через інтеграцію з ігровим рушієм Unity. Програми, створені на платформі Vuforia, сумісні з широким спектром пристроїв, включаючи iPhone, iPad, смартфони та планшети на Android з версії 2.2 і процесором, починаючи з архітектур ARM v6.

Всі плагіни і функціональні можливості платформи безкоштовні, але включають водяні знаки Vuforia. Обмеження стосуються тільки кількості об'єктів VuMark і хмарного розпізнавання. Платні версії без водяних знаків коштують від 99 доларів на місяць.

Maxst [22] пропонує два різних інструменти для розпізнавання зображень та середовищ. Створення бази даних здійснюється онлайн через диспетчер відстеження (Tracking Manager). Для сканування 3D-об'єктів використовуються програми для Android і iOS. У редакторі Unity Maxst працює лише з 32-розрядною версією.

Maxst вільно поширюється для некомерційного використання. Вільна версія відрізняється від платної тільки водяним знаком.

Підтримувані платформи: Android, iOS, Windows, Mac OS.

DeepAR [11] складається з DeepAR SDK та DeepAR Studio для редагування контенту з 4 типами ефектів: жорсткі об'єкти, деформовані маски, маски морфинга і ефекти постобробки. Розробники можуть використовувати цей SDK для створення високоякісних лицьових лінз, аналогічних тим, які надають Snapchat та Facebook, а також для різних масок і ефектів для мобільних пристроїв. Цей SDK здатен швидко розпізнавати обличчя в режимі реального часу на основі моделей даних і методів машинного навчання (близько 70 лицьових точок на швидкості 60 кадрів в секунду).

Підтримувані платформи: ПК, Android, iOS, Windows, WebGL.

EasyAR [12], напевно, посідає друге місце після Vuforia SDK за функціональними можливостями. EasyAR підтримує розпізнавання зображень, розпізнавання 3D-об'єктів, сприйняття середовища, хмарне розпізнавання, «розумні окуляри», записи на екрані та ін. Оскільки бібліотека абсолютно вільна, для початку роботи з EasyAR, потрібно лише зареєструвати обліковий запис і створити ключ плагіна вашого пакету.

Підтримувані платформи: Android, iOS, UWP, Windows, Mac і Unity Editor.

Засоби розробки: C API, C++11 API, традиційний C++ API, Java API для Android, Swift API для iOS, Objective-C API для iOS.

ARToolKit [4] – найстаріший (з 1999 року) SDK для розробки засобів доповненої реальності, що реалізує відстеження позиції та орієнтації однієї або декількох камер, відстеження простих чорних квадратних маркерів, відстеження планарних зображень (маркерів у вигляді зображень), калібрування камери, оптичне стерео калібрування, генерацію маркерів, плагіни для Unity і OpenSceneGraph.

Підтримувані платформи: Android, iOS, Linux, Windows, Mac OS, «розумні окуляри».

ARtoolKit поширюється вільно.

Xzing [21] надає SDK, зокрема, для відстеження обличчя у реальному часі через плагін Unity. Xzing включає в себе три основних SDK:

- Augmented Vision – для комп'ютерного зору, розпізнавання і відстеження маркерів;
- Augmented Face – для розпізнавання людського обличчя у відео;
- Magic Face – для нежорсткого відстеження особи, рефакторінга від Augmented Face і поліпшення за допомогою інших функцій, таких як заміна особи, виявлення/відстеження особи та ін.

Підтримувані платформи: Android, iOS, Windows.

Усі SDK Xzing безкоштовні для некомерційного використання.

Серед інших вільно поширюваних засобів розробки доповненої реальності слід відзначити A-Frame, ApertusVR, ArUco, JavaCV, ATOMIC Authoring Tool, Goblin XNA, GRATF, mixare, PTAM, DroidAR, GeoAR, BeyondAR, Mangan, ARma, серед комерційних – 8th Wall, Layar SDK, Catchoom CraftAR AR SDK, Scangine [20].

Великий вибір засобів для проектування засобів доповненої реальності надає можливість вільно комбінувати їх, підключаючи у якості модулів до Unity – багатоплатформенного інструменту для розробки дво- та тривимірних програм, що працює на операційних системах Windows, macOS X та Linux. Створені за допомогою Unity програми працюють під управлінням Microsoft Windows, macOS, Linux, Xbox One, Wii, Wii U, PlayStation 3, PlayStation 4, PlayStation Vita, iOS, Android, WebGL, Tizen, Facebook, TvOS та Nintendo Switch.

Аналізуючи результати дослідно-експериментальної роботи з добору засобів навчання доповненої реальності студентів ЗВО, І. Сурал (Irfan Sural) доходить висновку про доцільність спільного використання Vuforia та Unity. На рис. 22 наведено запропоновану автором схему проектування систем із доповненою реальністю навчального призначення, ключовими компонентами якої є 3D-моделі або відео, посилання на які можуть бути асоційовані з маркерами, QR-коди, SDK Vuforia, інтегрована з Unity3D. Результуючі розробки пропонуються до використання на різних мобільних платформах, насамперед – під управлінням ОС Android [45, с. 569-570].

6 Зміст факультативного курсу «Розробка програмних засобів віртуальної та доповненої реальності» для майбутніх учителів інформатики

Відповідно до обраних програмних засобів, мінімальні технологічні вимоги до апаратного забезпечення факультативу «Розробка програмних засобів віртуальної та доповненої реальності» наступні:

- процесор: Intel Core i7-3770 @ 3.4 GHz або AMD FX-8350 @ 4.0 GHz чи краще;
- оперативна пам'ять: 8 Гб;
- відеокарта: NVIDIA GeForce GTX 780 або AMD Radeon R9 290X (3 Гб відеопам'яті);
- підтримка графіки: DirectX 11 (під управлінням Windows 10);
- операційна система: Windows 7 x64 та вище або macOS 10.11 та вище;
- середовище розробки: Unity 2017.2 та вище.

Зміст факультативного курсу комбінуватимемо із двох основних складових: масового відкритого дистанційного курсу із розробки засобів віртуальної реальності (рис. 23) [10] та посібника із проектування засобів доповненої реальності в Unity 2018 [16]. Відповідно, до курсу будуть входити два змістових модулі:

Змістовий модуль 1. Розробка засобів віртуальної реальності

- Тема 1.1. Віртуальна реальність та ігрові рушії
- Тема 1.2. Фізичні взаємодії та камера
- Тема 1.3. 3D-інтерфейс користувача та позиціонування
- Тема 1.4. 3D-взаємодія з користувачем
- Тема 1.5. Навігація та введення у віртуальній реальності

- Змістовий модуль 2. Розробка засобів доповненої реальності*
- Тема 2.1. Налаштування засобів доповненої реальності в Unity 3D
 - Тема 2.2. Розробка проекту з геопозиціонуванням
 - Тема 2.3. Розробка навчальних матеріалів за допомогою Vuforia
 - Тема 2.4. Розробка для перспективних пристроїв

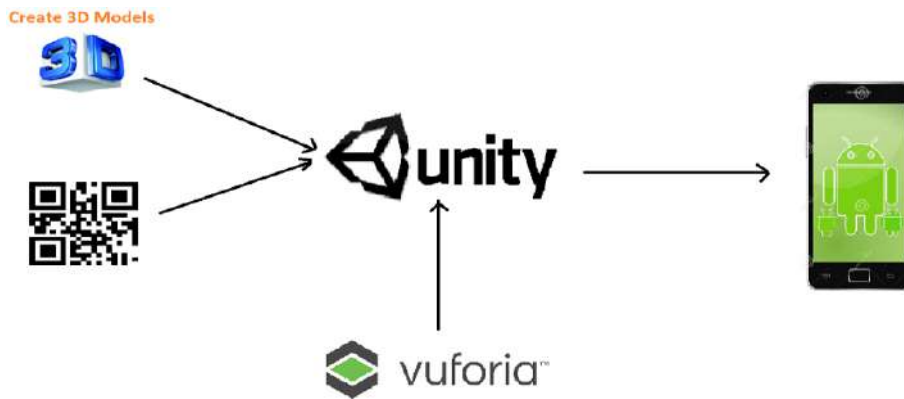


Рис. 22. Схема проектування систем із доповненою реальністю навчального призначення (за [45, с. 569])



Creating Virtual Reality (VR) Apps

Learn the tools and techniques to develop your own professional VR app in Unity 3D.

[UC San Diego](#)

View Course

This course is part of a
Professional Certificate Program

About this course

Build professional VR apps using Unity 3D, a powerful cross-platform 3D engine that provides a user-friendly development environment. In this course, part of the Virtual Reality Professional Certificate program, you will learn how to build a VR engine from the ground up, so you fully understand the entire rendering pipeline from 3D model to pixels in the VR display. We will also cover motion prediction, 3D stereo, lens distortion, time warp and other optimizations for a fluid, realistic VR experience.

You will also learn how to implement the most important VR interaction concepts such as selection, manipulation, travel, wayfinding, menus, and text input in Unity.

🕒 Length:	6 weeks
👤 Effort:	5 to 7 hours per week
💰 Price:	FREE Add a Verified Certificate for \$99 USD
🏛️ Institution:	UCSanDiegoX
📖 Subject:	Computer Science

Рис. 23. Головна сторінка курсу [10]

З метою адаптації змісту [10] для неангломовних студентів було виконано опрацювання та переклад запису відеоуроків з метою створення відповідних лабораторних робіт, розміщених у системі управління електронними навчальними курсами Криворізького державного педагогічного університету.

7 Висновки

У процесі дослідження проблеми професійної підготовки майбутніх учителів інформатики до використання технологій доповненої реальності в освіті розв'язані всі поставлені завдання та отримані результати, узагальнення яких надає можливість зробити наступні висновки:

1. Проведений історико-технологічний аналіз досвіду застосування засобів доповненої реальності для розробки інтерактивних навчальних матеріалів показав, що, саме концепція доповненої реальності спонукала до появи та розвитку нових інтерфейсів доступу людини до комп'ютерних систем. Водночас, незважаючи на значний (більше 60 років) період розвитку теорії та практики розробки таких систем, лише в останні 20 років з'явилися технологічні умови для їх широкого запровадження через масове поширення мобільних Інтернет-пристроїв. Виокремлені у 1970-1980-ті рр. методичні умови застосування систем віртуальної реальності в освіті, апробовані в лабораторних умовах, сьогодні також потребують корекції в умовах масового технологізованого педагогічного процесу.
2. Для розробки засобів доповненої реальності існує багато SDK, провідними з яких є Wikitude, ARKit, ARCore, Vuforia, Maxst, DeepAR, EasyAR, ARtoolKit, Xzing, безкоштовні для некомерційного використання. Великий вибір засобів для проектування засобів доповненої реальності надає можливість вільно комбінувати їх, підключаючи у якості модулів до Unity – багатоплатформенного інструменту для розробки дво- та тривимірних програм. Відповідно, технологічні вимоги для розробки програмних засобів віртуальної та доповненої реальності визначаються обраними SDK та IDE.
3. У процесі підготовки майбутніх учителів інформатики до застосування систем доповненої реальності для розробки інтерактивних навчальних матеріалів доцільно застосовувати інтегрований підхід, за якого проектування із застосуванням стандартних об'єктів виконується у середовищі візуального проектування, а надання стандартним об'єктам нових властивостей та створення нових виконується у пов'язаному із ним середовищем об'єктно-орієнтованого програмування. На сучасному етапі розвитку ІКТ доцільним є спільне використання середовища Unity для візуального проектування, Visual Studio чи подібного середовища програмування, а також платформ віртуальної (Google VR чи подібного) та доповненої (Vuforia чи подібного) реальності. Реалізацію інтегрованого підходу виконано у межах факультативного курсу «Розробка програмних засобів віртуальної та доповненої реальності» для майбутніх учителів інформатики, до складу якого входять два змістових

модулі: «Розробка засобів віртуальної реальності» та «Розробка засобів доповненої реальності».

Проведене дослідження може бути продовжене в напрямі розробки навчально-методичних комплексів із проектування предметно-орієнтованих систем віртуальної та доповненої реальності як складової професійної підготовки.

References

1. Akcayir, M., Akcayir, G.: Advantages and challenges associated with augmented reality for education: A systematic review of the literature. *Educational Research Review*. 20, 1–11 (2017)
2. ARCore - Google Developer | ARCore | Google Developers. <https://developers.google.com/ar> (2018). Accessed 25 Oct 2018
3. ARKit - Apple Developer. <https://developer.apple.com/arkit> (2018). Accessed 25 Oct 2018
4. ARToolKit Home Page. <http://www.hitl.washington.edu/artoolkit> (2007). Accessed 13 Sep 2018
5. Brooks, F.P.Jr., Ouh-Young, M., Battert, J.J., Kilpatrick, P.J.: Project GROPE – Haptic Displays for Scientific Visualization [Electronic resource]. *Computer Graphics*. **24**(4), 177–185 (1990)
6. Buzko, V.L., Bonk, A.V., Tron, V.V.: Implementation of Gamification and Elements of Augmented Reality During the Binary Lessons in a Secondary School. In: Kiv, A.E., Soloviev, V.N. (eds.) *Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018)*, Kryvyi Rih, Ukraine, October 2, 2018. *CEUR Workshop Proceedings*, vol. 2257, pp. 53–60. <http://ceur-ws.org/Vol-2257/paper06.pdf> (2018). Accessed 30 Nov 2018
7. Caudell, T.P., Mizell, D.W.: Augmented reality: An application of heads-up display technology to manual manufacturing processes. In: Nunamaker, J.F., Sprague, R.H. (eds.) *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, January 7-10, 1992. Kauai, Hawaii, volume 2: Software Technology Track, pp. 659–669. IEEE Computer Society Press, Los Alamitos (1992)
8. Cieutat, J.-M., Hugues, O., Ghouaiel, N.: Active Learning based on the use of Augmented Reality Outline of Possible Applications: Serious Games, Scientific Experiments, Confronting Studies with Creation, Training for Carrying out Technical Skills. *International Journal of Computer Applications*. **46**(20), 31–36. <https://hal.archives-ouvertes.fr/hal-00739730/document> (2012). Accessed 25 Nov 2017
9. Cohen, D.: Incremental Methods for Computer Graphics. PhD Thesis, Harvard University. Harvard Report, ESD-TR-69-193. <https://apps.dtic.mil/dtic/tr/fulltext/u2/694550.pdf> (1969). Accessed 5 Apr 2017
10. Creating Virtual Reality (VR) Apps. <https://www.edx.org/course/creating-virtual-reality-vr-apps> (2018). Accessed 25 Oct 2018
11. DeepAR ~ Snapchat Face Filters and Lenses Augmented Reality SDK. <https://deepar.ai> (2018). Accessed 21 Mar 2018
12. EasyAR-Best engine for developing Augmented Reality. <https://www.easyar.com> (2018). Accessed 21 Nov 2018
13. Ewalt, D.M.: *Defying Reality: The Inside Story of the Virtual Reality Revolution*. Penguin Random House, New York (2018)
14. Fiorentino, M., Monno, G., Uva, A.E.: *Tangible Interfaces for Augmented Engineering Data*

- Management. In: Maad, S. (ed.) *Augmented Reality*, pp. 113–128. IntechOpen. <https://cdn.intechopen.com/pdfs/6762.pdf> (2010). Accessed 1 Oct 2018
15. Fisher, S.S.: *Virtual Interface Environment*. In: *Space Station Human Factors Research Review*, December 3 – December 6, 1985. NASA Ames Research Center, Moffett Field California, pp. 85-87. <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19880014769.pdf> (1988). Accessed 6 May 2018
 16. Glover, J.: *Unity 2018 Augmented Reality Projects: Build four immersive and fun AR applications using ARKit, ARCore, and Vuforia*. Packt Publishing, Birmingham (2018)
 17. Greatorex, F.S., Cohen, D.: *Producing Dynamic Perspective Views for Vehicle Simulation*. *Data Processing Magazine*. April, 26–31 (1968)
 18. Haq, H.: In South Korea, all textbooks will be e-books by 2015. *The Christian Science Monitor*. <https://www.csmonitor.com/Books/chapter-and-verse/2011/0706/In-South-Korea-all-textbooks-will-be-e-books-by-2015> (2011). Accessed 17 Sep 2018
 19. Heilig, M.L.: *El Cine del Futuro: The Cinema of the Future*. *Presence: Teleoperators and Virtual Environments*. **1**(3), 279–294 (1992). doi:10.1162/pres.1992.1.3.279
 20. Herpich, F., Guarese, R.L.M., Tarouco, L.M.R.: *A Comparative Analysis of Augmented Reality Frameworks Aimed at the Development of Educational Applications*. *Creative Education*. **8**(9), 1433–1451 (2017). doi:10.4236/ce.2017.89101
 21. Home - XZIMG. <https://www.xzimg.com> (2017). Accessed 25 Aug 2018
 22. Home | MAXST Developer - The Right Choice for Your AR SDK. <https://developer.maxst.com> (2018). Accessed 25 Aug 2018
 23. Hrunтова, Т.В., Ячкало, Ю.В., Стриук, А.М., Пикільняк, А.В.: *Augmented Reality Tools in Physics Training at Higher Technical Educational Institutions*. In: Kiv, A.E., Soloviev, V.N. (eds.) *Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018)*, Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 33–40. <http://ceur-ws.org/Vol-2257/paper04.pdf> (2018). Accessed 30 Nov 2018
 24. Kolomoiets, T.H., Kassim, D.A.: *Using the Augmented Reality to Teach of Global Reading of Preschoolers with Autism Spectrum Disorders*. In: Kiv, A.E., Soloviev, V.N. (eds.) *Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018)*, Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 237–246. <http://ceur-ws.org/Vol-2257/paper24.pdf> (2018). Accessed 30 Nov 2018
 25. Krueger, M.W.: *Artificial Reality*. Addison-Wesley, Reading (1983)
 26. Krueger, M.W.: *Responsive environments*. In: *AFIPS '77 Proceedings of the National computer conference*, June 13-16, 1977, pp. 423–433 (1977). – doi:10.1145/1499402.1499476
 27. Martin-Gutierrez, J., Guinters, E., Perez-Lopez D.: *Improving strategy of self-learning in engineering: laboratories with augmented reality*. *Procedia – Social and Behavioral Sciences*. **51**, 832–839. doi:10.1016/j.sbspro.2012.08.249
 28. Merzlykin, O.V., Topolova, I.Yu., Tron, V.V.: *Developing of Key Competencies by Means of Augmented Reality at CLIL Lessons*. In: Kiv, A.E., Soloviev, V.N. (eds.) *Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018)*, Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 41–52. <http://ceur-ws.org/Vol-2257/paper05.pdf> (2018). Accessed 30 Nov 2018
 29. Milgram, P., Kishino, F.: *A taxonomy of mixed reality visual displays*. *IEICE Transactions on Information Systems*. **E77-D**(12), 1321–1329 (1994)
 30. Mintii, I.S., Soloviev, V.N.: *Augmented Reality: Ukrainian Present Business and Future Education*. In: Kiv, A.E., Soloviev, V.N. (eds.) *Proceedings of the 1st International*

- Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 227–231. <http://ceur-ws.org/Vol-2257/paper22.pdf> (2018). Accessed 30 Nov 2018
31. Modlo, Ye.O., Semerikov, S.O., Shmeltzer, E.O.: Modernization of Professional Training of Electromechanics Bachelors: ICT-based Competence Approach. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 148–172. <http://ceur-ws.org/Vol-2257/paper15.pdf> (2018). Accessed 30 Nov 2018
 32. Modlo, Ye.O., Semerikov, S.O.: Xcos on Web as a promising learning tool for Bachelor's of Electromechanics modeling of technical objects. In: Semerikov, S.O., Shyshkina, M.P. (eds.) Proceedings of the 5th Workshop on Cloud Technologies in Education (CTE 2017), Kryvyi Rih, Ukraine, April 28, 2017. CEUR Workshop Proceedings, vol. 2168, pp. 34–41. <http://ceur-ws.org/Vol-2168/paper6.pdf> (2018). Accessed 15 Sep 2018
 33. Morkun, V., Semerikov, S., Hryshchenko, S., Slovak, K.: Environmental Geo-information Technologies as a Tool of Pre-service Mining Engineer's Training for Sustainable Development of Mining Industry. In: Ermolayev, V., Bassiliades, N., Fill, H.-G., Yakovyna, V., Mayr, H.C., Kharchenko, V., Peschanenko, V., Shyshkina, M., Nikitchenko, M., Spivakovsky, A. (eds.) Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer (ICTERI, 2017), Kyiv, Ukraine, May 15-18, 2017. CEUR Workshop Proceedings, vol. 1844, pp. 303–310. <http://ceur-ws.org/Vol-1844/10000303.pdf> (2017). Accessed 15 Sep 2018
 34. Nechypurenko, P.P., Starova, T.V., Selivanova, T.V., Tomilina, A.O., Uchitel, A.D.: Use of Augmented Reality in Chemistry Education. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 15–23. <http://ceur-ws.org/Vol-2257/paper02.pdf> (2018). Accessed 30 Nov 2018
 35. Piekarski, W.: Interactive 3D Modelling in Outdoor Augmented Reality Worlds. A Research Thesis for the Degree of the Doctor of Philosophy, University of South Australia (2004)
 36. Rashevskaya, N.V., Soloviev, V.N.: Augmented Reality and the Prospects for Applying Its in the Training of Future Engineers. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 192–197. <http://ceur-ws.org/Vol-2257/paper18.pdf> (2018). Accessed 30 Nov 2018
 37. Rassovytska, M., Striuk, A.: Mechanical Engineers' Training in Using Cloud and Mobile Services in Professional Activity. In: Ermolayev, V., Bassiliades, N., Fill, H.-G., Yakovyna, V., Mayr, H.C., Kharchenko, V., Peschanenko, V., Shyshkina, M., Nikitchenko, M., Spivakovsky, A. (eds.) Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer (ICTERI, 2017), Kyiv, Ukraine, May 15-18, 2017. CEUR Workshop Proceedings, vol. 1844, pp. 348–359. <http://ceur-ws.org/Vol-1844/10000348.pdf> (2017). Accessed 15 Sep 2018
 38. Restivo, M.T., Chouzal, F., Rodrigues, J., Menezes, P., Patrão, B., Lopes, J.B.: Augmented Reality in Electrical Fundamentals. *International Journal of Online Engineering (iJOE)*. **10**(6), 68–72 (2014)
 39. Rizov, T., Rizova, E.: Augmented reality as a teaching tool in higher education. *International Journal of Cognitive Research in Science, Engineering and Education*. **3**(1), 7–16 (2015)
 40. Schmidt, J.: Best AR SDK for development for iOS and Android in 2018.

- <https://thinkmobiles.com/blog/best-ar-sdk-review> (2018)
41. Heilig, M.L.: Sensorama Simulator. US Patent 3,050,870, 28 Aug 1962
 42. Shapovalov, V.B., Atamas, A.I., Bilyk, Zh.I., Shapovalov, Ye.B., Uchitel, A.D.: Structuring Augmented Reality Information on the stemua.science. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 75–86. <http://ceur-ws.org/Vol-2257/paper09.pdf> (2018). Accessed 30 Nov 2018
 43. Shapovalov, Ye.B., Bilyk, Zh.I., Atamas, A.I., Shapovalov, V.B., Uchitel, A.D.: The Potential of Using Google Expeditions and Google Lens Tools under STEM-education in Ukraine. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 66–74. <http://ceur-ws.org/Vol-2257/paper08.pdf> (2018). Accessed 30 Nov 2018
 44. Sproull, R.F., Sutherland, I.E.: A clipping divider. In: Proceedings of the AFIPS Fall Joint Computer Conferences, December 9-11, 1968, vol. I, pp. 765–775. Thompson Books, Washington (1968)
 45. Sural, I.: Augmented Reality Experience: Initial Perceptions of Higher Education Students. *International Journal of Instruction*. **11**(4), 565–576 (2018)
 46. Sutherland, I.E.: A head-mounted three dimensional display. In: Proceedings of the AFIPS Fall Joint Computer Conferences, December 9-11, 1968, vol. I, pp. 757–764. Thompson Books, Washington (1968)
 47. Sutherland, I.E.: The Ultimate Display. In: Proceedings of the IFIP Congress, vol. 2, pp. 506–508 (1965)
 48. The Computer History Museum Presents An Evening with Ivan Sutherland | Press Releases | Computer History Museum. <http://www.computerhistory.org/press/an-evening-with-ivan-sutherland.html> (2005). Accessed 1 Apr 2018
 49. Thomas, B., Close, B., Donoghue, J., Squires, J., De Bondi, P., Morris, M., Piekarski, W.: ARQuake: An Outdoor/Indoor Augmented Reality First Person Application. In: Fourth International Symposium on Wearable Computers, Atlanta, USA, Oct 2000, pp. 139–146. IEEE (2000). doi : 10.1109/ISWC.2000.888480
 50. Vuforia | Augmented Reality for the Industrial Enterprise. <https://www.vuforia.com> (2018). Accessed 17 Aug 2018
 51. Weidlich, D., Scherer, S., Wabner, M.: Analyses Using VR/AR Visualization. *IEEE Computer Graphics and Applications*. **28**(5), 84–86 (2008). doi : 10.1109/mcg.2008.89.
 52. Wikitude Augmented Reality: the World's Leading Cross-Platform AR SDK. <https://www.wikitude.com> (2018). Accessed 17 Aug 2018
 53. Zelinska, S.O., Azaryan, A.A., Azaryan, V.A.: Investigation of Opportunities of the Practical Application of the Augmented Reality Technologies in the Information and Educative Environment for Mining Engineers Training in the Higher Education Establishment. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 204–214. <http://ceur-ws.org/Vol-2257/paper20.pdf> (2018). Accessed 30 Nov 2018
 54. Zinonos, N.O., Vihrova, E.V., Pikilnyak, A.V.: Prospects of Using the Augmented Reality for Training Foreign Students at the Preparatory Departments of Universities in Ukraine. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings, vol. 2257, pp. 87–92. <http://ceur-ws.org/Vol-2257/paper10.pdf>. Accessed 30 Nov 2018

Author Index

A		
	Azaryan, Albert A.	116
B		
	Bielinskyi, Andrii O.	37
D		
	Dolotii, Marharyta H.	55
F		
	Fadieieva, Liliia O.	83
	Franchuk, Vasyl M.	128
H		
	Hanistrat, Dmytro O.	94
	Haranin, Oleh M.	143
	Horlo, Anna M.	103
	Hrebeniuk, Bohdan V.	46
	Humenok, Vladyslav K.	87
K		
	Katsko, Oleksandr O.	151
	Kiv, Arnold E.	1
	Kotov, Igor A.	87, 94
	Kurhanov, Dmytro O.	116
	Kuropiatnyk, Danylo I.	70
	Kuznietsov, Vladyslav S.	156
L		
	Lehka, Liudmyla V.	76
M		
	Merzlykin, Pavlo V.	55, 83
	Mintii, Iryna S.	103
	Mintii, Mykhailo M.	162
	Modlo, Yevhenii O.	193
	Moiseienko, Natalia V.	143, 151, 156
O		
	Oleksiienko, Illia V.	128

P	
Petrova, Marharyta Ye.	162
Pirohov, Vladyslav M.	103
S	
Semerikov, Serhiy O.	1, 162, 193
Shokaliuk, Svitlana V.	76
Soloviev, Vladimir N.	1, 37
Somenko, Olena O.	132
Striuk, Andrii M.	1, 11, 109
Syrovatskyi, Oleksandr V.	193
T	
Tyshchenko, Yelyzaveta Yu.	109
V	
Volkova, Nataliia P.	162
Y	
Yechkalo, Yuliia V.	193
Z	
Zelenskyi, Arkadii A.	63
Zelinska, Snizhana O.	193