

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ
ВОЛОДИМИРА ДАЛЯ

Факультет інформаційних технологій та електроніки

Кафедра інформаційних технологій та програмування

Пояснювальна записка
до магістерської дипломної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему: Реалізація механізму формування множини питань при он-лайн тестуванні

Виконав: студент 2 курсу, групи ІСТ-22дм
126 «Інформаційні системи та технології»

(шифр і назва спеціальності)

Брувер Д. Д.

(прізвище та ініціали)

Керівник Лифар В.О.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ – 2023 року

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ
ВОЛОДИМИРА ДАЛЯ

Факультет інформаційних технологій та електроніки
Кафедра інформаційних технологій та програмування
Освітньо-кваліфікаційний рівень магістр
Спеціальність 126 «Інформаційні системи та технології»
(шифр і назва спеціальності)

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТП
_____ д.т.н., доц. Захожай О.І.
(підпис)
« ____ » _____ 2023 р.

ЗАВДАННЯ

на магістерську дипломну роботу студенту

Брувер Данило Дмитрович

(прізвище, ім'я, по батькові)

1. Тема роботи: Реалізація механізму формування множини питань при он-лайн тестуванні, _____
керівник роботи доцент, д.т.н. Лифар Володимир Олексійович,
(вчене звання, науковий ступінь, прізвище, ім'я, по батькові)
затверджені наказом університету від « ____ » _____ 2023 року № _____
 2. Строк подання студентом роботи: 06 грудня 2023 р.
 3. Вихідні дані до роботи: Матеріали науково-дослідної практики, науково-методична література; дані інтернет-мережі .
 4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - 4.1 Вступ
 - 4.2 Аналітичний огляд питання (огляд публічних джерел інформації)
 - 4.3 Основна частина, в якій висвітлити методи, які будуть використовуватися для реалізації проекту.
 - 4.4 Практична частина – огляд технологій, які використовуються під час реалізації проекту.
 - 4.4 Висновки
 - 4.5 Перелік використаних джерел
 5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
-
-

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 20 жовтня 2023р.**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Одержання завдання на виконання роботи	20.10.2023	
2.	Укладання і погодження з керівником плану і етапів виконання роботи	24.10.2023	
3.	Узагальнення даних літературних джерел	28.10.2023	
4.	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху виконання завдання	01.11.2023	
5.	Аналіз технічних засобів та існуючих систем	07.11.2023	
6.	Реалізація практичної частини завдання	24.11.2023	
7.	Укладання, оформлення та погодження пояснювальної записки з керівником	05.12.2023	
8.	Здача пояснювальної записки на кафедру	06.12.2023	
9.	Підготовка доповіді та презентації	09.12.2023	

Студент _____
(підпис)**Брувер Д.Д.**
(прізвище та ініціали)Керівник роботи _____
(підпис)**Лифар В.О.**
(прізвище та ініціали)

РЕФЕРАТ

Магістерська дипломна робота: 81 стор., 3 табл., 17 рис., 54 джерела.

Об'єкт досліджень – процеси збирання та систематизації інформації у системах проведення опитування громадян за допомогою сучасних інформаційно-комунікаційних технологій.

Мета роботи – розробити модель та алгоритми виконання для формування множини питань при он-лайн тестуванні, яка б відрізнялася від існуючих підходів та розширяла можливості розробників для створення тестових платформ або окремих сторінок тестування за вимогами організацій-замовників.

В роботі реалізовано механізм формування множин питань та ключових слів, які дозволяють ідентифікувати відповіді на питання з метою автоматичної обробки відповіді для отримання робочої інформації перед особистісною зустріччю з респондентом.

Прогнозні припущення про розвиток об'єкта дослідження – отримані в роботі дані можуть використовуватися для оперативного пошуку необхідної допомоги стосовно потреб осіб, які фізично чи психологічно постраждали від російської агресії, що особливо актуальним буде під час звільнення Луганської та Донецької областей.

Галузь застосування – опитування з метою збору інформації для підтримки прийняття рішень у ситуаціях, які потребують попередньої підготовки (медицина – прийом хворих, що знаходяться в стадії ремісії, потребують реабілітації або здійснюється пошуку лікарів для консультації за ознаками стабільного стану); освіта та навчання – для розробки тестів та ситуаційних задач, які потребують не шаблонного вирішення.

МАСИВ СЛІВ, МНОЖИНА ПИТАНЬ, МОДЕЛЬМАТЕМАТИЧНА
МОДЕЛЬ ТЕСТУ, МЕНТАЛЬНИЙ ОПЕРАТОР, КРОК, АЛГОРИТМ.

ABSTRACT

Master's thesis: 81 pages, 3 table, 17 figures, 54 sources.

The object of research is the process of collecting and systematizing information in citizen survey systems using modern information and communication technologies.

The goal of the work is to develop a model and execution algorithms for forming a set of questions during online testing, which would differ from existing approaches and expand the capabilities of developers to create test platforms or individual test pages according to the requirements of customer organizations.

The work implements a mechanism for generating sets of questions and keywords that allow identifying answers to questions for the purpose of automatic processing of the answer to obtain working information before a personal meeting with the respondent.

Predictive assumptions about the development of the research object - the data obtained in the work can be used to quickly find the necessary help in relation to the needs of people who have been physically or psychologically affected by Russian aggression, which will be especially relevant during the liberation of Luhansk and Donetsk regions.

The field of application is a survey for the purpose of collecting information to support decision-making in situations that require preliminary preparation (medicine – admission of patients who are in remission, need rehabilitation, or doctors are sent for consultation based on signs of a stable condition); education and training – for the development of tests and situational problems that require non-standard solutions.

ARRAY OF WORDS, SET OF QUESTIONS, MODEL MATHEMATICAL MODEL OF THE TEST, MENTAL OPERATOR, STEP, ALGORITHM.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ ТЕХНОЛОГІЙ ІНТЕРАКТИВНИХ МЕТОДІВ ОПИТУВАННЯ ТА ВИВЧЕННЯ ДУМКИ КОРИСТУВАЧІВ В МЕРЕЖІ ІНТЕРНЕТ	12
1.1 Сучасні платформи для здійснення опитування користувачів та технології їх реалізації	12
1.2 Дослідження механізму здійснення он-лайн тестування	15
1.3 Формування множини питань на основі структурованого запиту до системи	20
1.4 Коротка характеристика об'єкта автоматизації та вимоги до створення системи он-лайн тестування	23
1.5 Висновок до першого розділу	25
2 МОДЕЛЬ ФОРМУВАННЯ МНОЖИНИ ПИТАНЬ ДЛЯ СТРУКТУРОВАНОГО ТА НЕРЕГЛАМЕНТОВАНОГО ТЕСТОВОГО ЗАВДАННЯ	27
2.1 Схема елементарних дій респондента при відповіді на завдання	27
2.2. Використання ментальних операторів для формування множини питань	29
2.3. Алгоритмізація процесу обробки відповіді для віднесення наступного питання до вибірки	34
2.4 Висновок до другого розділу	39
3 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ ФОРМУВАННЯ МНОЖИНИ ПИТАНЬ ПРИ ОН-ЛАЙН ТЕСТУВАННІ	41
3.1 Формування бази даних системи тестування	41
3.2 Методика та алгоритм формування множини питань	46
3.3 Експериментальна перевірка механізму формування множини питань для он-лайн тестування	52

3.4 Висновок до третього розділу	55
ВИСНОВКИ	56
ПЕРЕЛІК ДЖЕРЕЛ	58
ДОДАТКИ	63

ВСТУП

Актуальність теми. Зараз інструментами та сервісами, що широко представлені на Інтернет-платформах, легко можна створити будь-який тест. Наприклад, широко використовуються Google Форми, щоб створювати онлайн-форми й опитування з різними типами запитань. Або створення інтерактивних завдань від платформи «На урок». Також існують різноманітні конструктори тестів, які дозволяють за допомогою готових форм, окремих блоків та на основі готових прикладів швидко розробити будь-який тест.

Цікавий підхід до створення тестових завдань надає навчальна платформа Moodle – здійснення перебору питань, перемішування окремих тестових завдань таким чином, щоб неможливо було зафіксувати відповіді за номером відкриття питання у тесті, автоматичне нарахування балів за часткою правильної відповіді.

Однак все перелічене має недолік – типовість, «заточеність» під навчальні програми середніх або вищих освітніх закладів, обмеженість у можливостях редагування та оперативної зміни питань, обмеженість перебору та вибору за рівнем складності в залежності від відповіді респондента. Це ускладнює або унеможлиблює роботу в тому випадку, коли виникає необхідність створити нетиповий тест. Наприклад, тест, призначений для медичного центру, що займається реабілітацією поранених військових, коли від того, яка була відповідь на попереднє питання, залежить те, яким буде наступне питання. Такий складний тест необхідний, щоб до прибуття пацієнта чітко мати уяву, яких лікарів слід запросити на консультацію, зробити заявки на відповідні послуги, підготуватися до приймання пацієнта, створивши особливі умови перебування.

Тест, який би враховував зазначені вимоги, був гнучким при адаптуванні до умов застосування, а також використовував сучасні інструменти, які дозволяють запам'ятовувати попередні відповіді для вибору наступного питання, повинен при цьому мати не лише зручний інтерфейс користувача та

мінімальні складності для втілення розробки, а й відобразити механізм процесу мислення людини, щоб пов'язати отриману від респондента відповідь із записом до бази даних, де містяться ключові слова, за якими ідентифікується отримана відповідь. Як правило, подібні технології реалізації програм та сервісів тестування є малодослідженими, бо захищені комерційною таємницею та авторським правом. Враховуючи викладене, зазначена тема кваліфікаційної роботи є актуальною та має важливе практичне значення.

Алгоритмізацію процесів виконання тестових завдань та моделі створення опитувальників розглядалися багатьма українськими та закордонними дослідниками. Серед них можна назвати Морозова А.О., Бідюка П.І., Рябчуна Ю.В., Raskin J., Wimmer R. D. Та інших, які в своїх роботах були дотичні до теми дослідження.

Загальною проблемою зазначених досліджень є те, що безпосередньо реалізація механізму формування та перебору множин питань при створенні тестів є недостатньо визначеною, іноді такі дослідження неможливі до реалізації саме через примітивність у використанні шаблонів реалізації, а також недосконалість існуючих алгоритмів формування множини питань зі створеного масиву даних.

Мета і задачі дослідження. Мета роботи: розробити модель та алгоритми виконання для формування множини питань при он-лайн тестуванні, яка б відрізнялася від існуючих підходів та розширяла можливості розробників для створення тестових платформ або окремих сторінок тестування за вимогами організацій-замовників.

Задачі дослідження:

- 1) Проаналізувати існуючі технології інтерактивних методів опитування, наукові дослідження за зазначеною тематикою та провести огляд джерел за темою кваліфікаційної роботи.
- 2) Розробити модель інформаційної технології формування множини питань для структурованого та нерегламентованого тестового

завдання при он-лайн тестуванні та навести математичне обґрунтування.

- 3) Представити алгоритми виконання вибору питань до множини та їх програмну реалізацію.

Об'єкт дослідження: об'єктом дослідження є механізми вибору питань з множини, та формування нової множини з можливістю переборів та заміни.

Предмет дослідження: предметом дослідження є моделі та алгоритми перебору, вибору та формування множин питань при реалізації технології на веб-платформах.

Методи дослідження: загальнонаукові, математичного аналізу, математичного та імітаційного моделювання.

Наукова новизна одержаних результатів кваліфікаційної роботи полягає у тому, що удосконалено механізм формування множини питань завдяки логічному поєднанню механізму користувача в інтернеті при відповіді на тестове питання, та математичних основ технології GOMS із застосуванням логічних операторів для аналізу висловлювань респондента.

Практичне значення одержаних результатів. Розроблено інформаційну модель та виконано алгоритми вирішення задачі перебору та вибору окремих питань з формуванням множини завдань та автоматизованою перевіркою відповіді респондента для формування нових питань з метою поглиблення або тлумачення відповіді.

Апробація результатів магістерської роботи. Основні результати проведених досліджень були використані в роботі ГО «Спортивно-реабілітаційний центр ім. О. Кравця» (Додаток А)

Публікації. Kryazhych O., Itskovych V., Iushchenko K., Hrytsyshyna V., Bruvier D., Nykytyuk V., Bodnarchuk I. (2023) The use of abstract moore automaton to control the sensors of a service-oriented alarm and emergency notification network. Scientific Journal of TNTU (Tern.), vol 109, no 1, pp. 111–120. (Публікація в українському виданні, категорія Б) (Додаток Б).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку літератури з 54 найменувань та 4 додатків. Загальний обсяг кваліфікаційної роботи складається з 81 сторінки, з них 62 сторінки основного тексту, який містить 17 рисунків та 3 таблиці.

1 АНАЛІЗ ТЕХНОЛОГІЙ ІНТЕРАКТИВНИХ МЕТОДІВ ОПИТУВАННЯ ТА ВИВЧЕННЯ ДУМКИ КОРИСТУВАЧІВ В МЕРЕЖІ ІНТЕРНЕТ

1.1 Сучасні платформи для здійснення опитування користувачів та технології їх реалізації

Технології проведення опитування на базі Інтернет-платформ не є новою – це активно використовується в Інтернет-маркетингу [1]. Саме тоді були закладені основи розробки інтерактивних тестів, які, здебільшого, формувалися з окремих блоків, які органічно структурувалися за допомогою html та CSS, але практично не піддавалися коригуванню [2]. Тобто, під кожен тест формувався свій набір питань та модоивих відповідей.

На даний момент функціонал для організації он-лайн тестування значно розширився, а інструменти створення тестів дозволяють значно поглибити тестові форми та автоматизувати виконання завдань. Існує декілька базових популярних платформ для створення тестування в режимі он-лайн. Найвідомішим є Google Classroom [3] з можливістю використовувати форми для створення опитування з чітким встановленням дедлайну. Безпосередньо сервіс Google Форм [4] дозволяє задати необхідну кількість балів за виконання завдань з автоматичною перевіркою. Електронні ресурси Quizizz [5] дозволяють представити опитування більш наочно – у вигляді вікторин. Ресурс Surveymonkey [6] має доволі обмежений функціонал, призначений для швидкого створення невеликих опитувань. Безкоштовний сервіс Formative [7] дозволяє створювати оціночні завдання з готових блоків на кшталт аудіовідповіді чи текстового вікна. Один з інструментів, призначених для оцінювання студентів – Polleverywhere [8], дозволяє вбудовувати інтерактивні опитування безпосередньо під час дистанційного семінару, але має обмеження за кількістю учасників (не більше 25 осіб). Веб-сервіс Socrative

[9] має схожий з попереднім сервісом функціонал з обмеженням у 50 учасників. Платформа Wooclap [10] використовується вищими навчальними закладами і у безкоштовному варіанті дозволяє підключати до 1 тис. студентів. Дозволяє проводити інтерактивне опитування в режимі реального часу. Значно ширший функціонал у Moodle [11] – можливість створювати складні тести з переміщенням блоків завдань для кожного користувача та деталізація процесу оцінки окремої відповіді. Ресурс Flippity [12] має інструменти для створення різноманітного інтерактиву для навчання, тестування, перевірки знань чи звичайного опитування стосовно будь-якого питання. Online Test Pad [13] – багатофункціональний інструмент саме для здійснення тестування, проте все ж орієнтований на навчання. Відома платформа Kahoot! [14] зорієнтована на вищі навчальні заклади, хоча активно використовується і середніми закладами освіти. На платформі багато різноманітних інструментів, але функціонал для розробки тестів доволі обмежений та без можливості прямого редагування.

Варто зазначити, що всі розглянуті платформи мають один суттєвий недолік – створення тестів не передбачає механізму формування вибірки питань в залежності від результатів відповіді на менш складне питання. Тобто, респондент повинен пройти тест до кінця, а якщо слід поглибити сутність питання, щоб взяти якийсь аспект проблеми або розширити сутність відповіді, слід залучити респондента до іншого опитування. Цей недолік знаходиться в площині взаємодії машини і людини та є проблемою інтелектуалізації інформаційних систем [15].

В даний час значна частина сфер застосування інформаційних технологій пов'язана зі створенням та ефективним використанням людино-машинних систем, орієнтованих на аналіз (інтелектуальні інформаційно-аналітичні системи, проблемно-орієнтовані системи підтримки прийняття рішень, ситуаційні центри, глобальні інформаційні системи) [16]. Серед різноманітних завдань проектування людино-машинних систем важливу роль відіграють завдання побудови інтерактивних тестів та, зокрема, інтерактивних

інтерфейсів. Якщо перефразувати відомий вислів Дж. Раскіна [17 – 18] стосовно інтерфейсів обчислювальних систем до поставленого в роботі завдання, то користувач не думатиме про те, як створено тест, його цікавитиме, чи правильно він відповість на завдання, і чи зрозуміє, як це зробити.

Прогрес у розвитку інформаційних систем (ІС) передбачає покращення співвідношення ціна/продуктивність за законом Мура [19]. Проте у випадку створення тестових завдань цей закон прослідковується слабо [20], бо досі відсутня єдина класифікація проблем та завдань у сфері вдосконалення механізму формування питань та засобів створення тестів при он-лайн роботі. Чітко систематизовано лише покоління базових платформ взаємодії «людина-комп'ютер» (before-WIMP, WIMP – Windows, Icons, Menus, Pointing Device, post-WIMP) та загальні вимоги до інтерфейсів користувача [21].

Загальна мета інтелектуалізації при розробці тестів полягає у зменшенні обсягу спеціальних знань та вмінь користувача, що виходять за рамки його професійної орієнтації щодо засобів спілкування з системою, та зменшення трудомісткості операцій із введення даних та інтерпретації вихідних повідомлень [22]. Виходячи з такого визначення, можна виділити два напрями інтелектуалізації при формуванні механізму вибору питань для створення тесту:

- створення лінгвістичних засобів спілкування «комп'ютер-користувач» мовою, близькою до природної;
- вдосконалення технологічних засобів для діалогу користувача із системою (сценаріїв, меню, повідомлень про помилки, підказок тощо).

Перший напрямок, який можна визначити як генеральний шлях інтелектуалізації [15]. Реалізація другого напрямку властива при розробці інтерактивних систем різних рівнів (базової, типової, спеціальної) і полягає у створенні такого механізму опитування, щоб він не відрізнявся від подібного опитування, що проводить людина [22].

Взагалі питання оцінки рівня інтелектуальності комп'ютерних систем та інформаційних технологій і, зокрема, технологій формування тестових завдань, поки що залишаються мало дослідженою областю. Самі поняття та терміни «інтелектуальна», «інтелектуалізована», «інтелектуалізація» стосовно ІТ використовуються у великій кількості робіт з теорії та практики створення освітніх платформ та інформаційно-аналітичних систем різних класів та призначень. Проте єдиного тлумачення цих термінів у тих кількісних оцінок міри інтелектуальності (інтелектуалізованості) поки що склалося. Загальним недоліком ряду визначених визначень є використання термінів, пов'язаних з інтелектуалізацією, виключно в інтуїтивному, неопераційному розумінні.

Проте, при реалізації технології опитування користувачів через Інтернет-платформи [23 – 24], неодмінно виникає питання щодо взаємодії інструментів штучного інтелекту (AI) та ідеології IQ [23]. Це дозволяє узагальнити вимоги до втілення механізму автоматизованого формування множини питань з деякої вибірки, коли ІС порівнює відповідь респондента із загальним набором базових показників деякої області знань. При цьому ступінь зниження або підвищення інтелектуального навантаження, який виражатиметься через вибір наступного питання для респондента з підвищенням чи зниженням рівня складності, може бути використаний як критерій віртуальної інтелектуальності прикладної програмно-технічної системи, незалежно від її структури та внутрішніх властивостей.

1.2 Дослідження механізму здійснення он-лайн тестування

Найзручніше механізм здійснення он-лайн тестування відслідкувати на прикладі розробленого тексту засобами Moodle [11]. В цій системі є можливість створювати як тести у класичному розумінні, де здійснюється вибір правильної відповіді із заданих варіантів, так і опитувальників, де треба зазначити свою відповідь (наприклад, для проведення опитування щодо підвищення якості освіти).

Варто зазначити, що розвиток інформаційних технологій призвело до виникнення нових форм організації он-лайн тестування [25]. Ці форми націлені не лише на визначення рівня знань чи вмінь за будь-яким напрямком освіти, а й на дослідження певних проблемних ситуацій чи моніторингу процесів [26]. А використовувати опитування або вирішення ситуаційної задачі у вигляді проблемного кейсу з метою збирання інформації та оцінці ситуації було запропоновано у роботі [27] із поданням схеми розробки бізнес-кейсу для пошуку відповіді задане проблемне питання. При цьому, як впливає з роботи [26], з використанням інформаційних технологій реалізуються такі підходи до механізму формування опитувальників:

- за участю експерта, який перевіряє відповіді, тому що відповідь відкрита чи неоднозначна для отримання критичної інформації чи оцінки ситуації;

- автоматизовані, оскільки відповідь однозначна і її можна вибрати зі списку варіантів для отримання конкретної інформації шляхом вичленування найбільш точної інформації з масиву даних;

- симулятори, які дозволяють переходити від завдання завдання з метою актуалізувати отриману інформацію.

Сучасні розробники інформаційних технологій [28] зазначають, що автоматизація подібних механізмів перебуває у стадії розробки. Відкритим питанням є виключення участі людини при аналізі відповідей та опрацювання отриманої інформації. Існуючі підходи та технології розробки інтерактивних методів опитування [29] у існуючих публікаціях представлені фрагментарно, в основному, у вигляді теоретичних викладок та загальних інформаційних моделей, як у роботі [26].

В процесі реалізації механізму здійснення он-лайн тестування можна використовувати підхід [30], але з модернізацією – урахуванням типової схеми поведінки користувача під час роботи з веб-ресурсами [31]. Зокрема, необхідно наголосити на використанні логіко-семантичного апарату досліджуваного процесу [32], розробити логічну формулу, за якою кейс

дозволить зробити висновок про стан питання дослідження, а також алгоритм, за яким можна співвіднести отриману відповідь з поставленим завданням [28]. Враховуючи викладене, узагальнена схема механізму реалізації інтерактивного опитування при он-лайн тестуванні може бути представлена в такий спосіб (рис. 1.1).

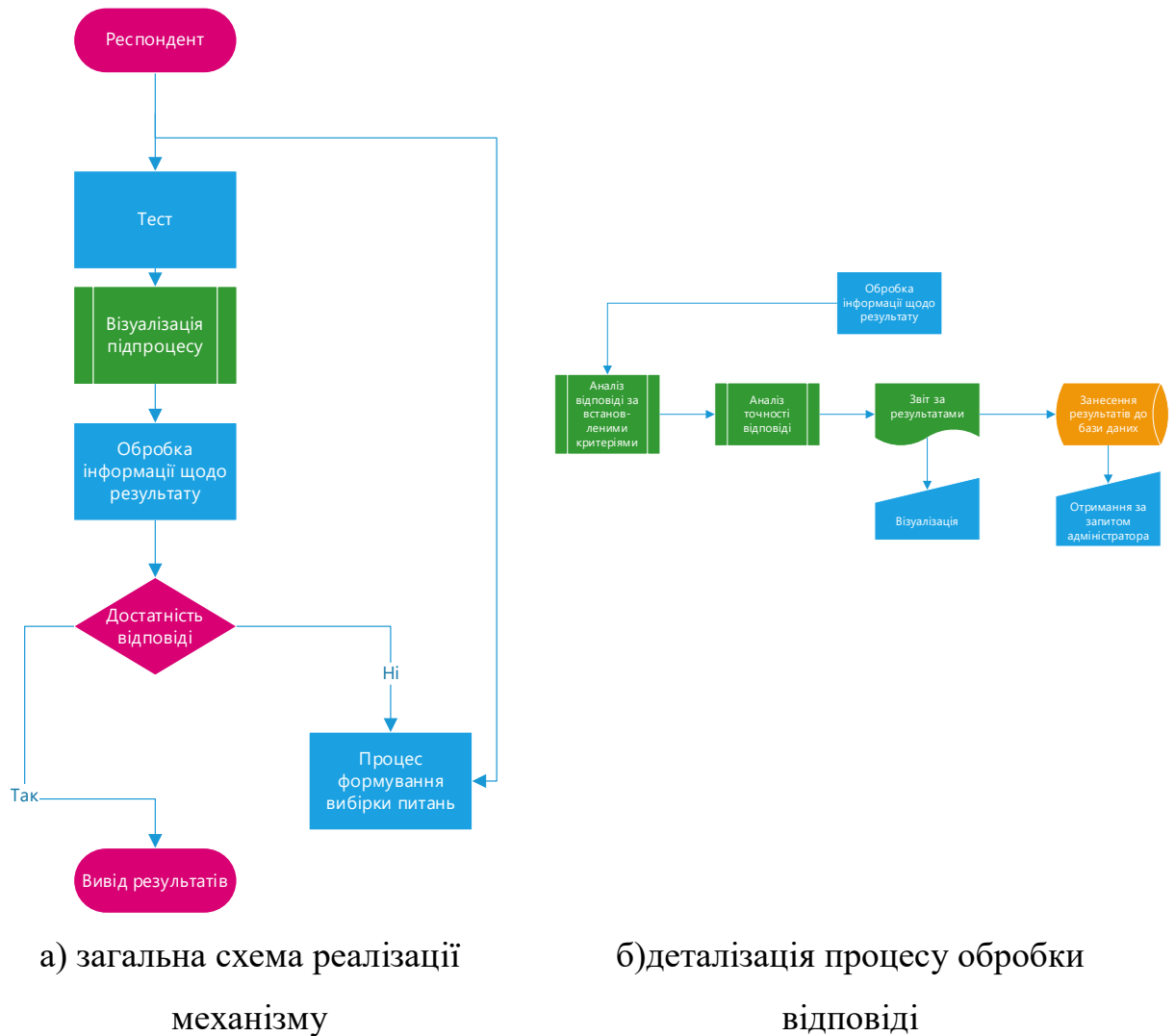


Рисунок 1.1 – Узагальнена схема механізму реалізації інтерактивного опитування

У наведеній схемі (рис. 1.1 – а) введено обмеження – опущено механізм ідентифікації користувачів. Це стандартна процедура, що передбачає реєстрацію користувача та/або ідентифікацію за IP адресою. У блоці

деталізації процесу обробки відповіді (рис. 1.1 – б) за окремими параметрами відбувається обробка даних для отримання чи заперечення відповідності відповіді поставленому питанню.

Механізм реалізації тесту традиційно можливий з підказкою та без підказки [33]. Перший варіант полягає в тому, що в процесі надання відповідей за запитанням можна використовувати вибір відповідей з переліку або вільну відповідь, тобто, в останньому випадку, ідентифікація відповідності відповіді може бути на основі еталонного слова або поєднань окремих символів. Якщо кожному слову-еталон відповідає певний елемент (або елементи) і для наведеного зразка існує відповідне слово-еталон, то процес завершується. В іншому випадку відповідь не приймається і тестування/відбір вважається не пройденим або зараховується негативний результат і з'являється нове завдання для визначення рівня компетентностей.

Варіант реалізації подібної системи з підказкою типовий [27], характерний більшості інформаційних систем, починаючи від пошукових систем Інтернету і закінчуючи звичайними тестами на платформах, типу [10 – 11, 14], з переліком можливих відповідей. Безпосереднє визначення того, наскільки відповідь близька до еталонної, можлива за допомогою реалізації чотирьох алгоритмів корекції з ймовірностями наступних подій на основі обробки введених термінів або символів:

- помилку знайдено, ідентифіковано, визначено відсоток правильної відповіді на заданому відрізьку оцінювання;
- помилку виявлено, але ідентифікувати її не вдається, тому відкривається уточнююче питання;
- помилка виявлена, але вона ідентифікована не так, тому відкривається блок аналогічних питань для уточнення знань на тему;
- помилка не виявлена, відповідь точна і їй присвоюється встановлений бал.

Особливістю реалізації механізму інтерактивного опитування є алгоритм, який дозволяє за ключовим словом побудувати логічну схему вибору пакета слів (масиву інформації) з бази даних тесту (тестів), на основі яких проводиться порівняння відповіді. Чим більше слів з масиву інформації використано респондентом, тим вищий бал проходження тесту, що у результаті характеризує предмет дослідження тесту. Підсумок тесту – автоматичне підбиття результатів, де бали присвоюються «1» за відповідністю та «0» за невідповідністю слова відповіді словам масивів.

За такого підходу виключається вплив людського чинника, оскільки аналіз результатів реалізується з допомогою типових алгоритмів порівняння [34]. Ці алгоритми можуть виконуватись послідовно. Наприклад, приймається, що деяка правильна відповідь описується набором з m слів $A_j=(a_1, \dots, a_i, \dots, a_m)$, де $(j=1, \dots, N)$, відсутність одного або кількох слів відповіді в масивах порівняння сприймається як помилка і описується $B=(b_1, \dots, b_k, \dots, b_n)$. У цьому випадку проводиться порівняння $P_A(A_j)=B$ і за результатами кейсу система видає звіт з висновком «швидше так, ніж ні» або «швидше ні, ніж так» на основі набраної кількості балів. Подібний аналітичний алгоритм пошуку відповідності може бути реалізований шляхом безпосереднього перебору з великою кількістю різнорідних величин на основі комбінаторного аналізу [35] відповідно до заданих правил і підходів логічного програмування [36]. Останнє, саме формування формули тесту з урахуванням логічного висловлювання, є базисом розробки механізму реалізації теснування. Формула тесту відображає певний структурований запит до системи стосовно формування множини питань для визначення певних аспектів для здійснення аналізу проблеми.

1.3 Формування множини питань на основі структурованого запиту до системи

Якщо взяти за основу для розробки системи он-лайн тестування архітектуру, близьку до архітектури сучасних інформаційно-аналітичних систем (ІАС), то можна визначити чотири наступних рівні [37]:

- транзакційні або операційні бази даних (БД), що є складовою частиною OLTP-систем (on-line transactional processing). Містять дані про результати повсякденної роботи системи;

- ETL – інструменти (extraction, transformation, loading), призначені для отримання даних з транзакційних БД, їх «очищення», консолідації та подальшого завантаження в цільові аналітичні бази даних наступних рівнів формування множин питань;

- сховища даних (Data Warehouse), призначені для організованого зберігання консолідованих даних, орієнтованих на аналіз відповіді респондента;

- вітрини даних (Data Marts), призначені щодо цільового аналізу складних відповідей. Будуються, як правило, на основі сховища даних, а у разі його відсутності (у простих ІАС) – безпосередньо з БД OLTP – систем. Багатовимірні вітрини організуються як багатомірних БД OLAP, а реляційні – як реляційних БД зі схемою «зірка». На відміну від сховища даних, вітрини «обслуговують» рівень не вище за окремий підрозділ ІАС, а іноді можуть створюватися і для окремого користувача-замовника [38].

З погляду користувача-замовника сервісу для он-лайн тестування, саме вітрини є безпосереднім інформаційним ресурсом, що використовується для реалізації складних тестових завдань. Це можна назвати тематичною базою даних (ТБД) он-лайн сервісу для тестування.

Залежно від складності формулювання запиту до ТБД із боку користувача можуть бути наступні загальні підходи для створення тестів [39]:

а) регламентовані тести (РЗ), створені заздалегідь. Їх опис та SQL-код збережено у системі реалізації запитів системи тестування. Функціональність (різноманітність можливих варіантів наборів даних) цих запитів мінімальна, а оперативність доступу – максимальна. Від користувача для створення тесту потрібно лише знайти потрібний регламент у запропонованому переліку та дати команду на виконання (рис. 1.2);

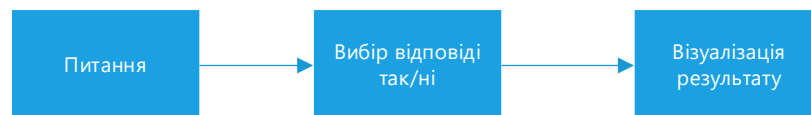


Рисунок 1.2 – Регламентований тест

б) параметричні регламентовані тести (ПРТ) теж передбачають використання регламентованого шаблону, але дозволяють специфікувати деякі параметри відповіді, що розширює функціональність при незначному погіршенні оперативності (рис. 1.3);

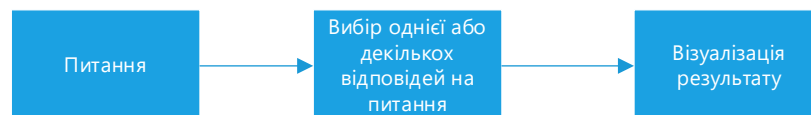


Рисунок 1.3 – Параметричний регламентований тест

в) довільні нерегламентовані тести (ДНТ) характеризуються максимальною функціональністю, оскільки відкривають можливість отримати глибоку і змістовну відповідь. Але мають мінімальну оперативність та потребують участі оператора або спеціальних програм обробки інформації для проведення оцінювання відповіді (рис. 1.4).



Рисунок 1.4 – Довільний нерегламентований тест

Підхід, наведений на схемі 1.4 найбільш довгий і витратний за працею виконання. І важкість тут полягає у створенні і організації такої БД, яка дозволяє формувати множини питань, за якими формуються підмножини ключових слів, які стають основою для автоматизованої перевірки відповіді користувача.

Існує ряд стандартних програм різних фірм полегшення побудови баз даних. Наприклад, Query Builder фірми Oracle, Query Builder для СУБД SQL-сервер, Query Builder фірми Borland, Microsoft Query фірми Microsoft, Microsoft Access та інші. Ці програми надають графічний інтерфейс для створення програмного забезпечення на мові SQL. До бази формується запит, який дозволяє створити вибірку питань – певну множину питань, що дозволить описати тему, що досліджується. Побудовальник запитів має область, яка відображає запит у графічному вигляді, та область, яка містить текст SQL-запиту. При цьому подібні запити потрібно своєчасно оновлювати, додавати записи до БД згідно внесених оновлень, видаляти застарілі записи.

Для створення запиту до БД, який допоможе сформувати множину питань для тестування, необхідно [40]:

- працювати безпосередньо з таблицями реляційної моделі БД;

- розуміти та знати реляційну модель БД;
- знати всі таблиці реляційної моделі та їх атрибути;
- виконувати з'єднання 2-х та більше таблиць за ключами;
- знати ключі та правила з'єднання таблиць (яку таблицю з якою або з якими з'єднувати);
- знати правила, які можна накладати на атрибути (наприклад, до атрибуту типу «текст» не можна застосувати функцію суми тощо);
- знати аналітичні функції, які використовуються у запиті SQL.

Окрім стандартних запитів, на даний час на ринку інформаційних технологій існує багато видів готових аналітичних систем, які мають у своєму складі засоби формування нерегламентованих запитів. Це такі системи, як Oracle Discoverer, Business Objects, Hummingbird та інші [41]. Однак, як правило, всі вони досить складні, дороги, зазвичай орієнтовані на торгівлю, вимагають від користувача відповідної кваліфікації, а найчастіше і змісту додаткових фахівців. Такі системи потребують додаткового адміністрування, оскільки поряд із адмініструванням БД необхідно адмініструвати також програмний продукт.

1.4 Коротка характеристика об'єкта автоматизації та вимоги до створення системи он-лайн тестування

Поставлена в роботі задача вирішується відповідно до вимог потенційного замовника – «Спортивно-реабілітаційного центру інвалідів та ветеранів війни» Закарпатського обласного осередку Української федерації спорту інвалідів з ураженням опорно-рухового апарату.

Мета «Спортивно-реабілітаційного центру» – найбільш повне відновлення втрачених функцій організму та виведення із хворобливого стану. Якщо це неможливо-ставиться мета часткового відновлення або компенсації порушеної або втраченої функції, відновлення працездатності, або проведення перекваліфікації в реабілітаційному центрі. Саме для цього планується

проводити декілька рівнів тестування та опитування потенційних пацієнтів та пацієнтів, що вже пройшли реабілітацію для контролю їх стану. Жодний наявний ресурс на Інтернет-платформах не відповідає вимогам створення багатофункціонального тесту.

Етапи реабілітації, на яких повинен здійснюватися динамічний контроль за процесами шляхом тестування пацієнтів, наведений на рис. 1.5.

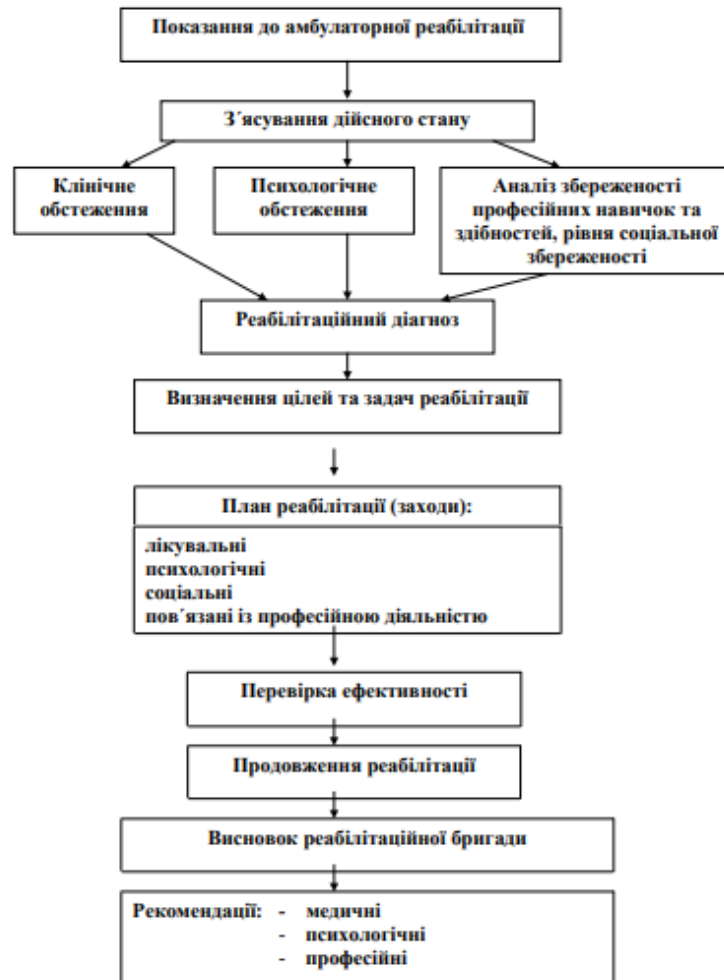


Рисунок 1.5 – Схема контролю за процесами реабілітації шляхом тестування

Планується, що даний «Спортивно-реабілітаційний центр» буде охоплювати всі можливі види реабілітації мультидисциплінарною командою, шляхом комбінування видів реабілітації результат для пацієнта буде досягнутий швидше ніж при використанні одного або декількох видів. Позитивний результат для пацієнта буде досягнутий за допомогою

кваліфікованих лікарів: реабілітологів, лікарів ФРМ, лікарів ерготерапевтів, психологів, психіатрів, лікарі ЛФК, лікарів первинної та вторинної допомоги, логопедів. На кожному з етапів (рис. 1.5) відбуватиметься тестування пацієнтів та внесення отриманих результатів до медичної картки. Тому є потреба створювати гнучкі тести, які можна налаштувати в залежності від того, якою була попередня відповідь. Подібну задачу можна вирішити шляхом реалізації механізму формування множини питань з БД, а реалізувати в он-лайн режимі на Інтернет-платформі реабілітаційного закладу.

1.5 Висновок до першого розділу

В першому розділі кваліфікаційної роботи даних та огляду наукових джерел з теми дослідження можна зробити наступні основні висновки та узагальнення:

1) більшість систем, які можна використати при розробці багатофункціонального он-лайн тестування, потребують спеціальних висококваліфікованих знань, а також реалізація багатьох інструментів в режимі он-лайн поки що недоступна. Для тієї частини користувачів, які здатні працювати з подібними системами, вони занадто високовартісні;

2) відповідно, від респондента, та, власне, і користувача системи тестування не повинно вимагатися розуміння предметної області ДНТ і знання термінів, якими описується дана предметна область, знання теорії реляційних баз даних, знання структури ДНТ, знання мови SQL;

3) основою створюваного механізму формування множини питань для проведення он-лайн тестування має бути попередній опис семантики ДНТ природною мовою за допомогою стандартного CASE-засобу, однією з основних переваг якого є можливість створення концептуальної (логічної) моделі реляційної бази даних;

4) структурованим запитом до БД може виступати спочатку вибір питань, які допоможуть виявити початкову думку чи знання респондента, а

далі, на основі першої відповіді або відповідей, яка вже виступить як структурований запит, формується множина питань для чіткого визначення думки, знань, вмінь і т. інш. респондента;

5) вимоги користувача (замовника) системи он-лайн тестування доволі специфічні (реабілітація людей, що постраждали від війни), тому їх неможливо вирішити за допомогою стандартних інструментів, представлених на платформах створення тестів. За вимогами тестування буде проводитися на низці етапів реабілітаційної роботи, тому механізм формування питань слід розробити таким чином, щоб результати охоплювали весь період роботи з пацієнтами, дозволяли легко вносити зміни та автоматично надавати вибірки питань в залежності від попередньої відповіді респондента, що можна вирішити за допомогою застосування механізму структурованого тестового завдання та довільного нерегламентованого тесту.

2 МОДЕЛЬ ФОРМУВАННЯ МНОЖИНИ ПИТАНЬ ДЛЯ СТРУКТУРОВАНОГО ТА НЕРЕГЛАМЕНТОВАНОГО ТЕСТОВОГО ЗАВДАННЯ

2.1 Схема елементарних дій респондента при відповіді на завдання

Щоб реалізувати механізм формування вибірки тестових завдань, слід проаналізувати типову схему поведінки респондента [42 – 44] при відповіді на тестові завдання в режимі он-лайн. Для цього можна розглянути елементарні дії при вводі користувачем відповіді на запитання (рис. 2.1).

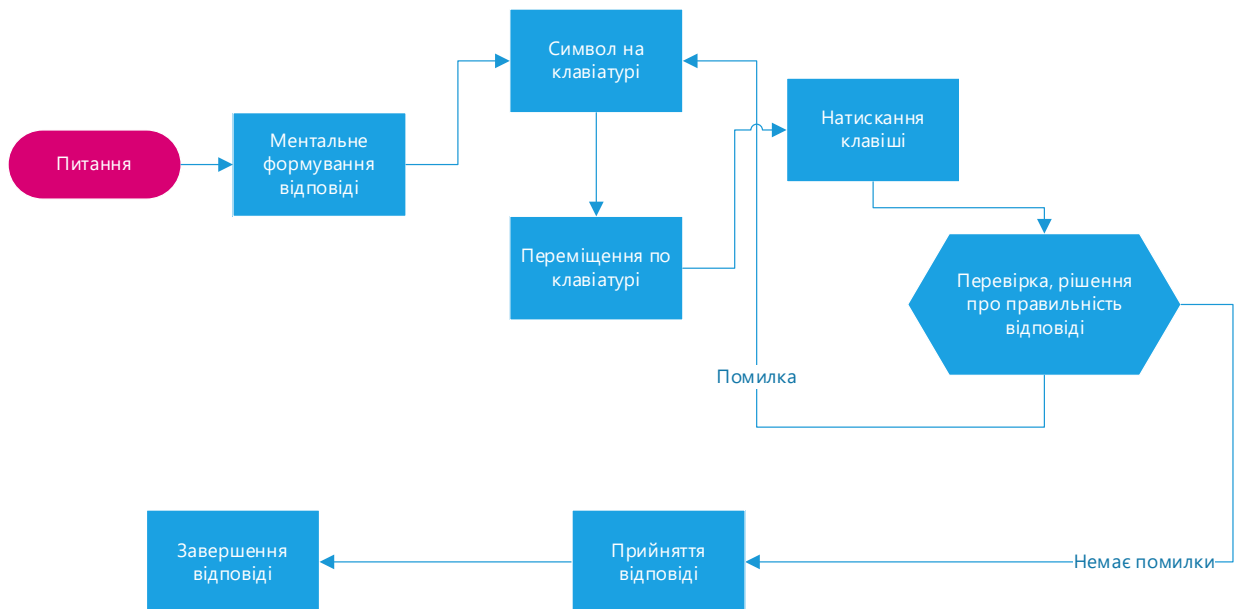


Рисунок 2.1 – Схема елементарних дій користувача при вводі слова-відповіді

Наведена схема (рис. 2.1) є ідеалізованою. Початковим кроком користувача виступає процес зрозуміння відповіді і прийняття рішення про введення слова. Останній етап – перевірка відповіді і, якщо помилки немає, прийняття відповіді, того, натискання клавіші введення у систему. Основна дія респондента при виконанні тесту – введення якогось слова або речення, що складається з n -символів.

Відповідно, можна вирахувати трудомісткість праці респондента над тестом за допомогою моделі за технологією GOMS [45]. Використовуючи цю технологію та схему (рис. 2.1) можна побудувати модель он-лайн тестування.

Враховуючи, що за технологією GOMS для кількостної оцінки витрат праці H виконання дій етапів за схемою (рис. 2.1) пропонується лише два стандартних оператори: M (1,2 сек.) та K (стандарт 0,28 сек.). Тобто, для оцінки H виконання завдання за тестом довжиною n літер пропонується формула:

$$H = n(K + xM) + yM, \quad (2.1)$$

де $x, y = \{0,1\}$.

Добір значень x, y може виступати рішенням тесту з розташування деяких ментальних операторів M , які можуть виступити точками переходів між питаннями не лише за фактом відповіді на попереднє питання, а й за фактом вірної чи невірної відповіді (істина/хиба). Практичне значення полягає у наступних варіантах рішення, наприклад: $x = 0, y = 1$; $x = 1, y = 0$; $x = 2, y = 0$. При цьому можна відійти від методології GOMS до визначення x, y і використати інший підхід до розстановки оператора M з варіантом пропозицій та рішень.

Сутність полягає у тому, що при розробці, наприклад, нерегламентованого тесту, не можна заздалегідь знати, скільки символів/слів \bar{v} ($\bar{v} \leq n$) введе користувач для відповіді. Звичайно, вікно вводу відповіді можна обмежити за кількістю символів. За часом T_1, T_2, T_3 для різних варіантів тестів – РЗ, ПРТ, ДНТ відповідно – відповіді будуть відповідати наступним математичним моделям:

$$T_1 = K\bar{v} + M + P + BV, \quad (2.2)$$

$$T_2 = (K + M)\bar{v} + P + BB, \quad (2.3)$$

$$T_3 = (K + M + M)\bar{v} + P + BB, \quad (2.4)$$

де дії користувача приблизно наступні за часом:

$K = 0,28$ сек – натискає клавішу;

$P = 1,1$ сек – клік мишкою;

$BB = 0,2$ сек – подвійний клік.

Наведені значення прийняті як ті, що рекомендуються для розрахунків дій користувача [46].

Значення T_1 розраховується з того, що користувач пише одне слово або робить позначку біля правильної відповіді. T_2 – вибір декількох відповідей або введення обмеженої кількості слів, T_3 – пишеться розширена відповідь. В цілому через схему елементарних дій користувача можна визначити, наскільки користувачу зручно працювати з розробленим тестовим завданням або опитувальним аркушем.

2.2. Використання ментальних операторів для формування множини питань

Під ментальним оператором в цій роботі називається поступове наближення наступного питання респонденту у відповідності до ключового слова попередньої відповіді тестового завдання або опитування [47 – 48]. Фактично – це рух за точками [49 – 50], які є поворотними для створення нового запиту на вибір питання з множини або формування нової множини для проведення уточненого опитування. Подібний механізм був апробований за допомогою автомату Мура, про що опубліковано в роботі [51].

Однак для формування множини питань не завжди достатньо алгоритму простого відбору з діями повторення чи однотипного вибору. Іноді треба

сформувати складну виборку, що охоплює змістовне коло питань. Тому було вирішено використати для мети формування множини питань для тестування ментальні оператори, оскільки їх набір буде залежати саме від проблемної області. Обґрунтованість цього шляху визначається трьома основними факторами:

- ментальна підготовка до наступного кроку до пропозиції запитання від системи тестування залежить від дій (відповіді) респондента;

- оператори K , $T(n)$ та P теж містять у собі якусь ментальну складову, інтелектуально навантажує користувача, тобто. вони також є частково ментальними;

- спеціалізація моделі тесту для конкретних класів завдань звужує області можливих значень операторів та дозволяє розкласти сукупність дій на простіші та зрозумілі кроки.

Схема декомпозиції операторів M , K наведена на рис. 2.2. Позначення для мікрооператорів $\mu_1 \div \mu_5, k$ наступні:

- 1 – читання тексту (μ_1) з первинного носія та його осмислення (запам'ятовування), сек/символ;

- 2 – пошук символів на клавіатурі для відповіді (μ_2) та затримка руки у позицію «над символом» / визначення системою за введеними ключовими словами або частинами слів істини чи хибі відповіді, сек/символ;

- 3 – візуальний аналіз тестового запитання (μ_3) та прийняття рішення про подальші дії (зокрема, про наявність чи відсутність помилки при наданні відповіді), сек/символ;

- 4 – відправлення відповіді on-line (μ_4) / формування системою вибірки питань, сек/символ;

- 5 – μ_5 , затримка для осмислення нового питання / обробка системою нового питання, що йде до вибірки, сек/символ;

- k – набирання нової відповіді, сек/символ.

За своєю суттю оператори μ_1, μ_3, μ_5 – чисто ментальні, k – оператор руху, μ_2 та μ_4 – комплзиція ментальних дій і рухів. Фактично ці оператори є оберненими значеннями для моделі тесту – у ті моменти, коли респондент замислюється над відповіддю, система обробляє попереднє питання та вибирає нове питання згідно відповіді, додає його до вибірки, вилучаючи те питання, на яке відповіді користувач не зможе дати з об'єктивних причин.

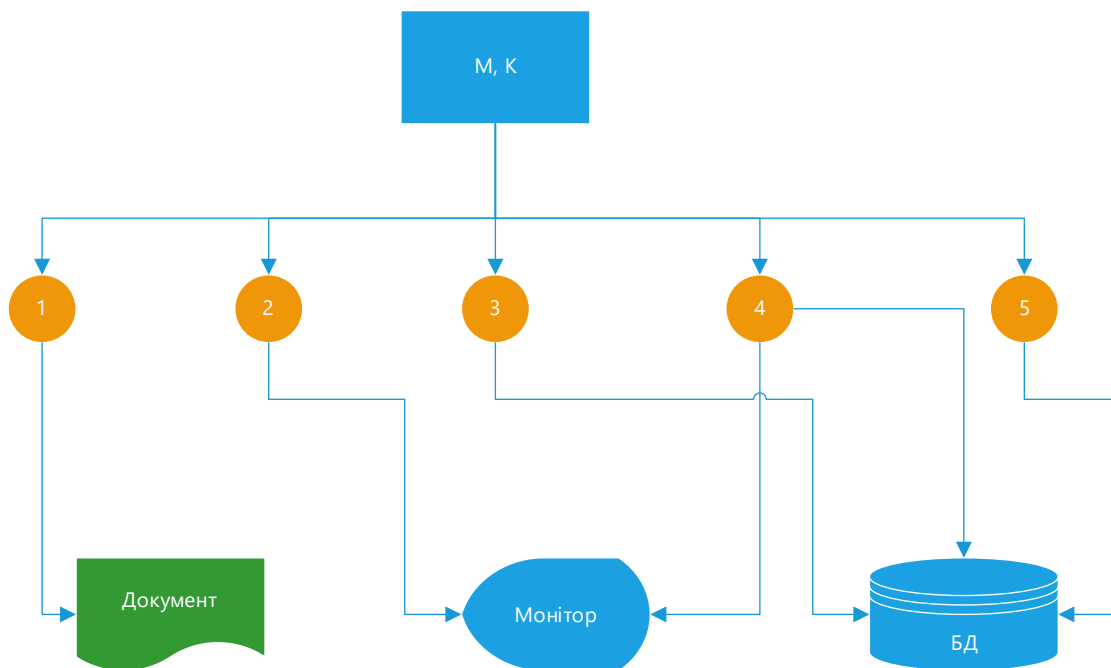


Рисунок 2.2 – Схема декомпозиції ментальних операторів – зворотня схема тесту

Для отримання експериментальних оцінок значень μ, k істотні такі чинники:

- характер інформації, що вводиться;
- тайп-кваліфікація користувача (рівень володіння комп'ютерною технікою).

З точки зору першого фактора, що враховує вплив вихідного знання користувача про словник допустимих символів і слів, виділимо 3 типи інформації, що вводиться:

1) цифрові та алфавітно-цифрові коди. Тут знання користувача про словник допустимих слів відсутнє практично повністю та можливості запам'ятовування повного слова при введенні – мінімальні;

2) мнемотекст (текст природною мовою, рідною мовою користувача). Тут можливості запам'ятовування слова та знання словника допустимих слів – максимальні;

3) кодотекст (текст природною мовою, але користувач термінами не володіє) У цьому випадку мнемонічні можливості користувача займають деяке проміжне положення.

З погляду другого чинника, прийнято, що типовий користувач комп'ютерної техніки немає спеціальної підготовки, отже його кваліфікація невисока. Тому для тестування системи респонденти-учасники експериментів вибиралися, виходячи з цього припущення про їхню кваліфікацію.

Значення $\mu_1 - \mu_5$ можуть бути визначені на підставі даних вимірів, отриманих у результаті наступних комп'ютерних експериментів, які проводилися з використанням алгоритмів, реалізованих у VBA [52]. Зокрема:

а) реалізований алгоритм генерації випадкового вибору питання з масиву для формування вибірки тестування респондента. Особливістю реалізації є розбиття масиву на інтервали за складністю питання та виір питання з цього інтервалу. Фрагмент коду наведено на рис. 2.3.

```
Sub Генератор_()
''
Randomize
'**** Початок циклу розрахунків ****
For k = 1 To 2000
'****Генерація випадкового номеру питання
a_rand = Rnd
'**** Вибір інтервалу за допомогою селектору***
Select Case a_rand
Case 0 To 0.3065
```

Рисунок 2.3 – Фрагмент коду реалізації алгоритму вибору випадкового питання з масиву

Після цього генерується випадковий номер питання всередині інтервалу та відбувається округлення розрахункового номеру до цілого числа;

б) вибірки питань формуються за окремими каналами. Якщо при обробці першого питання виявляється хиба, то запускатиметься для відповіді нова вибірка питань (другий канал), де відповідь на друге питання визначає, яким шляхом далі відповідатиме респондент. Фрагмент коду виконання наведено на рис. 2.4.

```
If canal(1) = 0 Then
i = 1
GoTo Test2
End If
If canal(2) = 0 Then
i = 2
GoTo Test3
End If
If canal(3) = 0 Then
i = 3
GoTo Test1
```

Рисунок 2.4 – Фрагмент коду реалізації алгоритму для перебору питань

Особливістю реалізації цього алгоритму є збереження вмісту каналів у кожний такт та додавання чи віднімання питань, які не потребують або потребують поглиблення відповіді.

В підсумку експериментів проводилися заміри часу відгуку системи при введенні правильної відповіді, завідомо невірної відповіді з очікуванням виводу номеру іншого питання, а також часу обробки питання. Підсумок експерименту дозволив зробити висновок, що механізм перебору та вибору питання працює, однак потребує більш досконалих інструментів реалізації. Також необхідно чітко прописати механізм обробки відповіді, яка дозволяє вибрати саме те питання, яке потрібне для уточнення або поглиблення відповіді на попереднє питання респондента.

2.3. Алгоритмізація процесу обробки відповіді для віднесення наступного питання до вибірки

Обробка відповіді системою залежить від того, чи ввів користувач ключове слово, за яким визначається «приналежність» його відповіді до правильної. Користувач може ввести це слово відразу, а може наприкінці відповіді. Тобто, виникає деяка невідома величина t , що може бути виміряна m при $\Pi_j (j=1, m)$. Процес вимірювань та його результати схильні до випадкових помилок. У підсумку можна отримати пакет вимірювань виконання тестового завдання респондентом, що складається з деякої кількості N_j значень τ_{ij} величини $(i=1, N_j)$.

Метою та результатом обробки значень τ_{ij} є визначення математичного очікування $\tau(m) \approx t(m)$ та середньоквадратичного відхилення $\sigma(m)$ проходження тесту (рис. 2.5).

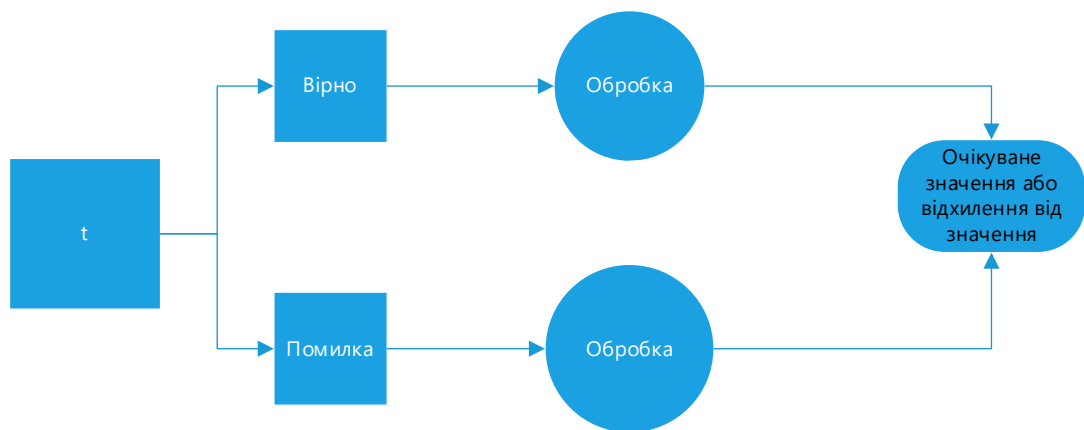


Рисунок 2.5 – Схема оцінки проходження тесту респондентом

Величина $\tau(m)$ є очікуваним значенням вимірювання величини t , а $\sigma(m)$ визначає очікуване відхилення від значення $\tau(m)$.

З визначення математичного очікування випадкової величини τ_{ij} випливає

$$t(m) \approx \tau(m) = \frac{\sum_{j=1}^m \sum_{i=1}^{N_j} \tau_{ij}}{\sum_{j=1}^m N_j}. \quad (2.5)$$

Проте для алгоритмізації наведена модель (2.5) є незручною, оскільки уточнення значення $t(m, m+1)$ проходження тесту за кожним новим питанням потребує всіх значень τ_{ij} для $i = 1, N_{m+1}$, а значення $t(m, m+1)$ повинно перераховуватися повністю не один раз. Тому для алгоритмізації процесу краще звести обчислення $t(m, m+1)$ до додавання визначення ймовірності вірної відповіді до значення $t(m)$. Тоді алгоритм формування множини питань тесту можна представити схемою 2.6.

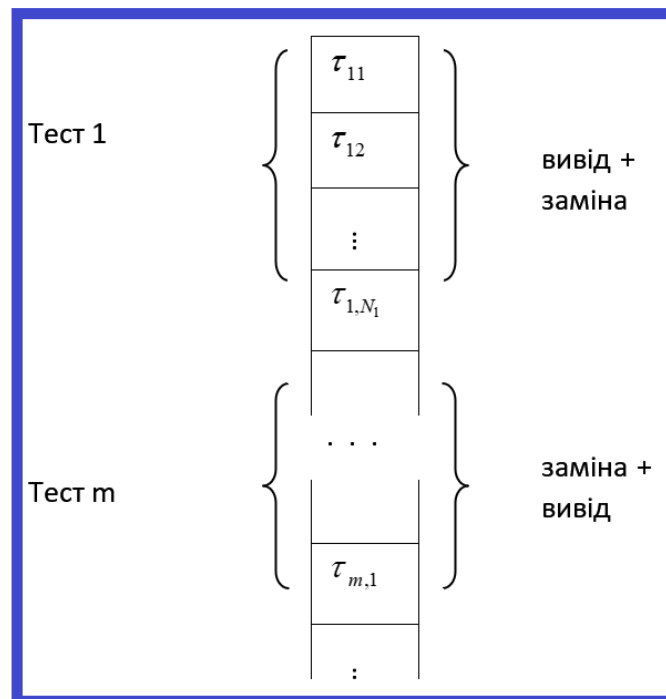


Рисунок 2.6 – Схема формування множини питань

Можна ввести поняття ваги обчислень для визначення того, яке питання буде наступним додане у вибірку. Зазначене можна обчислити за допомогою формул:

$$w(m) = \frac{\sum_{j=1}^m N_j}{\sum_{j=1}^{m+1} N_j},$$

$$w_{m+1} = \frac{N_{m+1}}{\sum_{j=1}^{m+1} N_j},$$

де $w(m)$ – означення процесу перебору питань у базі даних;

w_{m+1} – прийняття вибору питання Π_{m+1} .

У цьому випадку $w(m) + w_{m+1} = 1$.

Алгоритм обчислень може бути побудований за наступною моделлю:

$$t(m, m+1) = t(m) \cdot w(m) + t_{m+1} \cdot w_{m+1}. \quad (2.6)$$

Коректність моделі (2.6) можна довести за допомогою наступних виразів, приймаючи, що $j = 1, \dots, m, m+1$.

$$\begin{aligned} t(m, m+1) &= t(m) \cdot \frac{\sum_{j=1}^m N_j}{\sum_{j=1}^{m+1} N_j} + t_{m+1} \cdot \frac{N_{m+1}}{\sum_{j=1}^{m+1} N_j} = \\ &= \frac{\sum_{j=1}^m \sum_{i=1}^{N_j} \tau_{ij}}{\sum_{j=1}^m N_j} \cdot \frac{\sum_{j=1}^m N_j}{\sum_{j=1}^{m+1} N_j} + \frac{\sum_{i=1}^{N_{m+1}} \tau_{ij}}{N_{m+1}} \cdot \frac{N_{m+1}}{\sum_{j=1}^{m+1} N_j} = \\ &= \frac{\sum_{j=1}^m \sum_{i=1}^{N_j} \tau_{ij} + \sum_{i=1}^{N_{m+1}} \tau_{ij}}{\sum_{j=1}^{m+1} N_j} = \frac{\sum_{j=1}^{m+1} \sum_{i=1}^{N_j} \tau_{ij}}{\sum_{j=1}^{m+1} N_j}. \end{aligned}$$

Якщо зазначену модель (2.6) ввести у схему оцінки проходження тесту респондентом (рис. 2.5), то можна візуалізувати алгоритм перебору питань та формування множини питань для тесту зі змінними питаннями в залежності від відповідей респондента, як представлено на блок-схемі (рис. 2.7).

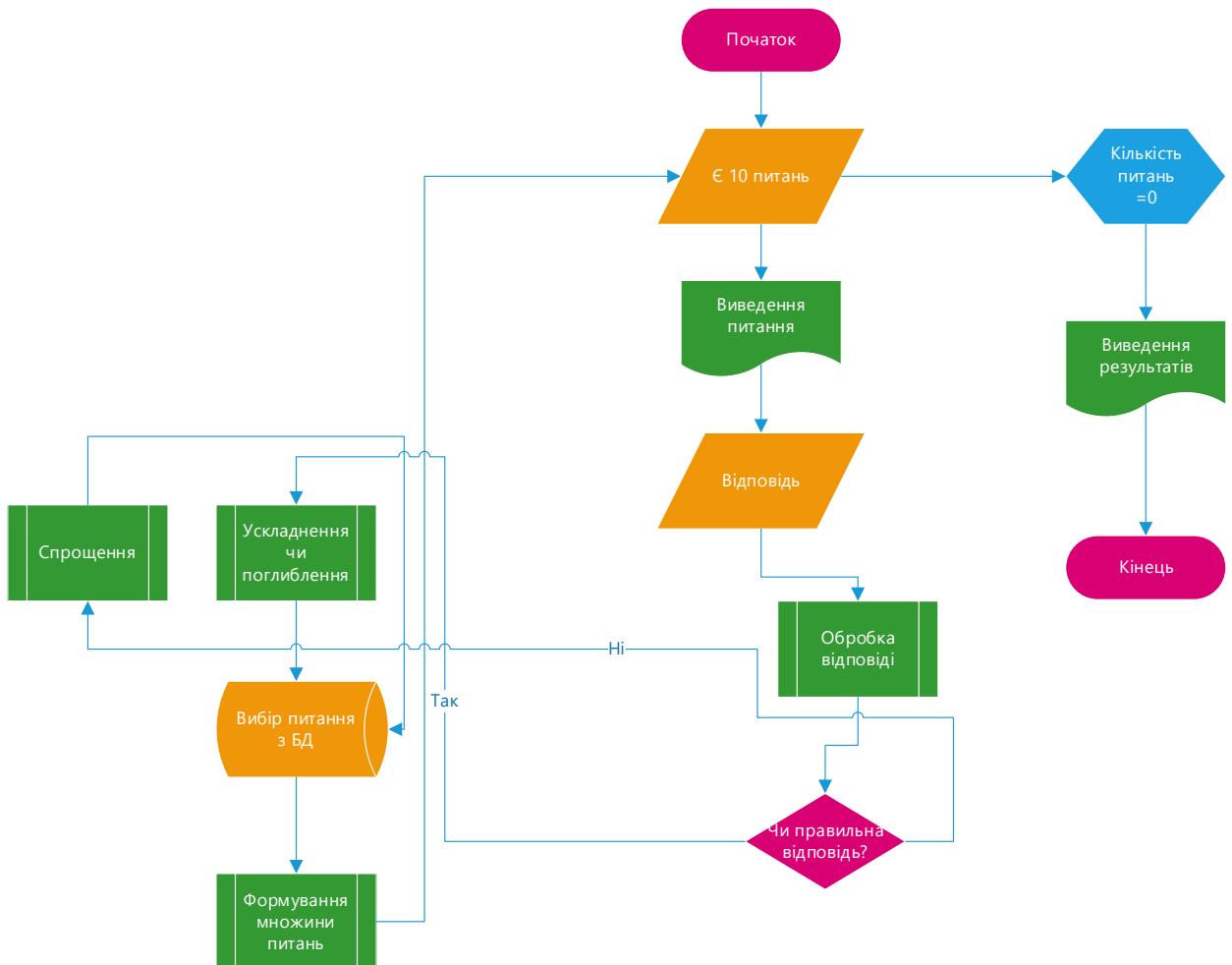


Рисунок 2.7 – Блок-схема алгоритму перебору питань та формування множини питань для тесту

Для наведеної схеми (рис. 2.7) при додаванні кожного наступного кроку з аналізу відповіді та негативної відповіді на запитання Π_{m+1} необхідно надавати лише значення t_{m+1}, N_{m+1} , а при обчисленні уточненого значення від позитивної відповіді $t(m, m+1)$ оперувати лише значенням $t(m), \sum_{j=1}^m N_j, t_{m+1}, N_{m+1}$. Відповідно, застосування блок-схеми (рис. 2.6) до

уточнених результатів з вибору питань $\Pi_{m+1}, \Pi_{m+2}, \dots$ дозволяє спростити процес обміну додатковою інформацією та обчислення уточнених значень за кожним кроком $t(m, m+1), t(m+1, m+2), \dots$

Але при цьому було враховане лише математичне очікування того, що респондент дасть вірні відповіді, які відповідають поставленому завданню, тобто, $\tau(m) \approx t$. Якщо на схемі (рис. 2.6) математичне очікування τ_{ij} представити як величину $d_{ij} = (t - \tau_{ij})^2$, то вираз

$$D(m) = \frac{\sum_{j=1}^m \sum_{i=1}^{N_j} d_{ij}}{\sum_{j=1}^m N_j} \quad (2.7)$$

дозволяє визначити дисперсію $D(m)$ випадкової величини τ_{ij} . Тоді блок-схема (рис 2.7) можлива до реалізації при застосуванні згортки значень $(t - \tau_{ij})^2$, що значно спростить алгоритмізацію та процес обчислень $\sigma(m, m+1)$.

Але все одно математичне обчислення за алгоритмом (рис. 2.7) буде занадто складним до реалізації, а процес обчислення після кожного визначення відповіді за відповідністю до деякого ключового слова з БД системи тестування займатиме тривалий інтервал часу (в експериментальній версії 45 – 67 сек). Тому необхідно внести ще низку вдосконалень до математичної основи алгоритму, вираженого блок-схемою.

Враховуючи, що проходження тесту відбувається за первинним рандомним вибором питань, при збільшенні числа вимірів результати відповіді на перше запитання «хиба» чи «істина» потрапляють до вибірки, що описується за нормальним законом розподілу. Тож при лагортмізації процесу формування вибірки питань, при розбитті питань на окремі інтервали, з яких знову вибирається питання (приклад, рис.2.3 та рис. 2.4), вибір можна здійснювати за оцінки довірчих інтервалів вибіркового середнього. Для цього

можна використати критерій Стюдента [53]. У цьому випадку значення часу на обчислення операції вибору нового питання і формування вибірки з урахуванням похибок виміру (\tilde{t}) може бути обчислено за наступною моделлю:

$$\tilde{t} = t + \Delta t = \tau(m) \pm \Delta\tau(m) = \tau(m) \pm t_s \frac{\sigma}{\left(\sum_{j=1}^m N_j\right)^{1/2}}, \quad (2.8)$$

де
$$\sigma = \sqrt{\frac{\sum_{j=1}^m (\tau_j - \tau(m))^2}{\sum_{j=1}^m N_j - 1}};$$

$\tau(m)$ – середнє значення вибірки, що формується (кількість питань);

$\Delta\tau(m)$ – довірчий інтервал;

t_s – коефіцієнт Стюдента для заданої довірчої ймовірності P_d .

Якщо прийняти $P_d = 0,95$, то обчислення за алгоритмом (рис. 2.7) з формування множини питань відбуватимуться за 1-3 сек, що буде практично непомітним для респондента.

2.4 Висновок до другого розділу

За другим розділом кваліфікаційної роботи рівня «Магістр» можна зробити наступні висновки і узагальнення:

1) наведена схема елементарних дій респондента при відповіді на завдання є дзеркальним відображенням схеми побудови тестового завдання. Це дозволяє побудувати тестове за таким механізмом та відповідною моделлю, який буде сприятливим для виконання респондентом, який не

володіє спеціальним знаннями та вміннями у роботі з комп'ютерною технікою та мережею Інтернет;

2) засобами VBA було реалізовано два алгоритми, які дозволили генерувати випадковий номер питання із масиву (бази даних системи тестування) з розбиттям всього масиву на інтервали за складність, що потім дозволяє створювати вибірки питань та здійснювати перебор в залежності від відповіді респондента на попереднє питання тесту. Проведений експеримент дозволив провести деякі заміри часу відгуку тестової системи та визначити питання, що потребують подальшого удосконалення;

3) проведена алгоритмізація процесу обробки відповіді для віднесення наступного питання до вибірки, представлені математичні основи, виконане моделювання зі спрощення обчислень, що дозволяють скоротити час обробки інформації для формування вибірки питань тестування.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ ФОРМУВАННЯ МНОЖИНИ ПИТАНЬ ПРИ ОН-ЛАЙН ТЕСТУВАННІ

3.1 Формування бази даних системи тестування

Реалізація системи тестування на веб-платформі потребує створення бази даних, яка утримуватиме масиви питань за темою, ключових слів, даних реєстрації та ідентифікації респондентів, а також тимчасову базу для утримання вибірки питань для поточного тестування.

При реалізації механізму тесту осовою, як зазначалося раніше, є схема елементарних дій респондента. Сукупність правил формування відповіді може бути побудована за наступною схемою:

а) ключовим словом є безпосередньо слово або висловлювання, яке входить до словника сфери опитування. Таке слово приймається формулою (осовою) для відповіді на запитання тесту;

б) кроком тесту визнається перехід між ключовим словом (словами) та словами із базового словника, що описують сферу опитування через логічні висловлювання: кон'юнкція, диз'юнкція, логічне заперечення, імплікація, еквівалентність;

в) обмеженням є те, що крок тесту розглядається лише за принципом істини або хибі.

Фактично застосування такого правила є рух по вузлах деякого дерева висловлювань: кореневий вузол – ключове слово, а далі покрокові вузли-зв'язки, що дозволяють кожному кроці тестування з'ясувати відповідність відповіді поставленій задачі через словник тесту. Тоді базу даних тестової системи можна представити логічною формулою:

$$(p \wedge (q \vee r)), \quad (3.1)$$

де:

p, q, r – є різноманітними можливими висловлюваннями з утриманням ключових слів та слів словника за завданням тесту.

Крок тесту у формулі (3.1) фактично застосовується двічі – при побудові спочатку висловлювання $(q \vee r)$, данні якого порявнюються з наявними у словнику, а потім поєднання з висловлюванням p , яке міститься в умові завдання i , знов таки, перевіряється за словником БД. Якщо при побудові моделі міркування деякий елемент (слово) X пов'язані з елементом (словом) Y виступить елементом індукції (значенню присвоюється «істина», з урахуванням збігу з наявними даними), то цьому разі Y і буде об'єднаним висловлюванням, яке співпадає зі словами у БД.

Формування множини питань передбачає попереднє формування бази даних i , зокрема, словника, куди додаються необхідні терміни, з яких формуються логічні ланцюги за формулою (3.1) з урахуванням кроків тесту. Словник системи повинен забезпечувати дві вимоги: ключові слова повинні бути впорядковані, а сам словник повинен мати інформаційну надмірність. У цьому випадку механізм базового словника може бути описаний такими виразами:

– $A_j = (a_1 \dots a_i \dots a_n)$ - j -е слово словника;

– $i = \overline{1, n}; j = \overline{1, N}$;

– q – ключове слово.

Кожному ключовому слову a_i чи слову зі словника, з яким буде порівнюватися відповідь респондента протиставляється деяке значення $\alpha_{k_i} = \{\alpha_{0_i} \dots \alpha_{k_i} \dots \alpha_{(q-1)_i}\}$, за умови:

$$\alpha_0 = 0; \quad \alpha_{q-1} = q - 1; \quad \alpha_{k+1} = \alpha_k + 1.$$

Тобто, привласнюємо порядковий номер q з відповідністю ключовому слову або слову зі словника окремого завдання з переліку завдань системи. У цьому випадку база даних буде упорядкована за моделлю:

$$\sum_{i=1}^n (\alpha_{k_i} \cdot q^{n-i})_j \rangle \sum_{i=1}^n (\alpha_{k_i} \cdot q^{n-i})_{j+1}. \quad (3.2)$$

За використання моделі (3.2) база даних упорядковуватиметься за зменшенням значень $\frac{\alpha_{k_i}}{\alpha_{k_{i+1}}} = q$. Тобто, останнє використане слово, відповідність якому знайдена у словнику системи, вказує на відповідь респондента, що передбачає відкриття подальшого кроку у проходженні тестування (див. рис. 2.7). Інформаційна надмірність словника системи повинна забезпечувати вимогу, за якою q^n різноманітних значень комбінацій слів n для правильної відповіді завдання і повинна забезпечувати факторіал цього числа n .

У цьому випадку механізм формування БД тестової системи може бути представлено наступним алгоритмом:

- 1) Формування масиву слів a_1, a_2, \dots, a_i ;
- 2) Формування множини m_i ключових слів та слів базового словника з обмеженою областю пошуку відповідності $m_1 \rangle m_2 \rangle \dots \rangle m_i$;
- 3) Формування множини питань, яким відповідає ключове слово відповіді a_i зі словника БД m_i ;
- 4) Формування підмножини слів з меншею ймовірністю відповідності відповіді тестовим питанням, але яке буде прийняте як відповідь з обмеженням m_{i+1} ;
- 5) Перелік (множина слів), які точно вказують на невідповідність відповіді (присваюється значення «хиба»).

Зазначене можна представити наступною блок-схемою (рис. 3.1).

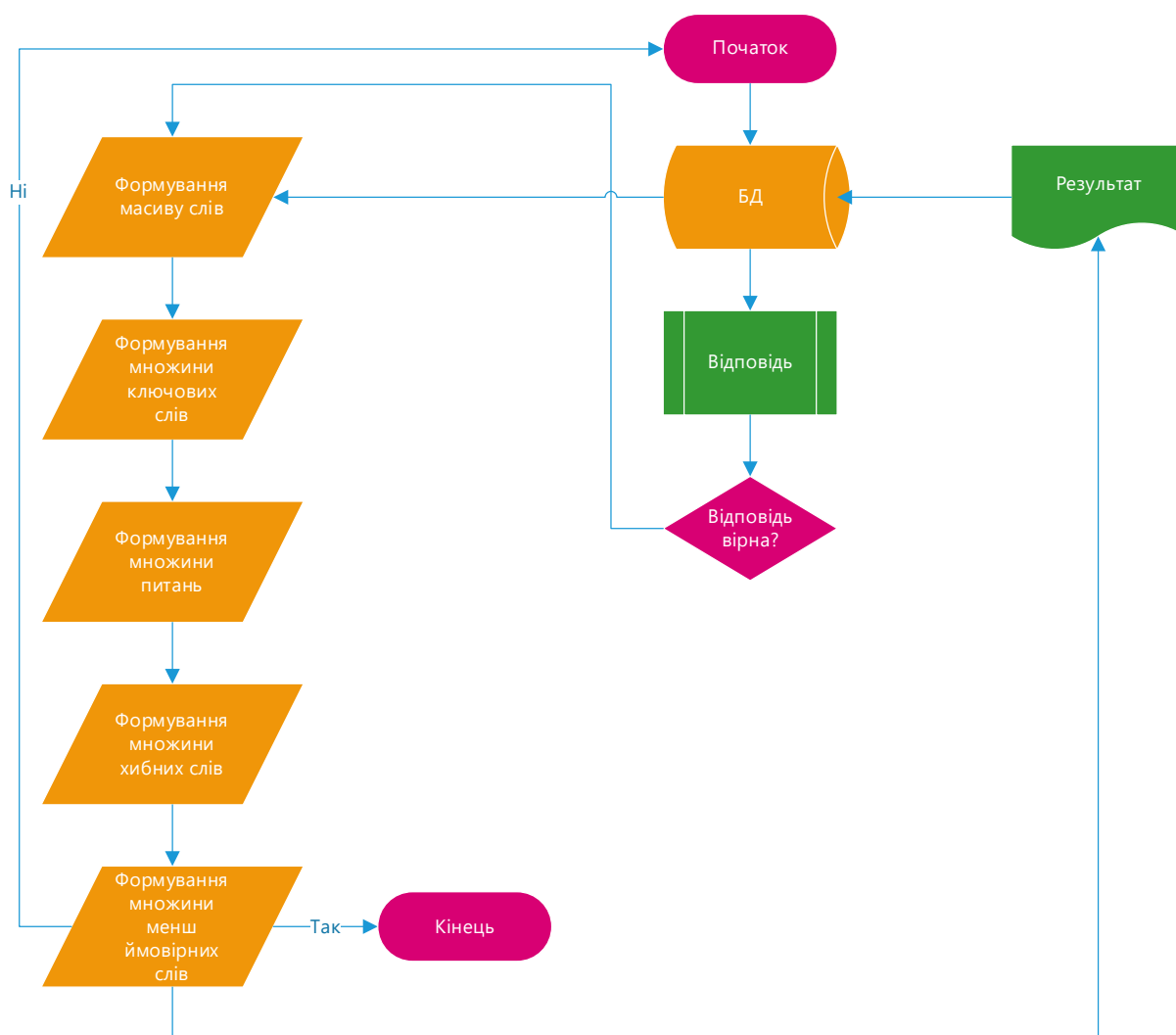


Рисунок 3.1 – Алгоритм механізму бази даних тестової системи

У Додатку В наведено лістинг формування базового словника тестової системи мовою універсального призначення Pascal, який сформовано як файл з N упорядкованих ключових слів, які мають n -символів використаного природного алфавіту q . Також у файлі наведена деяка кількість D слів, які можуть бути використані при відповіді, серед яких:

– a_i – слова-аналоги $a_1 a_2 \dots a_i$;

– \hat{a}_i -поле – частки слів, які схожі за значенням $a_1 a_2 \dots a_i$;

– $m(\hat{a}_i)$ – кількість слів \hat{a}_i -поля, які часто використовуються при

відповіді за даною тематикою;

- $AD^H(\hat{a}_i)$ – номер першого слова \hat{a}_i - поля;
- $AD^K(\hat{a}_i)$ – номер останнього слова \hat{a}_i - поля;
- $A_o^H(\hat{a}_i)$ – мінімально можлива кількість слів (символів) відповіді $a_1 a_2 \dots a_i 00 \dots 0$;
- $A_o^K(\hat{a}_i)$ – максимально можлива кількість слів (символів) відповіді $a_1 a_2 \dots a_i (q \cdot 1)(q \cdot 1) \dots (q \cdot 1)$;
- $A[j]$ – ознаки співвідношення слова та завдання тету j .

Тобто, для введеного у відповіді слова A_d , існує ключове слово у масиві БД ($d = 1 \dots D$). Рішенням, при $i := 1, 2, \dots$, є відповідність умові:

$$m_{min} \leq m(\hat{a}_i) \leq m_{max}.$$

Цей підхід був протестований. Кількість використаних слів дорівнювало 100 – 1000 слів. Відповідь могла складатися від 5 до 15 слів або 228 символів. Одночасно з цим проводилися розрахунки можливої швидкості знаходження слова у БД. Результати тестування наведені у табл. 3.1.

Таблиця 3.1 – Результати розрахунку та експерименту з тестування представленої імітаційної моделі реалізації механізму бази даних тестової системи

Текст, кількість слів	Розрахункове значення, сек.	Експериментальне значення, сек.	Кількість слів у відповіді
1000 слів	3,29	3,30	11
500 слів	2,68	2,70	6
100 слів	1,07	1,08	3

Процедура розрахунку може бути реалізована за допомогою моделі за технологією GOMS [46] з використанням теорії ймовірності.

3.2 Методика та алгоритм формування множини питань

Базу даних тестової системи можна розглядати, як реєстр з q^n комірок, серед яких рандомно з середнім інтервалом $l = q^n/N$ розташовано N обраних комірок з врахуванням $q^n \gg N \gg 1$, як це було наведено при реалізації алгоритму генерації випадкового вибору питання з масиву для формування вибірки тестування респондента (рис. 2.3). Навернений на рис. 2.3 підхід доповнюється наступними властивостями:

а) приймається, що одна з обраних комірок (далі – «комірка+») відповідає ключовому слову деякого питання;

б) кожній «комірці+» присвоєно значення p_j з врахуванням $\sum_{j=1}^N p_j = 1$.

Це значення має ймовірність того, що при довільному зверненні до БД, «комірка+» є рішенням A_j . Хоча з урахуванням впорядкованості словника (рис. 3.2) значення p_j розподілені випадковим чином (рис. 3.3).

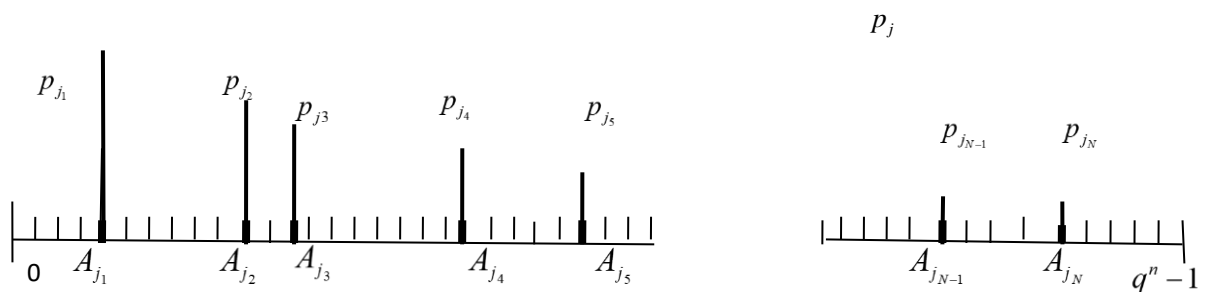


Рисунок 3.2 – Впорядкований розподіл словника

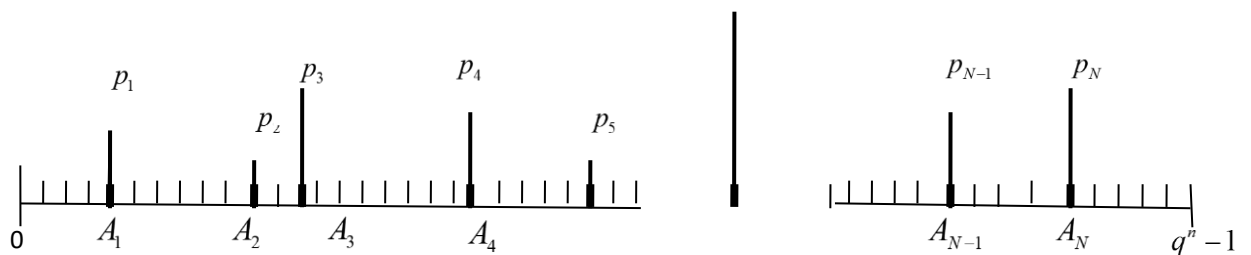


Рисунок 3.3 – Розподіл словника при зверненні на пошук слова

Якщо упорядкувати всі «комірки+» за убутанням p_j (рис. 3.2), та умовно перенести їх на позиції справа за номерами значень p_j у вибірці, то можна провести апроксимацію значень та отримати дискретний розподіл ймовірностей p_{j_s} безперервної функції $p(x)$ (рис. 3.4) та виділити на осі x точку $L = q^n$. Функція $p(x)$ буде відповідати умові $\int_0^L p(x)dx = 1$ достатності сформованої вибірки питань словам, що є у словнику БД тестової системи.

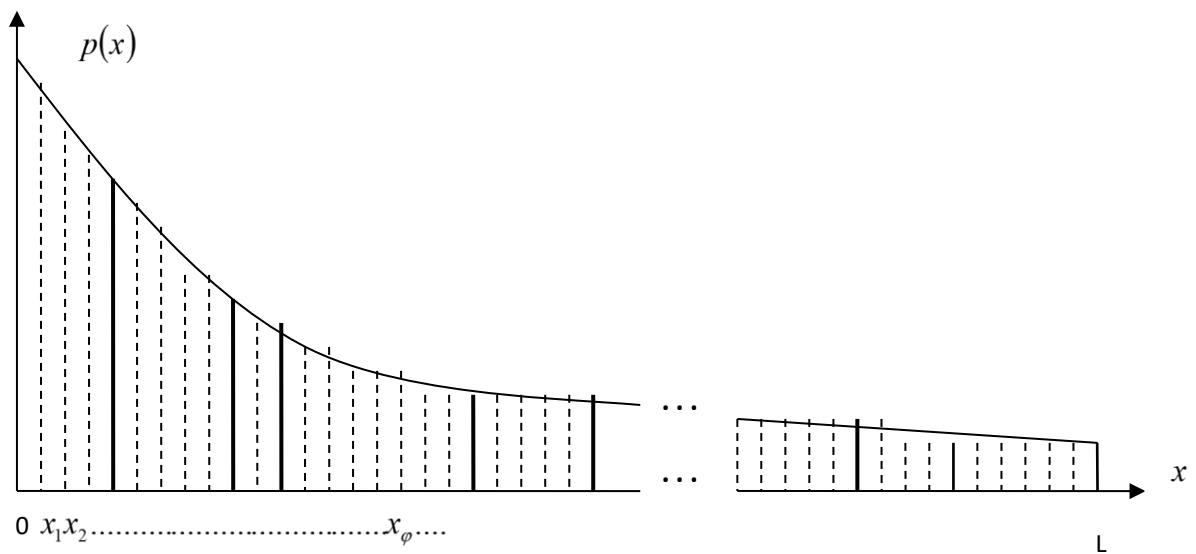


Рисунок 3.4 – Апроксимація значень вибірки для отримання множини питань

Якщо виділити на осі x (рис. 3.4) q^n точок з координатою x_φ ($\varphi = 1, 2, \dots, q^n$) та поставити у відповідність кожній точці число π_φ , то

$$\pi_\varphi = \int_{x_{\varphi-1}}^{x_\varphi} p(x)dx; \quad \sum_{\varphi=1}^{q^n} \pi_\varphi = \int_0^L p(x)dx = 1.$$

Число π_φ має ймовірність того, що для словника БД з параметрами N , q , n довільна комірка φ утримує ключове слово для відповіді на вибране питання.

Представимо алгоритм вибору питання до масиву питань тестового завдання як покроковий процес, що повторюється багато разів стосовно додавання та вилучення питань з вибірки:

- 1) В базі даних є q^n питань, кожному з яких є деяка відповідь π_φ ;
- 2) З цієї кількості питань випадково, з ймовірністю π_φ обирається питання, яке відповідає «комірці+» за умовою тесту;
- 3) На наступному кроці i ($i = 1, 2, \dots$) з q^{n-i+1} питань випадковим чином з рівною ймовірністю обирається наступне питання $q^{n-i} - 1$.
- 4) Перше питання є таким, яке визначатиме схему формування тесту при відповіді істинній або хибній, інші $q^{n-i} - 1$ питань можуть бути замінені;
- 5) Питання, що відбираються до множини, можуть з однаковою ймовірністю як залишитися у множині тесту, так і бути вилученими. Тому до множини потрапляють деяка кількість питань g_i , і їх у вибірці є $m_i = g_i + 1$ питань;
- 6) Кількість m_i упорядковуються з деякою ймовірністю відповіді на них (складності) π_φ ;
- 7) Після відповіді на перше питання є ймовірність з найбільшим значенням π_φ для деякої кількості питань m залишитися у вибірці. Вони заєжать від кількості ключових слів, що задіяв користувач у попередніх відповідях;
- 8) Якщо у першій відповіді не було ключових слів – алгоритм завершено;
- 9) Інакше $i := i + 1$ кроки заміни і додавання питань повторюються, поки не буде вичерпана мета тесту або респондент не застосує жодного ключового слова.

Лістинг коду реалізації наведеного алгоритму формування множини питань у вигляді фреймового дерева наведена у Додатку Г.

Сутність програмної реалізації наведеного алгоритму полягає у наступному: слід визначити розподіл ймовірності значень i , при якій закінчується багаторазовий процес вибору питань до множини. Також слід визначити середнє значення i , для того, щоб процес відбору питань до множини не був занадто тривалим. Для цього приймемо його вираження через $v_{\text{ср}}(m/m_i)$.

Оскільки на кожному кроці процесу набір ймовірностей π_φ множини щільністю заповнення g_i формується випадковим чином із сукупності π_1, \dots, π_{q^n} , впорядкований розподіл π_φ на кожному кроці звуження області пошуку можна апроксимувати функцією $p(x)$. При цьому варто ввести поняття довжини ξ_{g_i} деякої віртуальної «порції» вибраних питань за наступною схемою:

$$\xi_{g_i} = L/k(g_i, m),$$

де $k(g_i, m)$ – кількість «порцій» питань, об'ємом m або не менше (якщо $g_i < m - 1$) питань в кожній «порції», що мають в сумі $g_i + 1$ питань.

Це можна представити моделлю:

$$k(g_i, m) = \begin{cases} \frac{g_i + 1}{m} & \text{для } g_i > m - 1, \\ 1 & \text{для } g_i \leq m - 1 \end{cases}, \quad \xi_{g_i} = \begin{cases} \frac{L m}{g_i + 1} & \text{для } g_i > m - 1 \\ L & \text{для } g_i \leq m - 1 \end{cases}.$$

Тобто, безумовні ймовірності $p_1(i)$ завершення процесу на кроці i дорівнюють:

$$p_1(i) = p_{1g_i} \prod_{s=1}^{i-1} (1 - p_{1g_{i_s}}),$$

де p_{1g_i} є умовними ймовірностями завершення процесу на кроці i за умови, що процес не було завершено на кроках $1, 2, \dots, i-1$, які були задані фізично, обмежуючи питання тесту чітким набором питань. За зазначеного, відповідне середнє значення $v_{1cp}(m/m_i)$ дорівнюватиме:

$$v_{1cp}(m/m_i) = \sum_{i=1}^n i \cdot p_1(i) + n \prod_{s=1}^n (1 - p_{1g_{i_s}}).$$

Для визначення ймовірності p_{1g_i} можна прийняти позначення для деякого тесту S можливих замін при відповіді на попередні питання на кроці i :

S_{1g_i} – на кроці i процес закінчено;

$S(g_i)$ – до вибірки потрапило g_i питань $q^{n-i} - 1$, які не відповідають темі опитування або рівню респондента;

$S\left(\frac{m}{g_i}\right)$ – до «порції» з m питань, що обираються рандомно з вибірки $g_i + 1$ питань БД, потрапило питання, що вже було використане.

Відповідно,

$$S_{1g_i} = S(0) \wedge S\left(\frac{m}{1}\right) \vee S(1) \wedge S\left(\frac{m}{2}\right) \vee \dots \vee S(q^{n-i} - 1) \wedge S\left(\frac{m}{q^{n-i}}\right) = \bigcup_{g_i=0}^{q^{n-i}-1} S(g_i) \wedge S\left(\frac{m}{g_i}\right).$$

Звідси

$$p_{1g_i} = \sum_{g_i=0}^{q^{n-i}-1} \pi(g_i) \cdot \pi\left(\frac{m}{g_i}\right),$$

де

$\pi(g_i)$ – ймовірність того, що на кроці i при виборі $q^{n-i} - 1$ питань у вибірці виявилось g_i питань;

$\pi(m/g_i)$ – ймовірність того, що у вибірці обсягом m найбільш ймовірних питань, які вибираються з «порції» $g_i + 1$ питань, виявилось повторення питання.

ЦУ цьому випадку, якщо використати модель випробувань Бернуллі [54], може бути отримана наступна модель:

$$\pi(g_i) = P(g_i, r, q^{n-i} - 1) = C_{q^{n-i}-1}^{g_i} \cdot r^{g_i} (1-r)^{q^{n-i}-1-g_i}.$$

Це на основі «порції» питань дозволяє представити $\pi(m/g_i)$ як площу, яка обмежена абцисами $x = 0$, $x = \xi_{g_i}$ та кривою безперервної функції $p(x)$, яка апроксимується (рис. 3.5):

$$\pi(m/g_i) \approx \int_0^{\xi_{g_i}} p(x) dx.$$

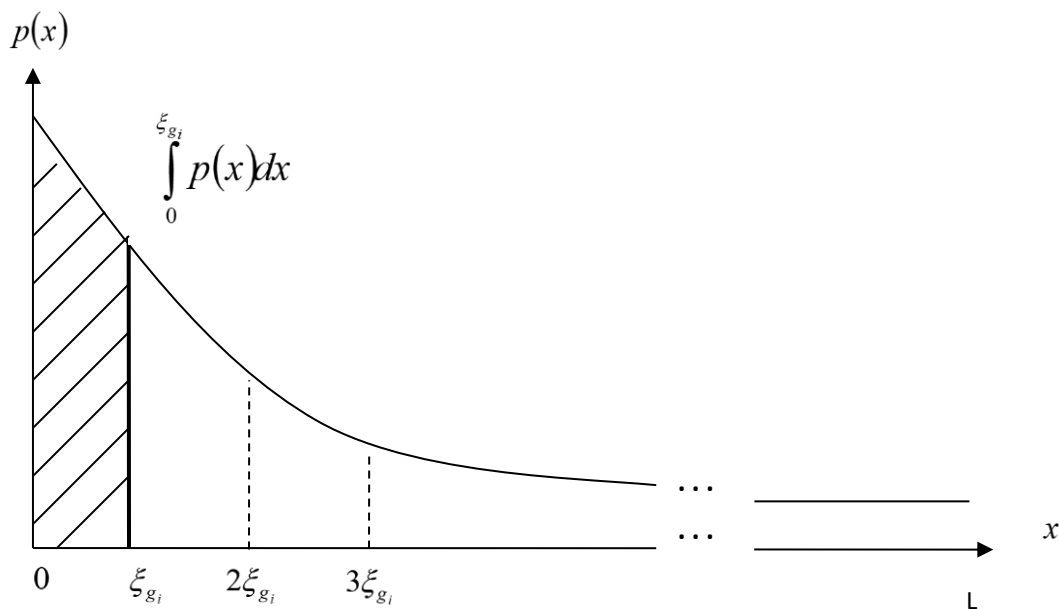


Рисунок 3.5 – Площа вирішення задачі

У підсумку отримується вираз

$$p_{1g_i} = \sum_{g_i=0}^{q^{n-i}-1} \left[P(g_i, r, q^{n-i} - 1) \cdot \int_0^{\xi_{g_i}} p(x) dx \right],$$

який дозволяє визначити розподіл ймовірностей $p_1(i)$ та середнє значення $v_{1cp}(m/m_i)$.

В цілому розуміння «порції» питань для реалізації даного алгоритму дозволяє розробнику визначити ймовірність $\pi(m/g_i)$ в залежності від зменшення величини q^{n-i} та поточних значень g_i через збільшення масштабу функції $p(x)$. Це досягається шляхом рівняння $g_i + 1 = L$ та збільшення обсягу «порції» (довжини відрізка $0 \div \xi_{g_i}$) до значення $m \cdot \frac{L}{g+1}$, та далі – до $g_i \leq m-1$, коли $\xi_{g_i} = L$ и $\pi(m/g_i) = 1$. Останнє виключає дублювання питань, що входять до множини.

3.3 Експериментальна перевірка механізму формування множини питань для он-лайн тестування

Для перевірки викладеного та наведеної програмної реалізації проведений експеримент для порівняння знаходження відповідності відповіді питанню за розробленою технологією та реалізацією тестового завдання через Google. Визначення точності відповідей T_1, T_2, T_3 в порівнянні з реальним значенням $T_{\text{експериментальне}}$ для конкретного ключового слова.

За експериментом слід було ввести частину слова, щоб система визначила, що це за слово. Слово визначалося за кількістю символів \bar{v} що були введені та значенням експериментального введення слова у систему при наявності в ній ключового слова, яке виступало зразком.

Алгоритм експерименту:

1. Запуск Google.

2. Введення першої (2-ї, 3-ї...) літери слова і очікування пропозиції системи з словами, які пропонуються для пошуку;

3. Фіксація пропозицій пошукової системи та кількості літер, на якій було визначене «задумане» слово. У випадку визначення – запис часу τ_i та кількості введених символів v_i .

Для швидкісного Інтернету час реакції системи через пошуковик Google практично не можливо зафіксувати. В процесі експерименту вводилося 20 слів трьома користувачами. Результат сотосвно тестування розробленого механізму наведено в табл. 3.2.

Таблиця 3.2 – Результати проведення експерименту

№	Слово	v_i	τ_i (сек)
1	захворювання	5	7
2	біль	3	5
3	гістограма	8	9
4	система	4	7
5	вибуховий	3	7
6	функція	3	6
7	зміст	3	5
8	форма	3	5
9	базис	3	5
10	група	4	5
11	поранення	5	7
12	масив	4	5
13	клас	3	4
14	символ	6	5
15	свідомість	6	7
16	провали	7	7

17	кадр	3	4
18	скрин	5	4
19	ампутація	5	7
20	експериментальний	8	10
Σ		91	121
$\bar{v} = \Sigma v_i / 60 = 1,52$ $T_{\text{екс}} = \Sigma \tau_i / 60 = 2,02(\text{сек})$			

Тобто, в середньому на обробку одного слова уходило майже дві хвилини, а на обробку одного символу – дві секунди.

В табл. 3.3 наведені середні результати виконання завдання трьома користувачами через систему Google та порівняння з експериментальними даними.

Таблиця 3.3 – Розрахункові та експериментальні дані щодо пошуку слова в системі

$T_1(\text{сек})$	$T_2(\text{сек})$	$T_3(\text{сек})$	$T_{\text{екс}}(\text{сек})$
<1	<1	<1	2,02

Як видно з табл. 3.3, оперативність розробленого механізму значно поступається за часом обробки механізму Google. Проте наведений експеримент не враховує дію стандартного оператора M , який дозволяє особливості роботи з користувачем із застосуванням ментальних операторів. Тобто, у даному разі лише перевірена зрадність до обробки ключового слова наведеної системи. Якщо ж врахувати, що це система для тестування і тут можливо застосування користувачем роздумів, змін своєї відповіді, то можливість визначити наступне питання за попередньою відповіддю респондента затримка системи у 2 сек буде не помітна кінцевому користувачу.

3.4 Висновок до третього розділу

За третім розділом кваліфікаційної роботи рівня «Магістр» можна зробити наступні висновки і узагальнення:

1) представлений опис та програмна реалізація бази даних розробленої системи з визначенням особливостей формування множин питань, ключових слів та слів словника, які можуть бути використані для ідентифікації відповіді стосовно обраного питання;

2) наведена методика та алгоритм формування множини питань, зокрема зазначено, що для реалізації алгоритму використаний підхід, коли визначається розподіл ймовірних значень (номерів питань), при яких закінчується багаторазовий процес вибору питань до множини. Крім того, необхідно в процесі реалізації механізму визначати середнє значення кількості питань, якщо кінцеве число не було встановлене, як незмінна величина. Це слід використати для того, щоб процес відбору питань до множини не був занадто тривалим;

3) експериментальна перевірка механізму формування множини питань для он-лайн тестування дозволила провести деякі аналогії для порівняння розробленого механізму формування множини питань при он-лайн тестування, зокрема, ідентифікації ключового слова системою та близьких за змістом дій у пошуковій системі Google. Середній час виконання завдання розробленою системою складає 2,02 сек. І хоча час тривалий, проте отриманий результат задовільний, адже розроблений механізм передбачає врахування ментальних операторів, які є точками «мислення» респондента, а, отже, затримка системи не буде помітна кінцевому користувачу, що забезпечується специфікою опитувань від замовника системи і підтверджено апробацією.

ВИСНОВКИ

У кваліфікаційній роботі рівня «Магістр»:

1) виконано огляд систем, які можна використати при розробці багатофункціонального он-лайн тестування, визначені основні існуючі проблеми та окреслено можливі шляхи рішення;

2) зазначено, що від респондента тестування не повинно вимагати розуміння механізму і знання термінів, якими описується предметна область, знання теорії реляційних баз даних, знання структури та мови SQL;

3) підкреслено, що основою створюваного механізму формування множини питань для проведення он-лайн тестування має бути попередній опис семантики тесту природною мовою за допомогою стандартного CASE-засобу, однією з основних переваг якого є можливість створення концептуальної (логічної) моделі реляційної бази даних;

4) означено, що структурованим запитом до бази даних може виступати спочатку вибір питань, які допоможуть в'яснити початкову думку чи знання респондента, а далі, на основі першої відповіді або відповідей, яка вже виступить як структурований запит, формується множина питань для чіткого визначення думки, знань, вмінь і т. інш. респондента;

5) визначено вимоги користувача (замовника) системи он-лайн тестування доволі специфічні (реабілітація людей, що постраждали від війни), тому їх неможливо вирішити за допомогою стандартних інструментів, представлених на платформах створення тестів;

6) наведена схема елементарних дій респондента при відповіді на завдання є дзеркальним відображенням схеми побудови тестового завдання;

7) реалізовано засобами VBA два алгоритми, які дозволили генерувати випадковий номер питання із масиву (бази даних системи тестування) з розбиттям всього масиву на інтервали за складністю, що потім дозволяє створювати вибірки питань та здійснювати перебор в залежності від відповіді респондента на попереднє питання тесту;

8) проведена алгоритмізація процесу обробки відповіді для віднесення наступного питання до вибірки, представлені математичні основи, виконане моделювання зі спрощення обчислень, що дозволяють скоротити час обробки інформації для формування вибірки питань тестування;

9) представлено опис та програмна реалізація мовою Pascal підходу до формування бази даних розробленої системи з визначенням особливостей формування множин питань, ключових слів та слів словника, які можуть бути використані для ідентифікації відповіді стосовно обраного питання;

10) наведено методику, алгоритм формування та програмну реалізацію множини питань, зокрема зазначено, що для реалізації алгоритму використаний підхід, коли визначається розподіл ймовірних значень (номерів питань), при яких закінчується багаторазовий процес вибору питань до множини;

11) експериментальна перевірка механізму формування множини питань для он-лайн тестування дозволила провести деякі аналогії для порівняння розробленого механізму формування множини питань при он-лайн тестування, зокрема, ідентифікації ключового слова системою та близьких за змістом дій у пошуковій системі Google. Середній час виконання завдання розробленою системою складає 2,02 сек.

ПЕРЕЛІК ДЖЕРЕЛ

1. Литовченко І.Л. Інтернет-маркетинг: навч. посіб. для студ. вищ. навч. закл. / І. Л. Литовченко. — К. : Центр. учб. л-ри, 2011. — 332 с. : іл., табл. — Бібліогр.: с. 324—331 (100 назв). — ISBN 978-611-01-0346-6
2. Arrivé S. (2021). Digital brand content: underlying nature and rationales of a hybrid marketing practice. *Journal of Strategic Marketing*. 30 (4): 340–354. [doi:10.1080/0965254X.2021.1907612](https://doi.org/10.1080/0965254X.2021.1907612)
3. Google Classroom – <https://classroom.google.com/>
4. Google Форми – https://www.google.com/intl/uk_ua/forms/about/
5. Quizizz – <https://quizizz.com/>
6. SurveyMonkey – <https://www.surveymonkey.com/>
7. Formative – <https://goformative.com/>
8. Polleverywhere – <https://www.polleverywhere.com/>
9. Socrative – <https://socrative.com/>
10. Wooclap – <https://www.wooclap.com/>
11. Moodle – <https://moodle.org/>
12. Flippity – <https://www.flippity.net/>
13. Online Test Pad – <https://onlinetestpad.com/>
14. Kahoot! – <https://kahoot.it/>
15. Морозов А. О., В’юн В. І., Кузьменко Г. Є. Інтелектуалізація інформаційних систем: орієнтація на формування знань в процесах аналізу “інформаційних згорток” / А.О. Морозов, В.І. В’юн, Г.Є. Кузьменко // *Мат. машини і системи*. – 2005. – № 2. – С. 140-146. – Бібліогр.: 8 назв. – укр.
16. Бідюк П. І., Коршевніук Л.О. Проектування комп’ютерних інформаційних систем підтримки прийняття рішень: Навчальний посібник. – Київ: ННК „ІПСА” НТУУ „КПІ”, 2010. – 340 с.
17. Raskin, Jef (1999): The User Interface in Text Retrieval Systems Revisited, A Letter to the Editor. In *ACM SIGCHI Bulletin*, 31 (1) pp. 37. <https://www.acm.org/sigchi/bulletin/1999.1/raskin.pdf>

18. Raskin J. (2001). Turning the Art of Interface Design into Engineering. In: Little, Murray Reed, Nigay, Laurence (eds.) EHCI 2001 - Engineering for Human-Computer Interaction, 8th IFIP International Conference May 11-13, 2001, Toronto, Canada. pp. 5-6.
<https://link.springer.de/link/service/series/0558/bibs/2254/22540005.htm>
19. Waldrop M. Mitchell (2016). [The chips are down for Moore's law](#). Nature. 530 (7589): 144–147. [doi:10.1038/530144a](https://doi.org/10.1038/530144a)
20. Thackray A., Brock D. C., Jones R. (2015). Moore's Law: The Life of Gordon Moore, Silicon Valley's Quiet Revolutionary. New York: Basic Books. [ISBN 978-0-465-05564-7](#).
21. Jacob R., Girouard A., Hirshfield L., Horn M., [Shaer O.](#), [Solovey E.](#), [Zigelbaum J.](#) (2008). Reality-Based Interaction: A Framework for Post-WIMP Interfaces. CHI '08: Proceedings of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems. Florence, Italy: ACM. pp. 201–210. [doi:10.1145/1357054.1357089](https://doi.org/10.1145/1357054.1357089)
22. Рябчун Ю. В. Інтелектуалізація систем підтримки прийняття рішень щодо вибору спеціалізації навчання / Ю. В. Рябчун // Управління розвитком складних систем : зб. наук. праць / Київ. нац. ун-т буд-ва і архітектури ; гол. ред. Лізунов П. П. – Київ : КНУБА, 2019. – № 39. – С. 95 - 99. <https://repository.knuba.edu.ua/handle/987654321/2694>
23. Kozinets R. V. (2002). The Field Behind the Screen: Using Netnography for Marketing Research in Online Communities. Journal of Marketing Research, 39 (February), p. 3-8.
24. Wimmer R. D. (2011). Mass Media Research: An Introduction/ Roger D. Wimmer, Joseph R. Dominick. – Cengage Learning, 2011. – P. 260.
25. Карканица А.В. Модели, алгоритмы и технология построения адаптивных систем поддержки принятия решений: автореф. дис. канд. техн. наук : 05.13.17 / А. В. Карканица ; Белорусский государственный университет. – Минск, 2019. – 22 с. <https://elib.grsu.by/doc/61438>

26. Карканица, А.В. (2010). Построение динамической модели предметной области для решения сложно структурированных задач / А.В. Карканица // Известия Гомельского государственного университета имени Ф. Скорины. - 2010. - № 5(62). - С. 73-78. <https://elib.gsu.by/bitstream/123456789/28840/1/%D0%9A%D0%B0%D1%80%D0%BA%D0%B0%D0%BD%D0%B8%D1%86%D0%B0%D0%9F%D0%BE%D1%81%D1%82%D1%80%D0%BE%D0%B5%D0%BD%D0%B8%D0%B5.pdf>
27. Skrimizea, E.; Haniotou, H.; Parra, C. (2019). On the 'complexity turn' in planning: An adaptive rationale to navigate spaces and times of uncertainty. Planning Theory. 18: 122–142. [doi:10.1177/1473095218780515](https://doi.org/10.1177/1473095218780515)
28. Карканица, А.В. (2016). Оценка неопределенности в адаптивных системах принятия решений / А.В. Карканица // Вестник Брестского государственного технического университета. Серия «Физика, математика, информатика». - 2016. - № 5. - С. 17-20. <https://elib.grsu.by/doc/47700>
29. York S., Lavi R., Dori Y. J., Orgill M-K. (2019). Applications of Systems Thinking in STEM Education J. Chem. Educ. 96, 12, 2742–2751 Publication Date: May 14, 2019 <https://doi.org/10.1021/acs.jchemed.9b00261>
30. Саати, Т. Принятие решений. Метод анализа иерархий / Т. Саати. - М. : Радио и связь, 1993 - 278 с.
31. Fielding R. T. (2000) Architectural styles and the design of network-based software architectures. University of California, Irvine.
32. Hintikka J. (2007). Socratic Epistemology: Explorations of Knowledge-Seeking by Questioning. <https://doi.org/10.1017/CBO9780511619298>
33. Machado P.; Vincenzi A.; Maldonado J.C. (2010). Chapter 1: Software Testing: An Overview. In Borba, P.; Cavalcanti, A.; Sampaio, A.; Woodcock, J. (eds.). Testing Techniques in Software Engineering. Springer Science & Business Media. pp. 13–14. [ISBN 978-3-642-14334-2](https://doi.org/10.1007/978-3-642-14334-2).
34. Stober F., Weiß A. (2023). Lower Bounds for Sorting 16, 17, and 18 Elements. In 2023 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX) (pp. 201-213). Society for Industrial and Applied Mathematics

35. Rosen K.H., Michaels J.G. (2000). Handbook of Discrete and Combinatorial Mathematics, CRC Press.
36. Pratt T.W., Zelkovitz M.V. (2000). Programming languages, design and implementation (4th ed.). Prentice Hall.
37. Inmon W.H. Building the Data Warehouse / Inmon W.H. – New York: John Wiley & Sons, 2005. – 543 p.
38. Ланде Д.В. Новітні підходи й технології інформаційно-аналітичної підтримки прийняття рішень. // Національна безпека: український вимір: щокв. наук. зб. / Рада нац. безпеки і оборони України, Ін-т пробл. нац. безпеки; - К., 2008. - Вип. 1-2 (20-21). - С. 87-105.
39. Кузьменко Г.Є. Інструментальні засоби створення єдиного інформаційного середовища корпоративних систем / Г.Є. Кузьменко, І.М. Оксанич, Ю.Г. Пилипенко // Математичні машини і системи. - 2004. - №4. - С. 126–135.
40. A universal relation model for a nested database, The Nested Universal Relation Database Model, Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, vol. 595, pp. 109–135, 1992, [doi:10.1007/3-540-55493-9_5](https://doi.org/10.1007/3-540-55493-9_5), [ISBN 978-3-540-55493-6](https://www.springer.com/978-3-540-55493-6)
41. Pratt P. J.; Last M. Z. (2014-). Concepts of Database Management (8 ed.). Course Technology. p. 29. [ISBN 9781285427102](https://www.course.com/9781285427102).
42. Wiegenstein, F. Weidemann, Dr. M. Schumacher, S. (2006) Schinzel Web Application Vulnerability Scanners - a Benchmark –5p. http://www.virtualforge.de/whitepapers/web_scanner_benchmark.pdf
43. 100+ питань для співбесід та технічних інтерв'ю – бази даних, зокрема SQL (частина 1). Електрон. текст. дані. – Режим доступу : <https://www.quality-assurance-group.com/super-puper-shpargalka-100-pytan-dlya-spivbesid-ta-tehnicnyh-intervyu-bazy-danyh-zokrema-sql-chastyna-1/>
44. Боднар Л. В. Методичні рекомендації щодо створення Інтернет-сайту освітнього закладу / Л. В. Боднар. – Одеса, 2019. – 52 с.
45. West R. L.; Nagy N. (2007). Using GOMS for Modeling Routine Tasks Within Complex Sociotechnical Systems: Connecting Macrocognitive Models to

- Microcognition. *Journal of Cognitive Engineering and Decision Making*. 1 (2): 186–211. [doi:10.1518/155534307X232848](https://doi.org/10.1518/155534307X232848)
46. Bonnie J. E., Vera A.; Mattesa M. (2002). Automating CPM-GOMS. *ACM*. [doi:10.1145/503376.503404](https://doi.org/10.1145/503376.503404)
47. Kershaw T.C., Ohlsson S. Multiple causes of difficulty in insight: The case of the nine-dot problem // *Journal of Experimental Psychology: Learning, Memory, and Cognition*. 2004. 30. 1. 3–13.
48. Knoblich G., Ohlsson S., Haider H., Rheniu D. Constraint relaxation and chunk decomposition in insight problem solving // *Journal of Experimental Psychology: Learning, Memory, and Cognition*. 1999. 25. 1534–1555.
49. Lung Ch.-T., Dominowski R.L. Effects of strategy instructions and practice on nine-dot problem solving // *Journal of Experimental Psychology: Learning, Memory, and Cognition*. 1985. II. 4. 804–811.
50. MacGregor J.N., Ormerod T.C., Chronicle E.P. Information processing and insight: a process model of performance on the nine-dot and related problems // *Journal of Experimental Psychology: Learning, Memory, and Cognition*. 2001. 27. 176–201.
51. Kryazhych O., Itskovych V., Iushchenko K., Hrytsyshyna V., Bruvier D., Nykytyuk V., Bodnarchuk I. (2023) The use of abstract moore automaton to control the sensors of a service-oriented alarm and emergency notification network. *Scientific Journal of TNTU (Tern.)*, vol 109, no 1, pp. 111–120.
52. Стивен Буллен, Роб Боуви, Джон Грин Профессиональная разработка приложений Microsoft Office Excel = Professional Excel Development: The Definitive Guide to Developing Applications Using Microsoft Excel and VBA. — М.: «Вильямс», 2007. — С. 736. — [ISBN 0-321-26250-6](https://www.isbn-international.org/product/9780321262506)
53. Edgell, Stephen E.; Noon, Sheila M. (1984). "Effect of violation of normality on the t test of the correlation coefficient". *Psychological Bulletin*. 95 (3): 576–583. [doi:10.1037/0033-2909.95.3.576](https://doi.org/10.1037/0033-2909.95.3.576)
54. Карташов М. В. Імовірність, процеси, статистика. — Київ : ВПЦ Київський університет, 2007. — 504 с.

ДОДАТКИ

Апробація результатів кваліфікаційної роботи рівня «Магістр»



СПОРТИВНО-РЕАБІЛІТАЦІЙНИЙ ЦЕНТР ДЛЯ ОСІБ З ІНВАЛІДНІСТЮ ТА ВЕТЕРАНІВ ВІЙНИ ІМЕНІ ОЛЕГА КРАВЦЯ

Україна, 88000, Закарпатська область,
 Ужгородський район, місто Ужгород,
 вул. Франтішека Тихого, будинок 13Б
 UA363052990000026000023604118
 +380 (066) 150 28 94

№ 110 від 01 . 12 . 2023 року

ДОВІДКА

щодо апробації результатів кваліфікаційної роботи рівня «Магістра» на тему
 «Інформаційна технологія формувань множини питань при онлайн
 тестувань» Брувера Данила Дмитровича

Результат кваліфікаційної роботи Брувера Данила Дмитровича на тему
 «Інформаційна технологія формувань множини питань при онлайн тестувань»
 було апробовано під час проведення оцінки інтернет опитувань осіб з
 інвалідністю (порушення опорно-рухового апарату), що зазнали негативного
 впливу під час російської агресії на території Закарпатської області.

Одержані Брувером Д.Д. результати роботи збіглися з даними даного
 спостереження. З використанням запропонованих у роботі алгоритмів було
 здійснено контроль платформ для здійснення опитувань осіб з інвалідністю,
 спричинених непереборними впливами на соціальну сферу Закарпатської
 області.

Директор «Спортивно-реабілітаційного центру
 для осіб з інвалідністю та ветеранів війни»


Віктор ПЕТРОВ



Публікація наукової статті (журнал кат. Б)

Вісник Тернопільського національного технічного університету

https://doi.org/10.33108/visnyk_tntu

Scientific Journal of the Ternopil National Technical University

2023, № 1 (109) https://doi.org/10.33108/visnyk_tntu2023.01ISSN 2522-4433. Web: visnyk.tntu.edu.ua

UDC 004.415

**THE USE OF ABSTRACT MOORE AUTOMATON TO CONTROL THE
SENSORS OF A SERVICE-ORIENTED ALARM AND EMERGENCY
NOTIFICATION NETWORK**

Olha Kryazhych^{1,2}; Victoria Itskovych³; Kateryna Iushchenko³; Veronika Hrytsyshyna⁴; Danylo Bruvier⁴; Vyacheslav Nykytyuk²; Ihor Bodnarchuk²

¹*Institute of Telecommunications and Global Information Space of National Academy of Sciences of Ukraine, Kyiv, Ukraine*

²*Ternopil Ivan Puluj National Technical University, Ternopil, Ukraine*

³*Institute of Telecommunications and Global Information Space of National Academy of Sciences of Ukraine, Kyiv, Ukraine*

⁴*Volodymyr Dahl East Ukrainian National University, Severodonetsk, Ukraine*

Summary. The paper aims to achieve the ability of an abstract Moore automaton to control the sensors of a service-oriented alarm system for notification of emergency situations in a metropolis. An important procedure is minimization of the internal settings of the automaton, on the basis of which there will be a graph of transitions to the machine for which a synchronous trigger is turned on – a switch between sensors, - which allows not only to receive information, but also to confirm that it is necessary to notify about the current emergency situation. The relevance of those proves the need for security of living in the urban environment. Constant monitoring of the parameters of the medium is of high necessity. This allows to balance the key factors influencing the system in order to make a sound management decision. The possible ways of using an abstract Moore automaton to control the sensors of a service-oriented alarm system about emergency situations in a metropolis is analysed.

Key words: Aufenkamp-Honu algorithm, equivalence, trigger, synchronous transition, graph, signal, combination, parameter.

https://doi.org/10.33108/visnyk_tntu2023.01.111

Received 11.01.2023

The problem formulation. Urbanization not only excessively loads the environment but causes the problems bound with society functioning of in overpopulated cities. One of these issues is the safety of residence places. The problem lies in comprehensive study of parameters that influence the result and make it possible to balance the controlling influences on the system to obtain a considered management solution. This can be achieved through the development of a service-oriented network, which signals the possibility of an emergency situation, a threat to the environment and people, by means of installed sensors which constantly monitor the urban environment.

Analysis of available research results. British Standards Institution embraces all the mentioned by the term ‘Smart City’ [1]. In particular, this notion also denotes keeping the safety of human activity and environment by means of modern information communicative technologies [2]. At the same time, the energy efficiency of networks that transmit information about the state of the studied urban environment is of considerable importance [3]. That is why researchers [4–6] tend to use LoRaWaN technology to build service-oriented networks for notification of emergency situations and critical parameters within the framework of smart city technology [7]. The main problems to be solved in this case are creating of technologies for combining devices into a network and their management [8], as well as developing the models

of a complex system of the research object [9], where network devices are points-sources of input information [5].

Objective of the research is to study the possible use of the Moore abstract automaton to control the sensors of the service-oriented network of alarm and emergency notification in the metropolis. A well-balanced network device management system allows receiving information without interference, failures and noise, which ultimately leads to efficient network operation. At the same time, the procedure for minimizing the internal states of automaton is important, based on which the transition graph of the automaton is built, after which a synchronous trigger – a switch between sensors – is turned on. This allows not only to obtain data but also to confirm or deny the occurrence of an emergency.

Statement of the problem. This paper considers the task of sensor management of a service-oriented network. The sensors of such a network are of the same type according to the types of tasks. For example, CO₂ carbon dioxide sensors in the residential area of the city. There can be several dozens to hundreds of such network devices in just one district of the metropolis. In addition to sensors, executive devices, controllers and other objects that require control automation with minimization of repetition of the same type of actions are connected to the network.

The use of abstract automaton for algorithmization of processes in service-oriented network requires the division of all states of the original abstract automaton into classes of equivalent states that do not overlap, and the replacement of each equivalence class by one state. According to this approach, it is possible to describe a mathematical model of a system where there are many input actions of the same type, transitions between states that allow to isolate similar input data, and to draw a conclusion for each group of input results, taking into account the limitations caused by transitions between states [10, 11]. That is, a minimum of states and a given output function are obtained. The idea of using an abstract automaton to control the sensors of a service-oriented network arose according to [11].

The given problem can be solved by the Aufenkamp-Hohn algorithm [12] while minimizing the number of internal states of the Milli automaton $S = \{X, Y, A, \lambda, \delta, a_0\}$:

1. Find successive partitions $\pi_1, \pi_2, \dots, \pi_k, \pi_{k+1}$, of the set A into classes of one-, two-, ..., $K, (K + 1)$ -equivalent states until at $(K + 1)$ step it will turn out that $\pi_k = \pi_{k+1}$. In this case, the K -equivalent states are equivalent. The number of steps K , at which $\pi_k = \pi_{k+1}$, does not exceed $N-1$, where N is the number of internal states of the automaton.

2. In each equivalence class π , one representative of the class is selected, which form sets A' of the states of the minimal automaton S' .

3. The function of transitions δ' and outputs λ' of the automaton S' is determined on the set $A' \times X$. For this, in the table of transitions and exits, the columns corresponding to the states included in the set A' are crossed out. The remaining columns of the transition table change their state to the equivalent one from the set A' to the corresponding elements.

4. One of the states equivalent to state a_0 is selected as a'_0 .

In the minimization of the Moore automaton, the concept of 0-equivalence of states and the division of the set of states into 0-classes is introduced: any states of the Moore automaton, marked by the same output signals, are called 0-equivalent. For example, minimization of the Moore automaton can be given as follows (Table 1).

Table 1

Table of transitions and outputs of the Moore automaton

Y	y1	y1	y3	y3	y3	y2	y3	y1	y2	y2	y2	y2
A	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12
x1	a10	a12	a5	a7	a3	a7	a3	a10	a7	a1	a5	a2
x2	a5	a7	a6	a11	a9	a11	a6	a4	a6	a8	a9	a8

Then the partition of π_0 will be as follows:

$$\pi_0 = \{B_1, B_2, B_3\};$$

$$B_1 = \{a_1, a_2, a_8\}, B_2 = \{a_6, a_9, a_{10}, a_{11}, a_{12}\}, B_3 = \{a_3, a_4, a_5, a_7\}.$$

Accordingly, the partition by π_0 can be presented in the Table 2:

Table 2

Table of partition by π_0

Y	B ₁			B ₂					B ₃			
A	a ₁	a ₂	a ₈	a ₆	a ₉	a ₁₀	a ₁₁	a ₁₂	a ₃	a ₄	a ₅	a ₇
x ₁	B ₂	B ₂	B ₂	B ₃	B ₃	B ₁	B ₃	B ₁	B ₃	B ₃	B ₃	B ₃
x ₂	B ₃	B ₃	B ₃	B ₂	B ₂	B ₁	B ₂	B ₁	B ₂	B ₂	B ₂	B ₂

According to the given algorithm, π_1 is split:

$$\pi_1 = \{C_1, C_2, C_3, C_4\};$$

$$C_1 = \{a_1, a_2, a_8\}, C_2 = \{a_6, a_9, a_{11}\}, C_3 = \{a_{10}, a_{12}\}, C_4 = \{a_3, a_4, a_5, a_7\},$$

with the corresponding partition table (Table 3).

Table 3

Table of partition by π_1

Y	C ₁			C ₂			C ₃		C ₄			
A	a ₁	a ₂	a ₈	a ₆	a ₉	a ₁₁	a ₁₀	a ₁₂	a ₃	a ₄	a ₅	a ₇
x ₁	C ₃	C ₃	C ₃	C ₄	C ₄	C ₄	C ₁	C ₁	C ₄	C ₄	C ₄	C ₄
x ₂	C ₄	C ₄	C ₄	C ₂	C ₂	C ₂	C ₁	C ₁	C ₂	C ₂	C ₂	C ₂

If perform the partition of π_2 , the results will be obtained:

$$\pi_2 = \{D_1, D_2, D_3, D_4\};$$

$$D_1 = \{a_1, a_2, a_8\}, D_2 = \{a_6, a_9, a_{11}\}, D_3 = \{a_{10}, a_{12}\}, D_4 = \{a_3, a_4, a_5, a_7\},$$

indicating the completion of the splitting procedure, since the π_2 split repeats the π_1 split. After that, it is sufficient to choose arbitrarily one representative from each equivalence class D_1, D_2, D_3, D_4 – in this case, by the minimum number: $A = \{a_1, a_3, a_6, a_{10}\}$. Then, by removing ‘redundant’ states from the original table of transitions, the minimum Moore automaton can be determined (Table 4).

Table 4

The minimal Moore automaton obtained according to the Aufenkamp-Hohn algorithm

Y	y ₁	y ₃	y ₂	y ₂
A	a ₁	a ₃	a ₆	a ₁₀
x ₁	a ₁₀	a ₃	a ₃	a ₁
x ₂	a ₃	a ₆	a ₆	a ₁

Considering [12], it should be noted that the equivalence of states can be proved using iterative methods, but there may be restrictions on the input values. But for this study, this limitation is not significant, since the states of sensors of the system are clearly defined [13].

Note that, taking into account [14], it would be illogical to stop only at the specified one (Table 4) to solve the given problem, since further decisions are transferred to the plane of human management. And this can cause problems of obtaining relevant data during optimization [15]. That is why it is worth setting a trigger that will direct the sensors to perform actions according to a certain protocol.

For a trigger-type task, a shortened transition table with two information inputs and a full transition table of an asynchronous trigger with two information inputs X and Y are built in practice.

When constructing transition tables of a synchronous trigger, it should be borne in mind that when $C = 0$, the internal state of the trigger does not change regardless of the states of inputs X and Y, i.e. $Q(t+1) = Q(t)$, and when $C = 1$, the synchronous trigger functions as a corresponding asynchronous [16]. Taking this into account, it is possible to obtain both abbreviated and full transition tables of a synchronous trigger.

It should be mentioned that the study of the synchronous trigger was considered in [17], where the prospects of using digital automata in modern systems and networks were pointed out, and it was also noted that a similar task requires multi-faceted analysis in different directions.

Analysis of numerical results. We minimize the abstract Moore automaton designed to control the sensors responsible for monitoring the state of one object in the service-oriented network of emergency and emergency notification in the metropolis and specified by the table of transitions and outputs (Table 5).

Table 5

Table of transitions and outputs

Y	y ₁	y ₂	y ₃	y ₄	y ₁	y ₂	y ₄	y ₂	y ₁	y ₁
A	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇	a ₈	a ₉	a ₁₀
X	a ₁	a ₃	a ₆	a ₄	a ₇	a ₅	a ₄	a ₉	a ₇	a ₁₀
x ₁	a ₁	a ₃	a ₆	a ₄	a ₇	a ₅	a ₄	a ₉	a ₇	a ₁₀
x ₂	a ₅	a ₁	a ₃	a ₆	a ₁₀	a ₉	a ₈	a ₁₀	a ₉	a ₅
x ₃	a ₇	a ₄	a ₅	a ₂	a ₈	a ₈	a ₁	a ₃	a ₁₀	a ₇

We partition $\pi_0 = \{B_1, B_2, B_3, B_4\}$. For the Moore automaton π_0 is partitioned by the input signal:

$$B_1 = \{a_1, a_5, a_9, a_{10}\}, B_2 = \{a_2, a_6, a_8\}, B_3 = \{a_3\}, B_4 = \{a_4, a_7\}.$$

Construct the π_0 partition table (Table 6).

Table 6

 π_0 partition table

	B ₁				B ₂			B ₃	B ₄	
A	a ₁	a ₅	a ₉	a ₁₀	a ₂	a ₆	a ₈	a ₃	a ₄	a ₇
x ₁	B ₁	B ₄	B ₄	B ₁	B ₃	B ₁	B ₁	B ₂	B ₄	B ₄
x ₂	B ₁	B ₁	B ₁	B ₁	B ₁	B ₁	B ₁	B ₃	B ₂	B ₂
x ₃	B ₄	B ₂	B ₁	B ₄	B ₄	B ₂	B ₃	B ₁	B ₂	B ₁

Olha Kryazhych, Victoria Itskovych, Kateryna Iushchenko,
Veronika Hrytsyshyna, Danylo Bruvier, Vyacheslav Nykytyuk, Ihor Bodnarchuk

According to π_0 partition table (Table 6), the partition of π_1 is performed. When performing this partition, the analysis is carried out only within each separate set of B_i :

$$\pi_1 = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9\}$$

$$C_1 = \{a_1, a_{10}\}, C_2 = \{a_5\}, C_3 = \{a_9\}, C_4 = \{a_2\}, C_5 = \{a_6\}, C_6 = \{a_8\}, C_7 = \{a_3\}, C_8 = \{a_4\}, C_9 = \{a_7\}.$$

Due to the obtained results, the table of partition π_1 is constructed (Table 7)

Table 7

Table of partition of π_1

	C ₁		C ₂	C ₃	C ₄	C ₅	C ₆	C ₇	C ₈	C ₉
A	a ₁	a ₁₀	a ₅	a ₉	a ₂	a ₆	a ₈	a ₃	a ₄	a ₇
x ₁	C ₁	C ₁	C ₉	C ₉	C ₇	C ₂	C ₃	C ₅	C ₈	C ₈
x ₂	C ₂	C ₂	C ₁	C ₃	C ₁	C ₃	C ₁	C ₇	C ₅	C ₆
x ₃	C ₉	C ₉	C ₆	C ₁	C ₈	C ₆	C ₇	C ₂	C ₄	C ₁

On the basis of the above mentioned table of partition π_1 , the partition of π_2 is performed:

$$\pi_2 = \{D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9\}.$$

$$D_1 = \{a_1, a_{10}\}, D_2 = \{a_5\}, D_3 = \{a_9\}, D_4 = \{a_2\}, D_5 = \{a_6\}, D_6 = \{a_8\}, D_7 = \{a_3\}, D_8 = \{a_4\}, D_9 = \{a_7\},$$

which ultimately repeats the partition of π_1 .

That is, the partitioning procedure can be completed, which makes it possible to create a combined table of transitions with the construction of a graph of transitions and to consider the combinations of the trigger, which will allow transferring the sensors to two stable states (action/stop) and make transitions between them (receiving information, transmitting information, checking own state, saving previous information for control in the cell). These transitions must be performed by all sensors in a certain sequence to ensure the relevance of the information received and transmitted by the service-oriented network.

Results of the research. From each equivalence class, we select one representative of this class. We choose a_1 from the set $D_1 = \{a_1, a_{10}\}$, therefore $a_1 \equiv a_{10}$. In the transitions table, we cross out the column that corresponds to the state of a_{10} , and in the rest of the table, we replace a_{10} with a_1 . We obtain a combined table of transitions and outputs of the minimal automaton (Table 8).

Table 8

Combined table of transitions and outputs of the minimal automaton

Y	y ₁	y ₂	y ₃	y ₄	y ₁	y ₂	y ₄	y ₂	y ₁
A	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇	a ₈	a ₉
X									
x ₁	a ₁	a ₃	a ₆	a ₄	a ₇	a ₅	a ₄	a ₉	a ₇
x ₂	a ₅	a ₁	a ₃	a ₆	a ₁	a ₉	a ₈	a ₁	a ₉
x ₃	a ₇	a ₄	a ₅	a ₂	a ₈	a ₈	a ₁	a ₃	a ₁

Based on Table 8, it is possible to construct the transition graph of the Moore automaton (Fig. 1).

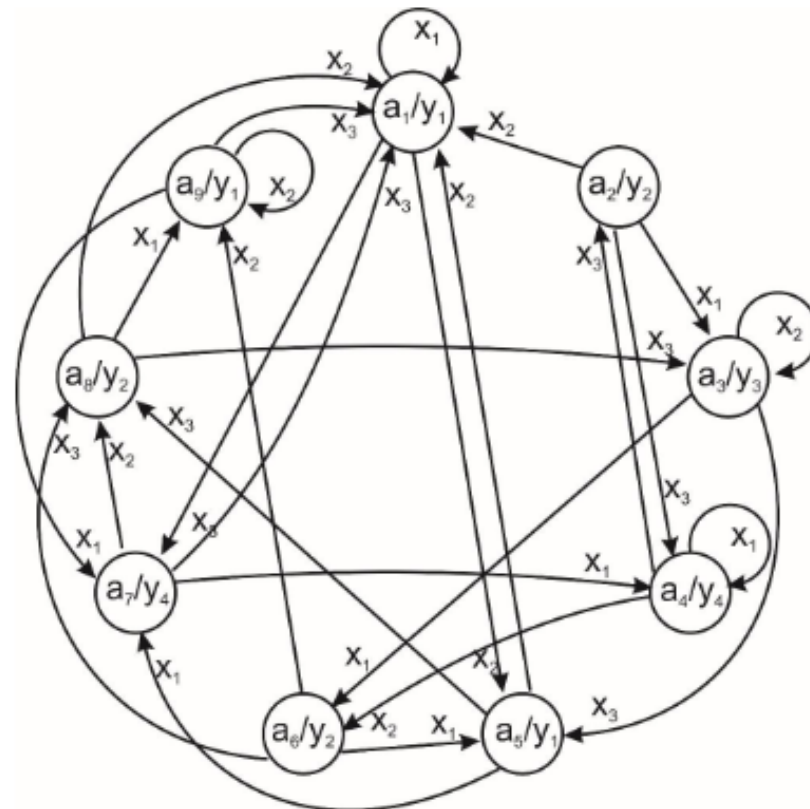


Figure 1. Transition graph of a Moore automaton

The automatic trigger has four possible options for transitions: «0-0», «0-1», «1-0», «1-1». Transitions table (Table 9) and signal combinations table (Table 10) were used to create the network test model.

Table 9

Experimental data of transitions

a) «0-0»

t		(t+1)
X	Y	Q
0	0	0
0	1	$\overline{Q(t)}$
1	0	$\overline{Q(t)}$
1	1	1

b) «0-1»

t			t+1
X	Y	Q	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Olha Kryzhych, Victoria Itskovych, Kateryna Iushchenko,
Veronika Hrytshynna, Danylo Bruvier, Vyacheslav Nykytyuk, Ihor Bodnarchuk

The end of the table

t			t+1
C	X	Y	Q
0	0	0	Q(t)
0	0	1	Q(t)
0	1	0	Q(t)
0	1	1	Q(t)
1	0	0	0
1	0	1	Q(t)
1	1	0	Q(t)
1	1	1	1

t				t+1
C	X	Y	Q	Q
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Table 10

Combination of signals

Q(t)	Q(t+1)	X	Y
0	0	0	0
		0	1
0	1	1	0
		1	1
1	0	0	0
		1	0
1	1	0	1
		1	1

It can be noted that according to the transitions table (Table 9, b) there are correspondences to combinations of X and Y signals (Table 10).

Thus:

1. For the transition «0-0» X = 0, Y can be equal to 0 or 1.
2. For the transition «0-1» X = 1, Y can be equal to 0 or 1.
3. For the «1-0» transition, X can be equal to 0 or 1, and Y = 0.
4. For the «1-1» transition, X can be equal to 0 or 1, and Y = 1.

Then the transition matrix of the trigger can be written as follows:

0-0	0	b1
0-1	1	b2
1-0	b3	0
1-1	b4	1

where:

b_1, b_2, b_3, b_4 are random signals (0 or 1).

Determining of interdependence between the input variables of the trigger is a mandatory condition that provides the possibility of maximum simplification of circuits with memory, which ultimately also contributes to the energy efficiency of the network.

The obtained results can also be used in the implementation of 3D resumes, which allow personnel selection in real time with transitions between task events depending on the completed/uncompleted previous task. In this case, the trigger type can be RS, and to meet the conditions of the 3D resume, transitions will be made when a unit is submitted to the output upon successful completion of the task with a transition to another stage, or zero in the case of an incorrect response with a transition to questions at a lower level or exit from the testing service.

Conclusions. Within the research conducted by Academician Viktor Glushkov, it was noted about the widespread use of digital automata in various areas of business. Modern informatization and digitalization of society requires new or fundamentally improved approaches to the automation of operations and process management.

The paper analyses the possibility of using Moore's abstract automaton to control the sensors of the service-oriented network of emergency and emergency notification in the metropolis. It has been established that the minimization of the Moore automaton can be solved using the Aufenkamp-Hohn algorithm, which allows to isolate the equivalent state of the automaton: choose arbitrarily from each equivalence class one representative with the minimum number. After that, it is sufficient to remove from the original table of transitions other states that are superfluous for this case and determine the minimum Moore automaton. Then, the matrix of transitions of the trigger is formed, which, in this case, is not only just an action counter, but also triggers the action of the next sensor depending on what the input signal was.

The developed approach and the results obtained from the implementation of the test service-oriented network of notification of emergency and emergency situations in the metropolis allow the use of an abstract Moore automaton to automate the process of surveying sensors with verification of the received information from the source.

The obtained results can also be used in the implementation of 3D resumes, which allow personnel selection in real time with transitions between task events depending on the completed/uncompleted previous task.

References

1. British Standards Institution. URL: <https://www.bsigroup.com/>.
2. Batty M. et al. 2012. Smart Cities of the Future. *European Physical Journal ST*. 214: 481–518. Doi:10.1140/epjst/e2012-01703-3. <https://doi.org/10.1140/epjst/e2012-01703-3>
3. Maksymov A., Vakhovych I., Hutnichenko T., Babicheva P., Vakulenko N., Iholnikova N., Tsyfra T., Molodid O., Molodid O., Bielienskova O., Iachmenova Yu., Doroshuk Yu., Skrypnyk A., Vakoliuk A., Boiko V., Sehedii M., Vakhovych D. Enerhoefektyvnist v munitsypalnomu sektori. Navchalnyi posibnyk dlia posadovykh osib mistsevoho samovriaduvannya. Asotsiatsiia mist Ukrainy. K.: TOV "Pidpriemstvo "VI EN EI", 2015. 184 p. [In Ukrainian].
4. Raza U., Kulkarni P., Sooriyabandara M., Low power wide area networks: an overview, *IEEE Commun. Surv. Tutor.* 19 (2). 2017. P. 855–873. <https://doi.org/10.1109/COMST.2017.2652320>
5. LoRaWAN Specification. v1.1, LoRa Alliance/ 2017. URL: <https://lorawan-specification-v1-1/>
6. Ploennigs J., Cohn J., Stanford-Clark A., The future of IoT, *IEEE Internet Things Mag.* 1 (1). 2018. P. 28–33. URL: <http://dx.doi.org/10.1109/IOTM.2018.1700021>.

Olha Kryazhych, Victoria Itskovych, Kateryna Iushchenko,
Veronika Hrytsyshyna, Danylo Bruvier, Vyacheslav Nykytyuk, Ihor Bodnarchuk

7. Shepard, Mark 2011. *Sentient City: Ubiquitous Computing, Architecture, and the Future of Urban Space*. New York City. Architectural League of New York. The Architect's Newspaper. ISBN 978-0262515863.
8. Zghurovskiy M. Z., Pankratova N. D. *Systemnyi analiz. Problemy, metodolohiia zastosuvannia*. K.: "Naukova dumka", 2011. 728 p. [In Ukrainian].
9. Gill A. *Vvedenie v teoriyu konechnikh avtomatov*. M.: Nauka, 1966. 272 p. [In Russian].
10. Glushkov V. M. *Abstraktnaya teoriya avtomatov*, UMN, 16:5(101) (1961), 3–62; *Russian Math. Surveys*, 16:5 (1961), 1–53. [In Russian]. <https://doi.org/10.1070/RM1961v016n05ABEH004112>
11. Aufenkamp D. D., Khon F. Ye., *Analiz posledovatel'nostnykh mashin*. *Matematika*, 3:3 (1959), 129–148; *IRE Trans.*, 6 (1957), 276–285. [In Russian].
12. Glushkov V. M. *Sintez tsifrovikh avtomatov*. M.: GIFML, 1962. 476 p. [In Russian].
13. Butsiy R., Lupenko S. Comparative analysis of neurointerface technologies for the problem of their reasonable choice in human-machine information systems. *Scientific Journal of TNTU*. 2020. Vol. 100. No. 4. P. 135–148. https://doi.org/10.33108/visnyk_tntu2020.04.135
14. Yemets V. Technological systems investigation machines tools with parallel structure kinematic. *Scientific Journal of TNTU*. 2021. Vol. 102. No. 2. P. 37–44. https://doi.org/10.33108/visnyk_tntu2021.02.037
15. Mano, M. Morris; Kime, Charles R. 2004. *Logic and Computer Design Fundamentals*, 3rd Edition. Upper Saddle River, NJ, USA: Pearson Education International. P. 283. ISBN 0-13-191165-1.
16. Lukac M., Kameyam M., Perkowski M. *Quantum Finite State Machines – a Circuit Based Approach*. *International Journal of Unconventional Computing*. 2013. Vol. 9. Issue 3/4. P. 267–301. 35 p.
17. Elshabrawy T., Robert J. Interleaved chirp spreading lora-based modulation, *IEEE Internet Things J.* 6 (2). 2019. P. 3855–3863. <https://doi.org/10.1109/IIOT.2019.2892294>

Список використаних джерел

1. British Standards Institution. URL: <https://www.bsigroup.com/>.
2. Batty M. et al. 2012. Smart Cities of the Future. *European Physical Journal ST*. 214: 481–518. Doi: 10.1140/epjst/e2012-01703-3. <https://doi.org/10.1140/epjst/e2012-01703-3>
3. Максимов А., Вахович І., Гутніченко Т., Бабічева П., Вакуленко Н., Ігольнікова Н., Цифра Т., Молодіт О., Молодіт О., Беленкова О., Ячменьова Ю., Дорошук Ю., Скрипник А., Ваколюк А., Бойко В., Сегедій М., Вахович Д. *Енергоефективність в муніципальному секторі. Навчальний посібник для посадових осіб місцевого самоврядування*. Асоціація міст України. К.: ТОВ «Підприємство «ВІ ЕН ЕЙ», 2015. 184 с.
4. Raza U., Kulkarni P., Sooriyabandara M., Low power wide area networks: an overview, *IEEE Commun. Surv. Tutor.* 19 (2). 2017. P. 855–873. URL: <http://dx.doi.org/10.1109/COMST.2017.2652320>.
5. LoRaWAN Specification. v1.1, LoRa Alliance/ 2017. URL: https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/.
6. Ploennigs J., Cohn J., Stanford-Clark A., The future of IoT, *IEEE Internet Things Mag.* 1 (1). 2018. P. 28–33. URL: <http://dx.doi.org/10.1109/IOTM.2018.1700021>.
7. Shepard, Mark 2011. *Sentient City: Ubiquitous Computing, Architecture, and the Future of Urban Space*. New York City. Architectural League of New York. The Architect's Newspaper. ISBN 978-0262515863.
8. Згуровський М. З., Панкратова Н. Д. *Системний аналіз. Проблеми, методологія застосування*. К.: «Наукова думка», 2011. 728 с.
9. Гилл А. *Введение в теорию конечных автоматов*. М.: Наука, 1966. 272 с.
10. Глушков В. М. *Абстрактная теория автоматов*, УМН, 16:5(101) (1961), 3–62; *Russian Math. Surveys*, 16:5 (1961), 1–53. <https://doi.org/10.1070/RM1961v016n05ABEH004112>
11. Ауфенкамп Д. Д., Хон Ф. Е., *Анализ последовательностных машин*. *Математика*, 3:3 (1959), 129–148; *IRE Trans.*, 6 (1957), 276–285.
12. Глушков В. М. *Синтез цифровых автоматов*. М.: GIFML, 1962. 476 с.
13. Butsiy R., Lupenko S. Comparative analysis of neurointerface technologies for the problem of their reasonable choice in human-machine information systems. *Scientific Journal of TNTU*. 2020. Vol. 100. No. 4. P. 135–148. https://doi.org/10.33108/visnyk_tntu2020.04.135
14. Yemets V. Technological systems investigation machines tools with parallel structure kinematics. *Scientific Journal of TNTU*. Vol. 102. No. 2. P. 37–44. https://doi.org/10.33108/visnyk_tntu2021.02.037
15. Mano, M. Morris; Kime, Charles R. 2004. *Logic and Computer Design Fundamentals*, 3rd Edition. Upper Saddle River, NJ, USA: Pearson Education International. P. 283. ISBN 0-13-191165-1.
16. Lukac M., Kameyam M., Perkowski M. *Quantum Finite State Machines – a Circuit Based Approach*. *International Journal of Unconventional Computing*. 2013. Vol. 9. Issue 3/4. P. 267–301. 35 p.
17. Elshabrawy T., Robert J. Interleaved chirp spreading lora-based modulation, *IEEE Internet Things J.* 6 (2). 2019. P. 3855–3863. <https://doi.org/10.1109/IIOT.2019.2892294>

УДК 004.415

ВИКОРИСТАННЯ АБСТРАКТНОГО АВТОМАТА МУРА ДЛЯ УПРАВЛІННЯ ДАТЧИКАМИ СЕРВІС-ОРІЄНТОВАНОЇ МЕРЕЖІ ОПОВІЩЕННЯ ПРО НАДЗВИЧАЙНІ ТА АВАРІЙНІ СИТУАЦІЇ

Ольга Кряжич^{1,2}; Вікторія Іцкович³; Катерина Ющенко³;
Вероніка Грицишина⁴; Данило Брувер⁴; Вячеслав Никитюк²;
Ігор Боднарчук²

¹Інститут телекомунікацій і глобального інформаційного простору НАН України, Київ, Україна

²Тернопільський національний технічний університет імені Івана Пулюя, Тернопіль, Україна

³Інститут телекомунікацій і глобального інформаційного простору НАН України, Київ, Україна

⁴Східноукраїнський національний університет імені Володимира Даля, Сєвєродонецьк, Україна

Резюме. Метою роботи є дослідження можливості використання абстрактного автомата Мура для управління датчиками сервіс-орієнтованої мережі оповіщення про надзвичайні та аварійні ситуації в мегаполісі. Важливою є процедура мінімізації внутрішніх станів автомата, на основі чого будується граф переходів автомата, за яким вмикається синхронний тригер – перемикач між датчиками, – що дозволяє не просто отримати інформацію, а й підтвердити чи спростувати настання екстреної ситуації. Актуальність теми підтверджується необхідністю забезпечення безпеки проживання в урбаністичному середовищі. Необхідним є постійний моніторинг параметрів середовища, які дозволяють збалансувати керуючі впливи на систему для отримання зваженого управлінського рішення. Проаналізовано можливість використання абстрактного автомата Мура для управління датчиками сервіс-орієнтованої мережі оповіщення про надзвичайні та аварійні ситуації в мегаполісі. Встановлено, що проведення мінімізації автомата Мура можна вирішити за алгоритмом Ауфенкампа-Хоу, що дозволяє вичленювати еквівалентний стан автомата: вибрати довільно з кожного класу еквівалентності по одному представнику за мінімальним номером. Після цього достатньо видалити з вихідної таблиці переходів інші, зайві для даного випадку, стани та визначити мінімальний автомат Мура. Після цього формується матриця переходів тригера, яка, у даному випадку, не лише є просто лічильником дій, а й запускає в дію чергового датчика залежно від того, який був вихідний сигнал. Розвинений підхід та отримані результати з реалізації у тестовій сервіс-орієнтованій мережі оповіщення про надзвичайні та аварійні ситуації в мегаполісі дозволяють використовувати абстрактний автомат Мура для автоматизації процесу опитування датчиків з перевіркою отриманої інформації від джерела. Отримані результати можуть бути також використані при реалізації 3D резюме, які дозволяють здійснювати відбір персоналу в режимі реального часу з переходами між подіями-завданнями залежно від виконаного/невиконаного попереднього завдання.

Ключові слова: алгоритм Ауфенкампа-Хоу, еквівалентність, тригер, синхронний перехід, граф, сигнал, комбінація, параметр.

https://doi.org/10.33108/visnyk_tntu2023.01.111

Отримано 11.01.2023

Додаток В

Лістинг формування базового словника тестової системи

(фрагмент)

```

unit mylib;
interface
uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, Math;
  function step(val: Real; st: Integer): Extended;
//сортування БД z..a
procedure quicks_c(first:Integer;last:Integer;count:integer);
//сортування масиву start-end
procedure quicks_n_records(first:Integer;last:Integer;count:integer);
function
find_index_first(onkvo_razr:integer;slovo:String;indfirstlast:integer):In
t64;
//Сортування словника a..z
procedure quicks_c_az(first:Integer;last:Integer;count:integer);
//Создание словаря
procedure cre_diction_abc_pr();
procedure cre_diction_pr();
//Процедура пошуку
function
find_c_pr(first:integer;last:integer;n:String;kvo_razr:integer):Integer;
procedure get_mem_pr();
implementation
uses  UnForm,UnForm1,Unit3;

function step(val: Real; st: Integer): Extended;
//Обчислення кроку
var
  i: Integer;
  s: Extended;
begin
  s:=1.0;
  for i := 1 to st do
    s:=s*val;
  Result := s;
end;
procedure quicks_c(first:Integer;last:Integer;count:integer);
//Сортування проміжної вибірки z..a
var
  i,j,l,a:Integer;
  x,buf:String;
begin
  a:=0;
  i:= first; //Перший елемент масиву
  j:= last; //Останній елемент масиву
  x:=array_words[(first+last) div 2]; //Визначити ...
  repeat
    while array_words[i]>x do i:=i+1;
    while x>array_words[j] do j:=j-1;

```

```

//збільшити число вибірки
a:=a+1;
if i<=j then
  begin          //обміняти місцями питання
    buf:=array_words[i];
    array_words[i]:=array_words[j];
    array_words[j]:=buf;
    i:=i+1;
    j:=j-1;
    // for l:=1 to count do write(' ',m_c[l]);
    // showmessage('ітерація '+IntToStr(a));
  end;
until i>j;
//showmessage('i='+IntToStr(i));
//showmessage('j='+IntToStr(j));
// quicks
if first<j then quicks_c(first,j,count);
// quicks
if i<last then quicks_c(i,last,count);
end;

```

(далі йде цикли вибору, так само з сортуванням, як і попереднє)

Процедура формування вибірки

```

//Очистити вибірки
array_words:=nil; //словник
array_veroyat:=nil; //шукане слово
array_veroyat_cur:=nil; //пошук слова
if FileExists(InitDir+trim(dic_nam_frag)+'.txt')=False then begin
  MessageDlg('Відсутній словник з таким ім'ям!',mtError,[mbOK],0);
  exit;
end;
//Прочитати словник у масив
try
  AssignFile(mFile1, InitDir+trim(dic_nam_frag)+'.txt');
  Reset(mFile1);
  //Прочитати 1-й рядок словника
  Readln(mFile1,kvo_slov_in_dic);
  SetLength(array_words, kvo_slov_in_dic+1);
  //Прочитати другий рядок...

```

(далі циклом)

Кінцеве формування множин питань для роботи системи

```

rabstr:=floattostr(AOH);
j:= round(AOH/1);
if j<1 then j:=1;
if j>kvo_slov_in_dic then j:=kvo_slov_in_dic;

if j>kvo_slov_in_dic then j:=kvo_slov_in_dic;
if j<0 then j:=0;
curslovo_j:=array_words[j];
//Визначити для curslovo_j
AOHj:=0;
for i:=1 to kvo_razrad do begin
  curpos:=pos(curslovo_j[i],abc)-1;
  AOH1:=AOH1 + round(curpos*step(q,kvo_razrad-i));

```

```

end;
x1:=AOH-AOH1;
if x1=0 then begin
  res:=j;
  result:=j;
end;
if x1<0 then begin
j:=j - round(abs(x1)/1);
end;
if x1>0 then begin
  j:=j + round(abs(x1)/1);
end;
if j>kvo_slov_in_dic then j:=kvo_slov_in_dic;
if j<1 then j:=1;
  //Визначити для curslovo_j
AOHj:=0;
curslovo_j:=array_words[j];
for i:=1 to kvo_razrad do begin
  curpos:=pos(curslovo_j[i],abc)-1;
  AOHj:=AOHj + curpos*step(q,kvo_razrad-i);
end;
xj:=AOH-AOHj;
if xj>0 then ind1:=1
else ind1:=-1;
ind:=0;
while ind=0 do begin
  curslovo_j:=array_words[j];
  //Визначити для curslovo_j
AOHj:=0;
for i:=1 to kvo_razrad do begin
  curpos:=pos(curslovo_j[i],abc)-1;
  AOHj:=AOHj + curpos*step(q,kvo_razrad-i);
end;
xj:=AOH-AOHj;
if xj=0 then begin
  if j>kvo_slov_in_dic then j:=kvo_slov_in_dic;
  if j<1 then j:=1;
  res:=j;
  ind:=111;
  result:=j;
end;
if xj<0 then begin
  j:=j-1;
  if j>kvo_slov_in_dic then begin
    j:=kvo_slov_in_dic;
    ind:=111;
  end;
  if j<1 then begin
    j:=1;
    ind:=111;
  end;
end;
if xj>0 then begin
  j:=j+1;
  if j>kvo_slov_in_dic then begin
    j:=kvo_slov_in_dic;
    ind:=111;
  end;
  if j<1 then begin

```

```
        j:=1;
        ind:=111;
    end;
end;
if (((ind1>0) and (xj<0)) or ((ind1<0) and (xj>0))) then ind:=111;
//Заміна питання
end;
if j>kvo_slov_in_dic then j:=kvo_slov_in_dic;
if j<1 then j:=1;
res:=j;
result:=j;
except
on Exception do begin
    MessageDlg('Помилка при формуванні словника!',mtError,[mbOK],0);
    Application.Terminate;
end;
end;//try
end;
end.
```

**Лістинг коду реалізації наведеного алгоритму формування
множини питань при тестуванні у вигляді фреймового дерева**

```

procedure TZapr_f.ShowTreeView(Sender: TObject);
var
  rr, rr1, rr2, rr3, rr4: TTreenode;
rs: ttreenodes;
  n, n1, n2, id_query, i_beg, i_f, ii, ij, ind : integer;
  st, s, srt_name, id_filter, sl, kod : string;
im : TImageIndex;
begin
  TreeView1.Items.Clear;
  rr:=nil;
  rr1:=nil;
  rr2:=nil;
  ADOQuery1.First;
  //////////Формування множини питань//////////
  n:=0;
  n:=ADOQuery1.RecordCount;
  SetLength (Mtvgl, n);
  while not ADOQuery1.Eof do
    BEGIN
  id_filter:=ADOQuery1objt.AsString;
  SelectColl(id_filter);
  n2:=Query3.RecordCount-1;
  n:= n+n2;
  ADOQuery1.Next;
  end;
  SetLength (Mtv, n, n);
  ADOQuery1.First;
  while not ADOQuery1.Eof do
    BEGIN
  id_query:=ADOQuery1objt.AsInteger;

```

```

        srt_name:= ADOQuery1.StringField.AsString;
        rr1:=TreeView1.Items.add(rr,srt_name);
        rr1.ImageIndex:=0;
ind:= rr1.AbsoluteIndex;
mtvgl[ind]:=inttostr(id_query);
        ADOQuery1.Next;
        END;
ind:= 0;
        ADOQuery1.First;
        if rr1 <> nil then
begin
        rr:=TreeView1.TopItem;
        rr1.ImageIndex:=0;
        rr.ImageIndex:=0;
        end;
n2:= 0;
        while rr<>nil do
        BEGIN
        ind:= rr.AbsoluteIndex;
        id_filter := mtvgl[n2];
        SelectColl(id_filter);
        rr1.ImageIndex:=0;
        rr.ImageIndex:=0;
        if Query3VKOD.AsString <> '' then kod := Query3VKOD.AsString;
mtv[ind,0] := kod; // kod View по дереву
                while not Query3.Eof do
                Begin
                if lowercase(copy(Query3StringField.AsString,1,2)) <> 'id' then
begin
                n1:=n1+1;
                id_query:= Query3OBJT.AsInteger;
                srt_name:= Query3StringField.AsString;
                rr1:=TreeView1.Items.addChild(rr,srt_name);
                rr1.SelectedIndex:= 2;
                rr.SelectedIndex:= 2;
                kod := Query3VKOD.AsString;

```



```
    ind:= rr1.AbsoluteIndex;
mtv[ind,1] := inttostr(id_query);
end;
    Query3.Next;
        End;
        rr:= rr.getNextSibling;
        n2:=n2+1;
    end;
rr:=TreeView1.TopItem;
    while rr<>nil do
        BEGIN
rr.StateIndex:= 1;
rr:= rr.getNext;
        end;
rs:=TreeView1.Items;
rr:= rs.GetFirstNode;
        TreeView1.Select(rr);
        TreeView1.SetFocus;
    end;
```