

Міністерство освіти і науки України
Харківський національний економічний університет імені Семена Кузнеця

СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ І СИСТЕМИ

Монографія

під ред. д.е.н., проф. Пономаренка В.С.

Харків, 2021

УДК 004.891.2

Авторський колектив: к.ф.-м.н., доц. В. П. Бурдаєв – глава 1; д.т.н., проф. Н. Г. Аксак, к.т.н. М. В. Кушнарєв – глава 2; к.т.н., доц. М. Ю. Лосєв – глава 3; д.т.н., проф. О. Г. Руденко, д.т.н., проф. О. О. Безсонов – глава 4; д.т.н., проф. М. М. Корабльов, к.т.н., ст. викл. О. О. Фомічов – глава 5; д.т.н., проф. С.Г. Удовенко, к.т.н., доц. Л. Е. Чала – глава 6; к.т.н., доц. Федорченко В.М. (33.3%), Гапон А.О. (33.3%), Поляков А.О. (33.3%) – глава 7; к.е.н., доц. І. О. Ушакова – глава 8; к.т.н., доц. О. В. Фролов – глава 9; д.т.н., проф. Гордєєв А.С. – глава 10.

Рецензенти:

Годлевський М. Д. – доктор технічних наук, професор, завідувач кафедри програмної інженерії та інформаційних технологій управління Національного технічного університету "Харківський політехнічний інститут";

Петров К. Е. – доктор технічних наук, професор, завідувач кафедри інформаційних управляючих систем Харківського національного університету радіоелектроніки;

Юхов О. Ю. – доктор технічних наук, доцент, начальник кафедри військового зв'язку та інформатизації Національної академії Національної гвардії України.

Рекомендовано до видання рішенням ученої ради Харківського національного економічного університету імені Семена Кузнеця (протокол № 4 від 31.05.2021 р.)

І 76 **Сучасні інформаційні технології і системи:** монографія / за заг. ред. В. С. Пономаренка. - Х. : Видавництво «Стиль-іздат», 2021. - 182 с .

В монографії розглянуті сучасний стан та перспективи розвитку сучасних інформаційних технологій і систем різних видів і різного прикладного характеру. Монографія представляє інтерес як для фахівців, сфера діяльності яких безпосередньо пов'язана з розробкою прикладних інформаційних технологій і систем, так і для більш широкого кола фахівців. Вона буде корисною викладачам, аспірантам і студентам, що спеціалізуються в області інформаційних технологій, і всім, хто серйозно цікавиться проблемами інформаційного суспільства.

За достовірність викладених фактів, цитат та інших відомостей відповідальність несе автор.

Колектив авторів , 2021

ЗМІСТ

ВСТУП	4
ГЛАВА 1 ІНТЕГРУВАННЯ ЧАТ-БОТА @ES_ECONOMY_KARKAS_VOT З ЕКСПЕРТНОЮ СИСТЕМОЮ ДЛЯ ВИЗНАЧЕННЯ КЛАСУ КРЕДИТОСПРОМОЖНОСТІ ПОЗИЧАЛЬНИКА	7
ГЛАВА 2. МУЛЬТИАГЕНТНА СИСТЕМА РОЗПОДІЛЕНОЇ ТОРГОВОЇ ФІРМИ	24
ГЛАВА 3. СТВОРЕННЯ ПРИСТРОЇВ КОНТРОЛЮ І ЛОКАЛІЗАЦІЇ ПОМИЛОК В РОЗПОДІЛЕНИХ СИСТЕМАХ	40
ГЛАВА 4. РЕГУЛЯРИЗОВАНИЙ РЕКУРЕНТНИЙ МЕТОД ПОТОЧНОГО РЕГРЕСІЙНОГО АНАЛІЗУ ДЛЯ ПОБУДОВИ ЛІНІЙНИХ МОДЕЛЕЙ	59
ГЛАВА 5. КЛАСИФІКАЦІЯ ОБ'ЄКТІВ НА ОСНОВІ МОДЕЛІ ДЕРЕВОВИДНОЇ ШТУЧНОЇ ІМУННОЇ МЕРЕЖІ	75
ГЛАВА 6. ДОСЛІДЖЕННЯ ПОСИЛАЛЬНИХ МЕТОДІВ РАНЖУВАННЯ ДОКУМЕНТІВ ЗА РЕЗУЛЬТАТАМИ ІНФОРМАЦІЙНОГО ПОШУКУ	91
ГЛАВА 7. МЕТОДОЛОГІЯ SAFE ТА ЇЇ ВИКОРИСТАННЯ ДЛЯ ВПРОВАДЖЕННЯ ВИКОНАННЯ ЦІЛЕЙ КІБЕРБЕЗПЕКИ У ГНУЧКОМУ ПІДХОДІ ДО РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	108
ГЛАВА 8. ПІДХОДИ ДО ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	125
ГЛАВА 9. МОДЕЛЮВАННЯ ВІДТВОРЕННЯ ПЛОСКИХ КРИВИХ ЛАМАНИМИ ЗА АСИМПТОТИЧНО - ОПТИМАЛЬНИМ АЛГОРИТМОМ ІНТЕРПОЛЯЦІЇ	141
ГЛАВА 10. РОЗРОБКА ПЕДАГОГІЧНОЇ МОДЕЛІ ФОРМУВАННЯ КОМПЕТЕНЦІЙ ФАХІВЦІВ ПОЛІГРАФІЧНОГО ПРОФІЛЮ З ПОЗИЦІЇ СИСТЕМНОГО ПІДХОДУ	158
ВІДОМОСТІ ПРО АВТОРІВ ТА АНОТАЦІЇ	174

ВСТУП

Сучасний розвиток Інтернет розглядає інформаційні технології як якісно нові технології, особливостями яких є розподілені інтелектуальні системи та системи глибокого навчання.

Сучасні інформаційні технології і системи використовують для створення електронних ринків, де можна відстежити і проконтролювати оплати. Крім того, вони впливають на створення додаткових робочих місць і розширюють можливості для прийняття рішення в медичній, освітній та інших сферах.

Ускладнення економічних процесів вимагає нових підходів до їх дослідження на основі принципів самоорганізації систем.

В монографії представлені результати наукових досліджень в галузі інформаційних технологій та систем, що представлені на Міжнародній науково-практичній конференції «Інформаційні технології та системи», що відбулася 8 - 9 квітня 2021 року на базі Харківського національного економічного університету імені Семена Кузнеця. У ній відображені найбільш цікаві результати за найсучаснішими науковими напрямками застосування, проектування та визначення проблем і перспектив інформаційних технологій в технічних системах тощо.

В главі 1 монографії представлені результати інтегрування чат-бота @es_economy_karkas_bot з прототипами експертних систем для прийняття рішення в режимі онлайн. Аналізується структура бази знань прототипу експертної системи для визначення класу кредитоспроможності позичальника. Наведено приклад онлайн консультації чат-бота в фінансовій предметної області на прикладі визначення оцінки кредитоспроможності позичальника.

В главі 2 монографії представлені результати розробки багатоагентної системи для планування і координації групових дій агентів з метою проведення переговорів і вибору оптимального рішення. Система реалізована на основі чотирьох рівневої архітектури, дозволяє ефективно обробляти запити і керувати продажем товарів в різних торгових точках компанії. Реалізовано основні функції агентів продавців, які полягають в продажу свого товару з каталогу, і агентів покупців - придбати товари за дорученням своїх власників за найменшою ціною. Імітаційне моделювання підтверджує доцільність застосування розробленої мультиагентної системи розподіленої компанії.

В главі 3 виконана формалізація процесу обробки інформації на основі методики синтезу багатоканальних пристроїв тестового контролю з локалізацією помилок у вхідних інформаційних послідовностях. Запропонована архітектура сигнатурного аналізатора, що дозволяє контролювати і діагностувати дані по групах. Наведені приклади створення пристроїв для різних умов використання в розподілених системах.

В главі 4 розглянуто задачу побудови рекурентної форми алгоритму поточного регресійного аналізу, що дозволяє здійснювати оцінювання невідомих параметрів при наявності стаціонарних завад. Отримано основні співвідношення, які описують процеси накопичення нової і скидання застарілої інформації. Показано, що алгоритми, які розглядаються, використовують допоміжну оцінку. Наведено результати імітаційного моделювання, які свідчать про ефективність підходу, що розвивається.

В главі 5 описано рішення проблеми класифікації об'єктів у багатовимірному просторі атрибутів на основі модифікованої моделі штучної імунної мережі з використанням принципів мінімального покриваючого дерева (MST – minimum spanning tree). При класифікації з неконтрольованим навчанням (кластеризації) на різних етапах реалізації штучної імунної мережі об'єкти використовуються як антигени та антитіла для формування навчальної вибірки (класифікатора). У випадку класифікації з контрольованим навчанням об'єкти з навчального набору використовуються як набір антигенів, а об'єкти класифікації – як набір антитіл. Визначення класу для кожного об'єкта базується на значенні авідності, яке описує силу кооперативної афінної взаємодії антитіл з антигеном. Використання запропонованої моделі дозволяє прискорити процес класифікації порівняно з моделями на основі методів MST та C-means, а також автоматизувати процес визначення кількості класів за відсутності навчальної вибірки.

В главі 6 досліджено посиальні методи ранжування документів за запитами користувачів PageRank, HITS, а також запропоновано багатоетапний метод ранжування. Для ітераційного обчислення показників авторитетності і посередницької оцінки використовувалися додатково розроблені функції. Багатоетапний метод ранжування передбачає проведення декількох послідовних етапів ранжування, використовуючи при цьому на кожному етапі певний запит. Даний метод дозволяє послідовно уточнювати список отриманих результатів.

Глава 7 посвящена процесу вибору методології розробки програмного

забезпечення є дуже важливим на початковій стадії проектування продукту. Розробка ПО, як і будь-яка інша технічна дисципліна, має справу з такими основними проблемами: якість, вартість і надійність. У зв'язку з цим правильна організація процесу розробки програмного забезпечення є основою досягнення кращого результату в очікувані терміни, з хорошим рівнем якості, безпеки і з запланованим бюджетом.

Метою глави 8 є аналіз проблем забезпечення якості та тестування програмного забезпечення в процесі його створення в ІТ-компаніях. Об'єктом дослідження є процеси тестування, контролю якості та забезпечення якості. Предметом дослідження є характеристики та моделі оцінювання якості та форми організації роботи команд QA і QC. В процесі дослідження був використаний системний підхід, порівняльний аналіз методів оцінювання якості та форм організації роботи з контролю та забезпечення якості програмних продуктів. Для оцінювання якості програмного забезпечення була запропонована класифікація цих методів, розглянуті методи інтегрального оцінювання якості, заснованого на витратах, і метод оцінки якості на основі ієрархічної моделі. Для оцінювання відмінностей між контролем якості та забезпеченням якості був проведений аналіз їх обов'язків, планування роботи та документації, що дало можливість порівняти виконувані функції і умови роботи.

В главі 9 розглянуто моделювання апроксимації плоских параметричних кривих ламаними за асимптотично оптимальним алгоритмом інтерполяції в хаусдорфовій метриці на основі обчислення похибок апроксимації, досліджено вплив кількісної характеристики ступеню дискретизації інтегральної функції - регулятора вузлів інтерполяції на показники розподілу похибок, а також оптимізація параметру функції - регулятора вузлів для випадку кривих з наявними точками перегину.

В главі 10 запропоновано методику по діагностиці рівнів підготовленості студентів до науково-технічної діяльності, відбору змісту курсової підготовки, розкрито методику проведення занять, що сприяють стимулюванню у учнів позитивних мотивів до науково-технічної діяльності, формуванню у них системного підходу до технічної творчості.

ГЛАВА 1

ІНТЕГРУВАННЯ ЧАТ-БОТА @ES_ECONOMY_KARKAS_BOT З ЕКСПЕРТНОЮ СИСТЕМОЮ ДЛЯ ВИЗНАЧЕННЯ КЛАСУ КРЕДИТОСПРОМОЖНОСТІ ПОЗИЧАЛЬНИКА

Вступ і постановка задачі

Один з головних напрямків штучного інтелекту - це розробка інтелектуальних систем. Цей напрямок переслідує моделювання мистецтва людини аналізувати, структурувати, витягувати і застосовувати знання в погано формалізованих предметних областях. Яскравим представником таких систем є експертні системи (ЕС).

Експертні системи - це клас комп'ютерних систем, які можуть давати пораду на рівні експерта: консультувати, ставити діагноз з певним ступенем впевненості, пояснювати свої висновки. Ядром експертної системи є база знань і алгоритм прийняття рішення (машина висновку).

На шляху створення експертних систем існує кілька перешкод, наприклад, несуперечливість знань в базі знань і її динамічний супровід.

Система «КАРКАС» є інструментарій для розробки прототипів баз знань для експертних и експертно-навчальних систем як в офлайн, так і онлайн режимах на смартфонах. Подання знань ґрунтується на ієрархічній функціональній системі, яка генерується системою "КАРКАС" на базі правил продукцій и фреймів.

Машина висновку використовує ієрархічну функціональну систему під час проведення консультації з користувачем. Користувач може вибрати різні режими роботи машини висновку: використання прямого висновку, зворотнього висновку, непрямого висновку, формули Байєса, таблиці критеріїв, коли консеквент продукції представляє собою список параметрів.

Система "КАРКАС" реалізує основні інструментальні засоби, сервіси (middleware, каркаси) для побудови баз знань предметної області за допомогою ієрархічної функціональної системи і таким чином, полегшує побудову експертних систем.

В даний час онлайн спілкування відіграє величезну роль в житті людей. Тому багато компаній використовують текстові повідомлення як для спілкування

між співробітниками (чати), так і для консультації з експертними системами в онлайн режимі за допомогою чат-ботів [1, 2].

У бізнес-середовищі корпоративним стандартом комунікацій став безкоштовний месенджер TELEGRAM. Це обумовлено наступними причинами: високим ступенем шифрування даних в ньому, стабільністю роботи, можливістю передачі великих обсягів інформації, відкритістю протоколу, кроссплатформенністю.

З іншого боку, що дуже важливо для інтегрування месенджера TELEGRAM в інших програмах це те, що розробники надає бібліотеку на основі API для роботи з чат-ботами.

Чат-боти популярні у фінансовій галузі оскільки володіють такими перевагами, як віддалене підключення і віртуальне присутність. Крім того, допомагають в обслуговуванні клієнтів, що дозволяє, наприклад, банкам скоротити операційні витрати.

Основна частина

Чат-бот @es_economy_karkas_bot месенджера TELEGRAM, використовує месенджер як інтерфейс для онлайн спілкування користувача з системою "КАРКАС" для прийняття ефективних рішень в економіко-фінансовій сфері. Повідомлення команди чат-бота / help показано на рис. 1.



Рис. 1. Повідомлення команди / help чат-бота @es_economy_karkas_bot

Детальніше про командах чат-бота @es_economy_karkas_bot:

/fa викликає прототип ЕС для аналізу фінансового стану підприємства, призначена для підвищення якості результату оцінки фінансового стану підприємства;

/finsost викликає прототип ЕС для аналізу фінансового стану підприємства (критичне, нестабільне, стабільне, перспективне);

/bank_commercial викликає прототип ЕС по підбору банку для фінансового обслуговування підприємства;

/insurance_company викликає прототип ЕС для вибору страхової компанії;
/credit_insurance викликає прототип ЕС для страхування комерційних кредитів;

/enterprise_strategy викликає прототип ЕС для вибору стратегії підприємства;

/product_suppliers викликає прототип ЕС для вибору постачальників продукції;

Розглянемо структуру бази знань прототипу ЕС для визначення класу кредитоспроможності позичальника, яка викликається командою / creditworthiness.

Оцінка кредитоспроможності становить особливий інтерес для банків, тому що їхня прибутковість і ліквідність багато в чому залежать від фінансового стану клієнтів. Надійність, фінансова стійкість клієнтів зменшують банківські ризики й сприяють одержанню банком більш високого доходу. Необхідно побудувати онтологію за оцінкою класу кредитоспроможності позичальника.

Призначення прототипу ЕС: консультування щодо питання оцінки кредитоспроможності підприємства для видачі банком кредиту й зменшення ризику.

Сфера застосування: банки, комерційні установи.

Ціль: визначити клас кредитоспроможності позичальника.

Очікувані результати: визначити значення класу кредитоспроможності позичальника залежно від фінансових і якісних показників, що потім буде прийматися до уваги працівниками банку або іншої комерційної установи при видачі кредиту позичальнику.

Вхідні дані:

для аналізу фінансових показників: значення абсолютної, поточної, загальної ліквідності; структури капіталу; оборотності капіталу; забезпеченості власними джерелами фінансування;

для аналізу якісних показників: аналіз і оцінка кредитної історії позичальника, оцінка ринкової позиції позичальника, оцінка ліквідності застави, оцінка ефективності керування й ділових якостей керівника.

В оцінці фінансової діяльності підприємства оцінюють такі економічні нормативи, як:

коефіцієнт абсолютної ліквідності;

коефіцієнт поточної ліквідності;
 коефіцієнт загальної ліквідності;
 коефіцієнт структури капіталу (незалежності);
 коефіцієнт оборотності капіталу;
 коефіцієнт забезпеченості власними джерелами фінансування.

У табл. 1 наведена характеристика системи коефіцієнтів, які використовуються для оцінки фінансового стану позичальника.

Таблиця 1

Система коефіцієнтів

Найменування коефіцієнта	Характеристика коефіцієнта
Поточна ліквідність	Характеризує, чи здатний позичальник розрахуватися за борговими зобов'язаннями у встановлений термін: $КТЛ = \frac{\text{Поточні активи}}{\text{Поточні пасиви}}$
Загальна ліквідність	Характеризує, наскільки обсяг короткострокових зобов'язань можна погасити за рахунок усіх ліквідних активів
Абсолютна ліквідність	Характеризує, як швидко короткострокові зобов'язання можуть бути погашені високоліквідними активами: відношення суми коштів і короткострокових фінансових вкладень до суми короткострокових зобов'язань
Структура капіталу	Характеризує ступінь фінансового ризику й визначає відношення притягнутих і власних засобів
Оборотність капіталу	Характеризує ступінь мобільності використання власних засобів: відношення виторгу від реалізації до вартості майна
Забезпеченість власними джерелами фінансування (левериджу)	Характеризує ступінь забезпеченості позичальника власним капіталом

Система коефіцієнтів дозволяє визначити фінансовий показник кредитоспроможності позичальника як відношення суми балів усіх показників до їхньої кількості. Кожний з коефіцієнтів буде оцінений за 5-бальною шкалою. Найбільш сприятливим значенням показників привласнюється оцінка "п'ять", найменш сприятливим – "один". Діапазон значень для їхньої оцінки наведений в табл. 2.

Таблиця 2

Граничні значення показників для 5-бального оцінювання

Найменування показників	Граничні значення показників				
	"5"	"4"	"3"	"2"	"1"
1	2	3	4	5	6
Поточна ліквідність	2 .. 2.5	1 .. 1.99	0.7 .. 0.99	0.5 .. 0.69	<0.5
Загальна ліквідність	1 .. 10	0.7 .. 0.99	0.4 .. 0.69	0.2 .. 0.39	<0.2
Абсолютна ліквідність	0.2 .. 10	0.15 .. 0.19	0.1 .. 0.14	0.06 .. 0.1	<0.06
Структура капіталу (незалежності)	0.7 .. 0.8	0.6 .. 0.69	0.5 .. 0.59	0.4 .. 0.49	<0.39
1	2	3	4	5	6
Оборотність капіталу	0.5 .. 0.55	0.3 .. 0.49	0.2 .. 0.29	0.1 .. 0.19	<0.1
Забезпеченість власними джерелами фінансування	0.6 .. 0.8	0.5 .. 0.59	0.3 .. 0.49	0.1 .. 0.29	<0.1

Для оцінки якісних показників діяльності позичальника використовуються наступні показники:

аналіз і оцінка кредитної історії позичальника в частині історії його взаємовідносин з банком;

оцінка ринкової позиції позичальника;

оцінка ефективності керування й ділових якостей керівника;

оцінка ліквідності застави.

За сукупністю балів, розрахованих при оцінці фінансового стану і якісних показниках діяльності, позичальник ставиться до відповідного класу кредитоспроможності. Усього встановлено 5 класів кредитоспроможності: А, Б, В, Г, Д [3]. Аргументація класу кредитоспроможності наведена в правилах бази знання.

У результаті такої комплексної оцінки позичальника повинне бути прийняте зважене управлінське рішення про доцільність видачі або відмову в кредиті даному конкретному позичальнику.

У предметної області виділені класи, які представлені в табл. 3.

Таблиця 3

Класи БЗ

Клас	Кількість примірників класу	Рівень ієрархії класів
Клас кредитоспроможності	26	1
Оцінка якісних показників	6	2
Оцінка фінансового стану	17	2

Схема можливостей для визначення класу кредитоспроможності позичальника подане на рис. 2.

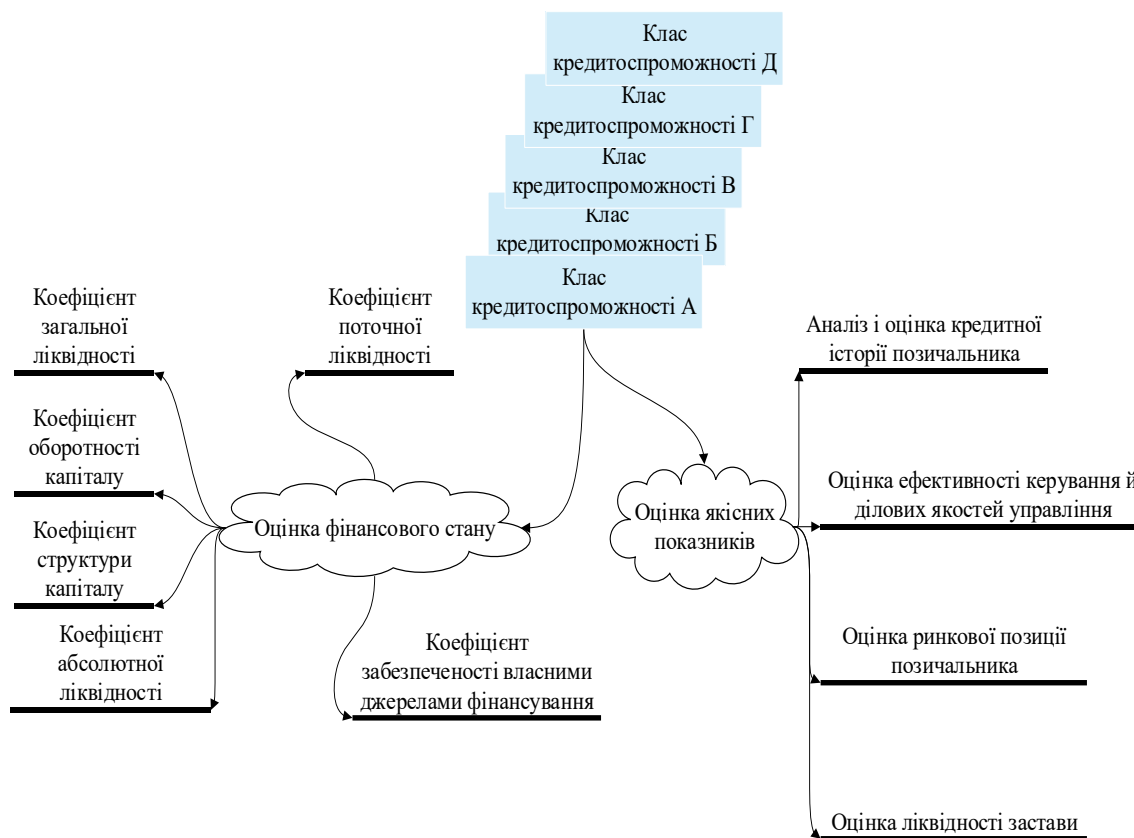


Рис. 2. Схема можливостей для визначення класу кредитоспроможності позичальника

Аналогічним чином прототип ЕС буде варіанти для визначення оцінки фінансових показників позичальника, оцінки якісних показників позичальника, оцінки класу кредитоспроможності позичальника.

Наприклад, дерево логічних можливостей визначення оцінки фінансових показників позичальника приведена на рис. 3.

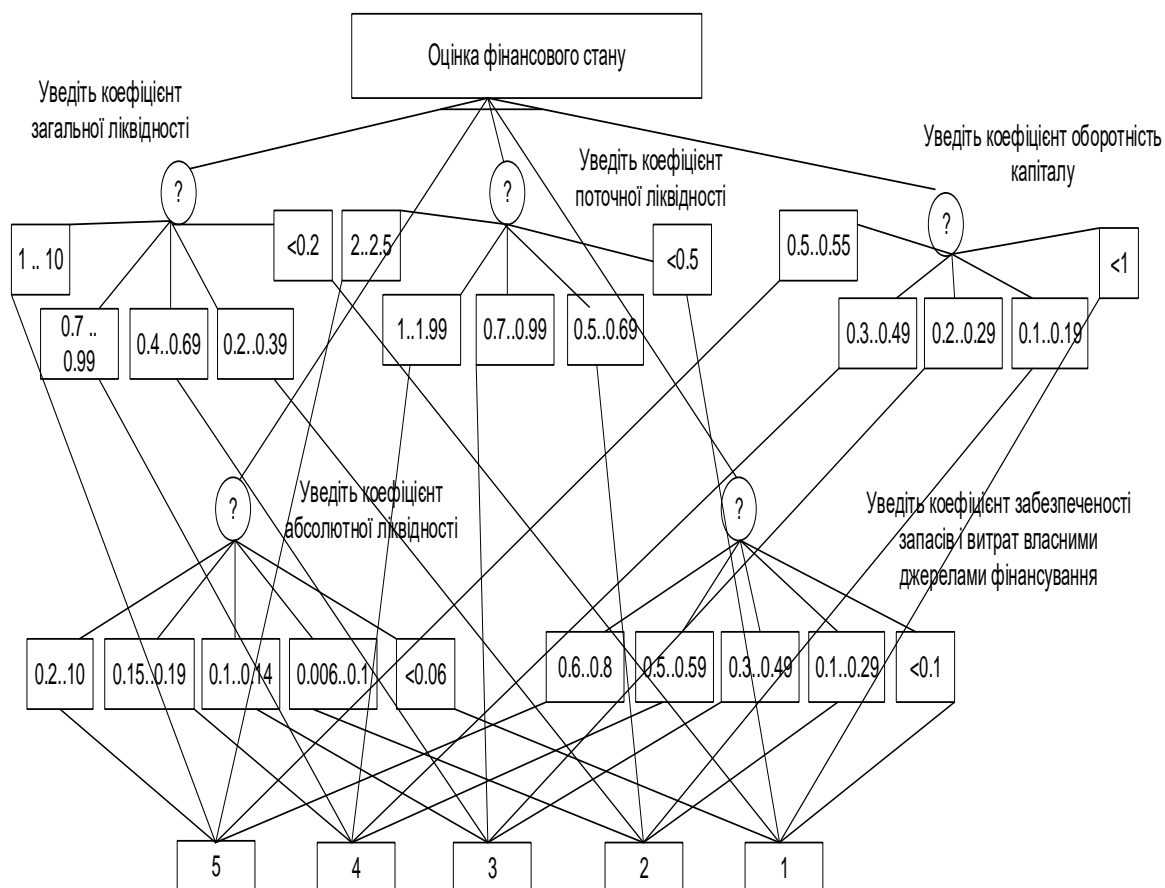


Рис. 3. Дерево логічних можливостей визначення оцінки фінансового стану позичальника

Оцінка класу кредитоспроможності визначається на основі оцінки фінансових і якісних показників позичальника, наприклад, в наступних представлених правилах.

Правило 1. A&B&C&D&E&F#.

ЯКЩО

A Коефіцієнт абсолютної ліквідності = 0.2 .. 10

B Коефіцієнт поточної ліквідності = 2 .. 2.5

C Коефіцієнт загальної ліквідності = 1 .. 10

D Коефіцієнт оборотності капіталу = 0.5 .. 0.55

E Коефіцієнт структури капіталу = 0.7 .. 0.8

F Коефіцієнт забезпеченості власними джерелами фінансування = 0.6 ..

0.8

ТО

Оцінка фінансового стану = 5.

Фрагмент дерева логічних можливостей визначення оцінки якісних показників позичальника приведена на рис. 4.

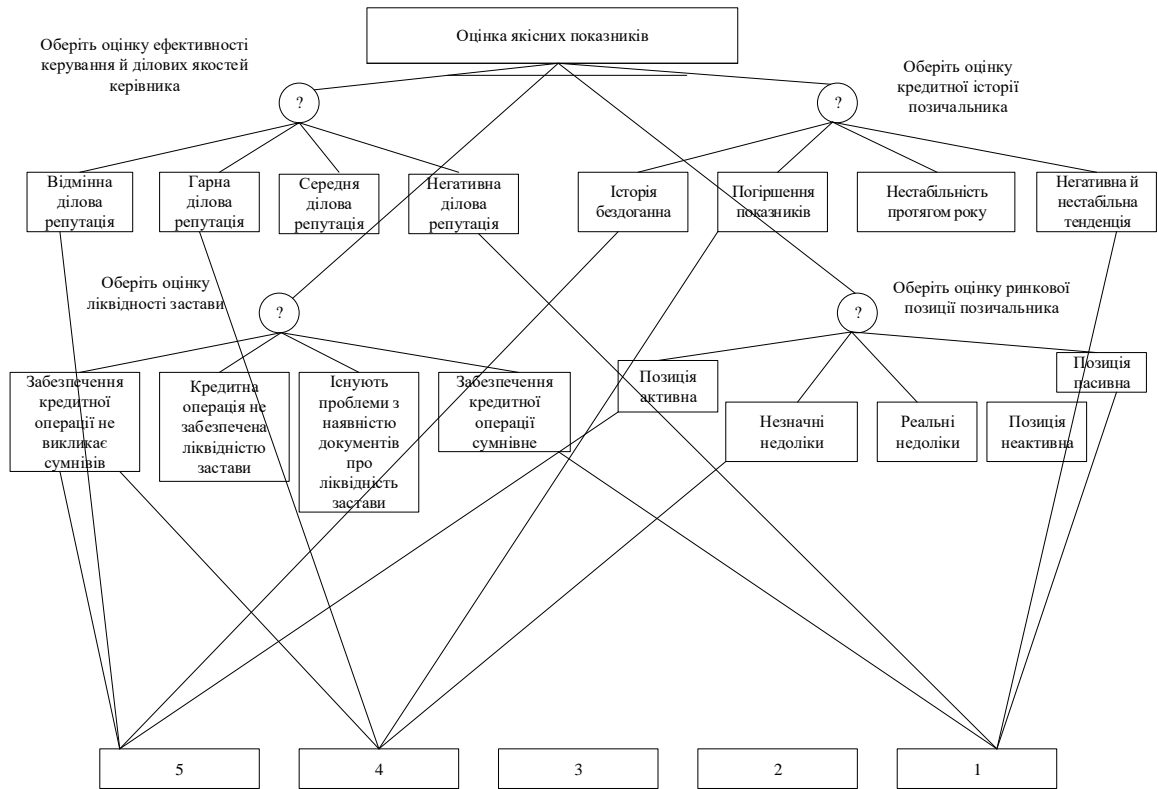


Рис. 4. Фрагмент дерева логічних можливостей визначення оцінки якісних показників позичальника

Одне з правил для визначення оцінки якісних показників позичальника має наступний вид:

Правило 6. A&B&C&D#.

ЯКЦО

A Аналіз і оцінка кредитної історії позичальника = Історія бездоганна

B Оцінка ринкової позиції позичальника = Позиція активна

C Оцінка ліквідності застави = Забезпечення кредитної операції не викликає сумнівів

D Оцінка ефективності керування й ділових якостей управління = Відмінна ділова репутація

ТО

Оцінка якісних показників = 5.

Зауваження. Якщо під час консультації машина висновку не знайшла відповідне правило, тоді активізується фрейм 3 слоти, якого заповнюються відповідями користувача. Таким чином, експерту надається можливість або додати нове правило або виключити такий варіант прийняття рішення машиною висновку.

Приклад фрейм 3:

Фрейм 3.

Ім'я слоту | Тип слоту | Спадкування

Аналіз і оцінка кредитної історії позичальника | Заміщення | н

Оцінка ефективності керування й ділових якостей керівника | Заміщення |

н

Оцінка ліквідності застави | Заміщення | н

Оцінка ринкової позиції позичальника | Заміщення | н

Цільовий слот.

Оцінка якісних показників | НЕ ВИЗНАЧЕНО.

Дерево логічних можливостей для визначення класу кредитоспроможності позичальника подане на рис. 5, де дужкою позначена вершина типу "Г", а відсутність дужки – "АБО".

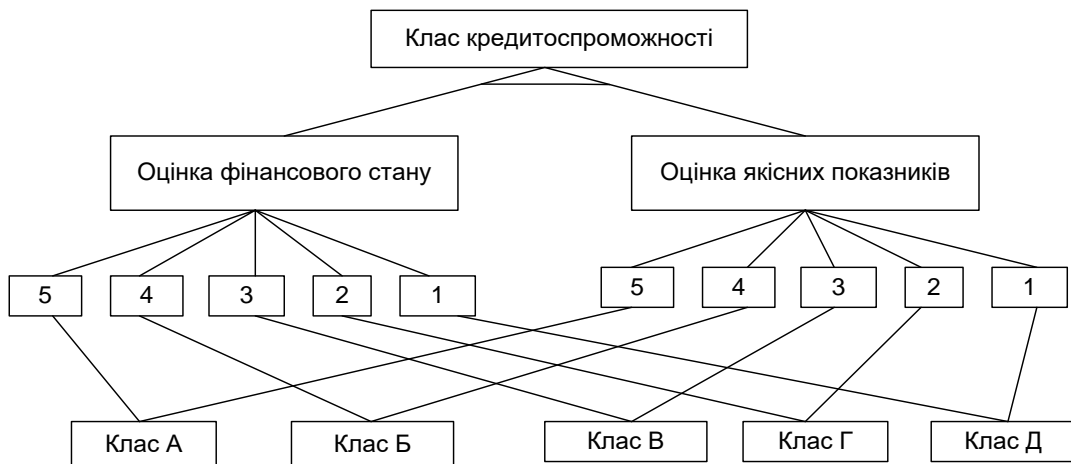


Рис. 5. Дерево логічних можливостей для визначення класу кредитоспроможності позичальника

Наступне правило демонструє аргументацію прийняття рішення.

Правило 11. А&В#.

ЯКЩО

А Оцінка якісних показників = 5

В Оцінка фінансового стану = 5

ТО

Клас кредитоспроможності = А.

Аргументація. Фінансова діяльність задовільна, що свідчить про можливість вчасно погасити заборгованість за кредитними операціями, в тому числі погашення основної суми боргу і відсотків по ньому відповідно до умов кредитної угоди і економічних показників в межах встановлених значень.

Кредитна історія позичальника бездоганна.

Ринкова позиція позичальника активна, що дає можливість гнучко реагувати на зміни кон'юнктури ринку, збільшувати власну конкурентно-здатність, зменшувати ризик неповернення кредиту.

Вище керівництво позичальника має відмінну ділову репутацію.

Забезпечення кредитної операції не викликає сумнівів.

Правило 14. А&В#.

ЯКЩО

А Оцінка якісних показників = 2

В Оцінка фінансового стану = 2

ТО

Клас кредитоспроможності = Г.

Аргументація. Фінансова діяльність незадовільна, економічні показники не відповідають установленим значенням.

Кредитна історія позичальника характеризується нестабільністю протягом року.

Ринкова позиція позичальника неактивна, що приводить до ризику значних збитків, до низької ймовірності повного погашення кредитної заборгованості й відсотків.

Вище керівництво позичальника має негативну ділову репутацію.

Забезпечення кредитної операції сумнівне.

Правило 15. А&В#.

ЯКЩО

А Оцінка якісних показників = 1

В Оцінка фінансового стану = 1

ГО

Клас кредитоспроможності = Д.

Аргументація. Фінансова діяльність незадовільна, є збитки, економічні показники не відповідають установленим значенням.

Кредитна історія позичальника характеризується негативними й нестабільними тенденціями.

Ринкова позиція позичальника пасивна, що свідчить про відсутність імовірності виконання зобов'язань позичальником.

Вище керівництво позичальника має негативну ділову репутацію.

Кредитна операція не забезпечена ліквідною заставою.

Фрейм 2.

Ім'я слоту | Тип слоту | Спадкування

Оцінка якісних показників | Заміщення | н

Оцінка фінансового стану | Заміщення | н

Цільовий слот.

Клас кредитоспроможності | НЕ ВИЗНАЧЕНО.

Зауважимо що, фрейм 2 активізується, коли не знайдено значення фінансового і якісного показників.

Машина висновку системи "КАРКАС" використовує ієрархічну функціональну систему під час проведення консультації з користувачем. Користувач може вибрати різні режими роботи машини висновку: використання прямого висновку, зворотнього висновку, непрямого висновку, формули Байєса, таблиці критеріїв, коли консеквент продукції представляє собою список параметрів.

Вид ієрархічної функціональної системи для вибору класу кредитоспроможності позичальника наведено на рис. 6.

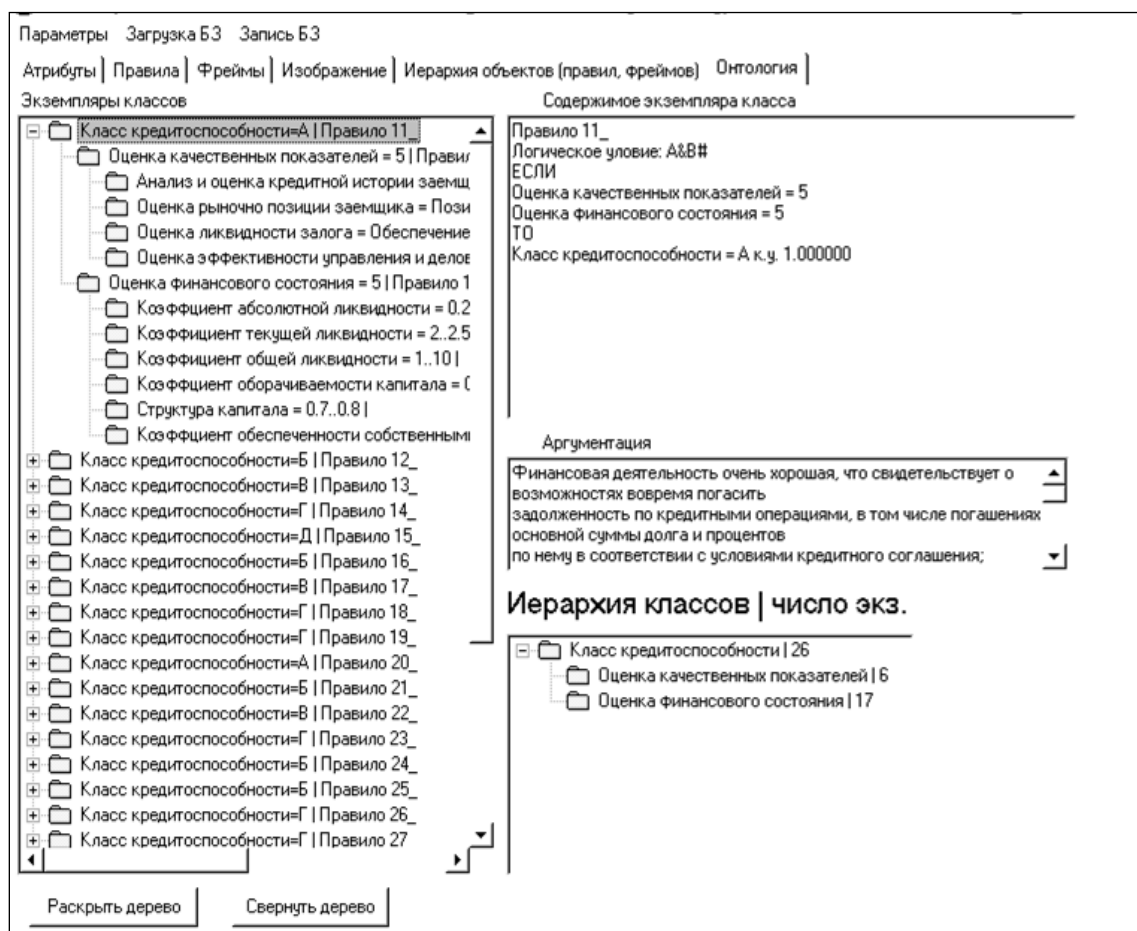


Рис. 6. Вид ієрархічної функціональної системи для вибору класу кредитоспроможності позичальника

Модуль онлайнної консультації (співрозмовник) системи "КАРКАС" дозволяє за допомогою месенджера TELEGRAM обмінюватися повідомленнями з базами знань через Інтернет.

Агенти консультації та діалогу обмінюються повідомленнями між собою для виконання наступних операцій:

1. Натискання: кнопок, чек боксів, радіо кнопок.
2. Передача і прийом повідомлень між візуальними об'єктами на формі.

Таким чином, зазначені вище модулі виконують функції агентів і в цьому сенсі імплементований чат-бот @es_economy_karkas_bot в систему "КАРКАС" можна розглядати, як багатоагентну систему.

Передача і прийом повідомлень агента консультації.

1. Активувати додаток (es_economy_karkas_bot.exe), що запускає бот, можна на ресурсі, що має доступ до інтернету (хостинг, домашній комп'ютер).

Потім в месенджері телеграм запустити його: @es_economy_karkas_bot. Набрати команди / help або / start бот і бот запропонує вибрати команди для запуску експертних систем, тестів (рис. 1).

2. Наприклад, при виборі команди /creditworthiness виконуються наступні операції:

завантажується база знань creditworthiness.knb з сайту <https://it-karkas.com.ua>;

виконується модуль консультації і запускається машина висновку експертної системи;

активізується модуль діалогу.

3. Результат консультації експертної системи передається боту по широкомовному протоколу.

Архітектура інтеграції чат-бота з системою "КАРКАС" наведено на рис. 7.

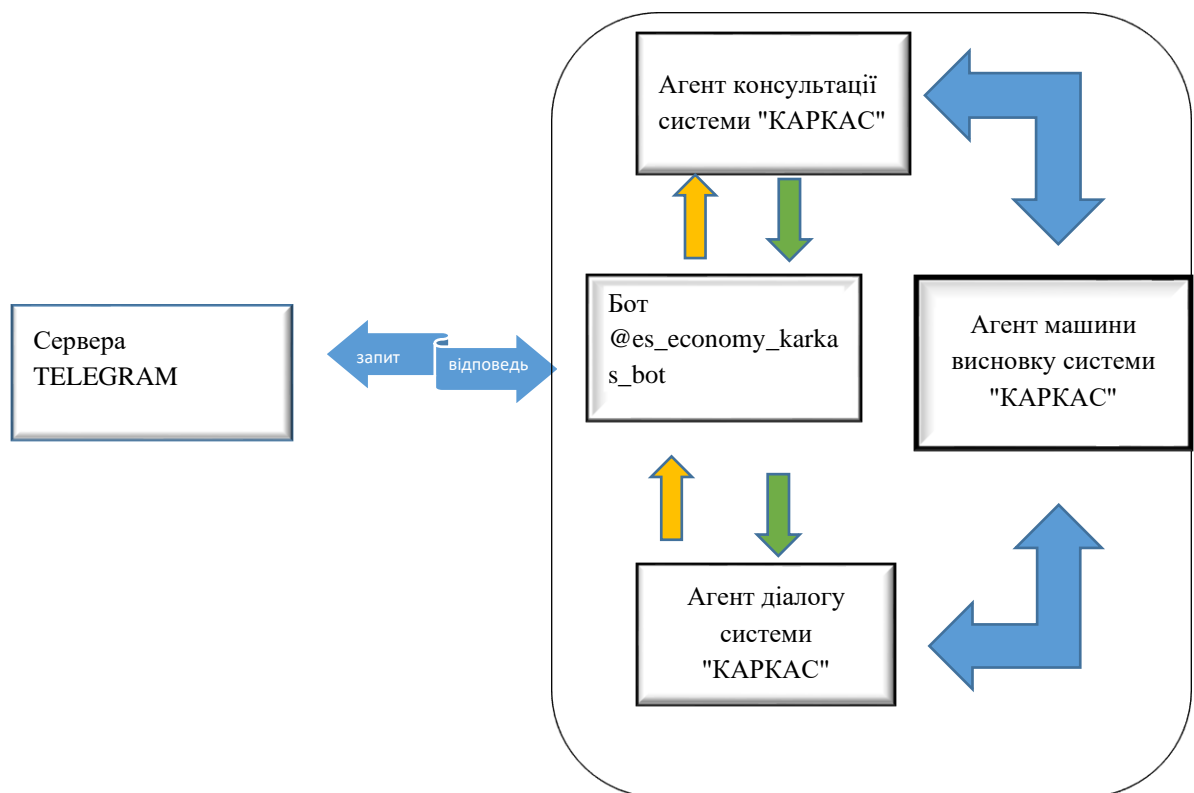


Рис. 7. Архітектура інтеграції чат-бота з системою "КАРКАС"

Таким чином, алгоритм роботи чат-бота @es_economy_karkas_bot складається з наступних кроків:

Крок 1. Активувати чат-бот @es_economy_karkas_bot в месенджері TELEGRAM.

Крок 2. Вибрати команди: /help або /start, потім, команда /creditworthiness викликає прототип ЕС для вибору класу кредитоспроможності позичальника.

Крок 3. Бот запускає агента консультації системи "КАРКАС".

Крок 4. Активізується машина висновку системи "КАРКАС".

Крок 5. Формується ієрархічна функціональна система для ведення діалогу з користувачем.

Крок 6. Активізується агент діалогу, котрий посилає боту повідомлення з текстом питання і відповідями. Бот приймає повідомлення у вигляді об'єкта JSON, виконує його парсинг, відображає повідомлення в чаті і чекає відповіді користувача.

Крок 7. Користувач в чат-боті вибирає або вводить відповідь. Бот відсилає відповідь машині висновку експертної системи.

Крок 8. Агент консультації експертної системи приймає повідомлення і передає його машині висновку, яка передає повідомлення агенту діалогу. Мета консультації уточнюється, на основі ієрархічної функціональної системи, під час діалогу з користувачем.

Крок 9. Ітеративний процес консультації триває поки машина висновку не отримає результат від експертної системи. Користувач може в будь-який момент припинити консультацію командою /quit.

Висновки

В роботі представлені результати інтегрування чат-бота @es_economy_karkas_bot з експертною системою для організації консультування в режимі онлайн. Розглянуто алгоритм взаємодії чат-бота і агентів експертної системи в онлайн режимі.

В результаті був створений повністю функціонуючий чат-бот @es_economy_karkas_bot, який інтегрований в систему "КАРКАС" і дозволяє в режимі онлайн проводити консультацію з прототипами експертних систем в економіко-фінансовій предметній області.

Подальшим напрямком даного дослідження може стати розробка мобільного додатка "КАРКАС" під платформами Android і iOS.

Система "КАРКАС" за допомогою чат-ботів: @es_economy_karkas_bot, @Ribs_karkas_bot, @test_karkas_bot, @es_info_tech_karkas_bot дозволяє проводити онлайн консультацію з користувачами і тестування знань студентів в різних предметних областях: комп'ютерна графіка, технології баз даних, веб-аналітика, системи бізнес-інтелекту.

Література

1. Рассел С. Искусственный интеллект: современный поход / С. Рассел, П. Норвиг. – 2-е изд.; [Пер. с англ.] – М.: Изд. дом "Вильямс", 2006. – 1408 с.
2. Бурдаєв В.П. Використання чат-бота @es_economy_karkas_bot для онлайн консультації з експертною системою. / Системи обробки інформації: збірник наукових праць. - Х.:Харківський університет Повітряних Сил імені Івана Кожедуба, 2020. - Вип. 1(160). - 100 - 106с.
3. Вовк В.Я., Хмеленко О.В. Кредитування і контроль: навч. посіб. / В.Я. Вовк, О.В. Хмеленко. — К.: Знання, 2008. — 463 с.
4. Бурдаєв В. П. Моделі баз знань: моногр. / В. П. Бурдаєв – Харків: ХНЕУ, 2010. – 300 с.
5. Burdaev V. P. "About one concept of constructing a temporal knowledge base", in Proc. of the 1st International Congress Fundamental and Applied Studies in the Pacific and Atlantic Oceans Countries, Tokyo University Press, 2014, pp. 272–276.

ГЛАВА 2

МУЛЬТИАГЕНТНА СИСТЕМА РОЗПОДІЛЕНОЇ ТОРГОВОЇ ФІРМИ

Введення та постановка задачі

З появою глобальної Інтернет-економіки складність і динаміка бізнесу почала стрімко зростати. При цьому бізнес вимагає постійного вдосконалення сервісу для клієнтів, все більше індивідуалізованого і персоналізованого підходу до кожного споживача, постійних інновацій та скорочення витрат, розширення бізнесу в нові сфери і мінімізації ризиків. Прийняття рішень в цих умовах характеризується високою невизначеністю і вимогою роботи в реальному часі, наявністю безлічі взаємно суперечливих критеріїв, необхідністю враховувати знання і досвід фахівців, які не тільки погано формалізуються, а й постійно змінюються.

Виникла необхідність в більш потужних і гнучких інтелектуальних програмних системах, які здатні безперервно здобувати нові знання і змінювати свою структуру і функції, розвиватися та адаптуватися до вирішуваних завдань і умов зовнішнього середовища.

Одним із шляхів вирішення подібних завдань можна назвати застосування мультиагентних систем (МАС), які отримали стрімкий розвиток. Мультиагентні технології приносять нові унікальні можливості для компаній, дозволяючи вирішувати складні завдання і створювати програмні системи, що надають нову якість сервісу, високу ефективність і ряд інших переваг перед конкурентами. Уже сьогодні агентно-орієнтований підхід знаходить широке застосування в таких галузях, як розподілене вирішення складних завдань, поєднане проектування виробів, реінжиніринг бізнесу і побудова віртуальних підприємств, імітаційне моделювання інтегрованих виробничих систем і електронна комерція, організація роботи колективів роботів, інтелектуальна обробка інформації, в тому числі підтримка логістичних мереж великої розмірності, адаптивне планування.

Також можна відзначити, що в даний час широку популярність придбала одна з форм спеціалізації по функціях збуту, яка передбачає паралельну діяльність двох груп торгового персоналу. Перша група працює з наявними каналами комунікацій – з телефоном, Інтернетом, мобільним зв'язком, електронною

поштою (кажучи простіше, «сидять на телефонах»). Їх діяльність отримала назву телемаркетингу. Друга група співробітників - це зовнішні торговельні представники, які працюють на місцях. Перед двома групами стоять різні завдання. Хоча очевидно, що різноманітність комунікаційних каналів не зможе замінити собою реальний збут, телемаркетинг відіграє значну роль в наступних видах діяльності (рисунок 1).



Рис. 1. Задачі телемаркетингу

Телемаркетинг широко поширюється, тому що, по-перше, він підвищує ефективність роботи продавців. З точки зору клієнтів: через централізацію закупівельної діяльності і надмірної пропозиції товарів і потенційних постачальників, вартість робочого часу агента із закупівель збільшується. Тому звернення по телефону більш краще, особливо коли мова йде про рішення рутинних питань, наприклад, про розміщення повторних замовлень, інформування про спеціальні програми продажу, надання знижок тощо. Телефонний дзвінок займає значно менше часу, ніж особистий візит. З точки зору компанії-продавця: поєднання діяльності внутрішніх співробітників і зовнішніх торгових агентів з добре продуманим комплексом таких заходів як цільова реклама, поштові розсилки, «гаряча лінія» та активний веб-сайт, підвищують ефективність роботи торгового персоналу в цілому. Комбінація телемаркетингу з іншими заходами щодо просування товарів дозволяють істотно знизити витрати на рутинні операції збуту і сконцентрувати зусилля продавців-аутсорсерів на діяльності, що забезпечує максимальну віддачу в довгостроковій перспективі (наприклад, пошук нових та обслуговування великих існуючих клієнтів). Застосування телемаркетингу найбільш доцільно, коли корпоративна політика взаємодії з клієнтами передбачає розподіл праці торгового персоналу по різних категоріям клієнтів в залежності від величини і купівельного потенціалу останніх. Розвиток комунікаційних каналів дає можливість торговому персоналу працювати з клієнтами безпосередньо з офісу при найменших витратах, що

дозволяє компанії охоплювати сегмент дрібних покупців. Різноманітні функції персоналу вимагають розробки різної політики та різних планів заходів для кожної групи. Щоб з максимальною віддачею використовувати можливості телемаркетингу, необхідно розробити типові сценарії для співробітників, спілкуватися з клієнтами по телефону і в онлайні. Торгові представники на місцях можуть користуватися більшою свободою дій і адаптувати свої презентації до потреб конкретних підготовки відповідних програм навчання і оплати праці різних груп працівників.

Таким чином, рішення задач контролю і управління дистриб'юторськими фірмами викликає великі труднощі. В даний час практично невирішеними залишаються питання, що пов'язані ефективною обробкою запитів і керуванням продажем товару в різних торгових точках компанії.

Застосування агентно-орієнтованого підходу з використанням автономних інтелектуальних агентів зможе подолати зазначені труднощі.

Для вирішення цієї проблеми пропонується мультиагентна система розподіленої торгової компанії, яка дозволяє проектувати агентів, здатних існувати та узгоджено взаємодіяти в складних системах та підвищує якість автоматизованого управління, надаючи додаткові послуги.

Модель мультиагентної системи розподіленої торгової компанії

Застосуванню агентних технологій для вирішення завдань планування, управління, діагностики та моніторингу приділяється дедалі більше уваги. Агентські технології використовуються під час вирішення задач реального світу як у теорії (комунікаційні витрати, виразність мов описання, подання знань), так і на практиці (засоби розробки для налагодження, моніторингу виконання, планування в безперервних просторах параметрів). Не дивлячись на те, що розроблено велике різноманіття методів і моделей описання поведінки агента у різноманітних ситуаціях, як і раніше актуальна проблема синтезу систем, що підтримують найкращі стратегії еволюції агентів у обставинах, що характеризуються високим рівнем невизначеності. При цьому бізнес вимагає постійного вдосконалення сервісу для клієнтів, все більш індивідуалізованого і персоналізованого підходу до кожного споживача, постійних інновацій та скорочення витрат, розширення бізнесу в нові сфери і мінімізації ризиків.

Виконання завдання у середовищі мультиагентної системи може вимагати співпраці між агентами. З огляду на набір агентів і набір завдань, які вони мають

задовольняти, розглядаються ситуації, коли кожна задача має бути приєднана до групи агентів, які виконуватимуть це завдання. Розподіл завдань по групах агентів необхідно, коли завдання не можуть бути виконані за допомогою одного агента.

Коли потрібно вирішувати завдання в гетерогенному розподіленому середовищі, потрібні утиліти для опису обробки в конкретній галузі і знання про апаратно-програмну інфраструктуру. Ці багатокomпонентні знання використовуються для ефективного прийняття рішень.

Індивідуальні знання відіграють важливу роль в організації. Керування індивідуальними знаннями має підкріплюватися моніторингом знань агентів і рухатися в напрямку того, щоб управляти їхніми особистими знаннями.

Традиційно, дуже важко обробляти та управляти знаннями агентів, що призводить до:

- більш високої трудомісткості для запитів знань;
- відсутності відповідних знань;
- деградації загальної якості;
- відсутності ефективної комунікації під час роботи із запитами клієнтів.

Мультиагентна система подається так

$$MAS = \{A, A_i^a, E_i^e, Res\} \quad (1)$$

де $A = \{A_{seller}, A_{buyer}\}$ – множина агентів, що функціонують у середовищі E ; A_i^a – множина дій агентів; E_i^e – множина станів середовища; Res – Web-портал, що надає бізнес послуги, побудований на основі організаційної моделі для взаємодії агентів.

Модель взаємодії агента продавця і агента покупця представлена у вигляді ієрархії взаємних уявлень про цінності товару, що продається. Агенти повинні прийти до угоди ціни, по якій відбудеться угода купівлі-продажу. Необхідна умова укладання угоди: з точки зору обох учасників ціни всіх продавців не перевищують суб'єктивні ціни покупців.

Запропонована модель розбивається на автономні сутності (покупець, продавець);

- кожна сутність має власних агентів;
- агенти покупців шукають собі можливість придбати товар за мінімальну ціну, агенти продавця намагаються задовольнити свою потребу - продати товар

за максимальну ціну;

– можливості агента покупця і потреби агента продавця перебувають у постійному пошуку відповідності, обумовленої заданими індивідуальними критеріями для кожної сторони.

Для звершення угоди між покупцем і продавцем за необхідної ціною використовується структура інформативності, яка включає таке: всі учасники повинні бути впевнені, що для всіх їхніх агентів (покупця з точки зору продавця, продавця з точки зору покупця) цінність товару в точності дорівнює встановленій ціні. Будемо вважати, що агент продавця A_{seller} та агент покупця A_{buyer} повинні прийти до компромісу щодо вартості деякого товару. Угода може відбутися, тільки якщо цінність товару для покупця не нижче, ніж для продавця:

$$\begin{cases} \mathfrak{R}_{ib} \geq \mathfrak{R}_i \\ \mathfrak{R}_{is} \leq \mathfrak{R}_i \end{cases}, \forall i \in \Omega \quad (2)$$

де \mathfrak{R}_b – уявлення покупця про цінність товару, що купується (максимальна ціна, яку він готовий за нього заплатити); \mathfrak{R}_s – уявлення продавця про цінність товару, що продається (мінімальна ціна, за яку він готовий продати товар); \mathfrak{R}_{bs} – уявлення покупця про наміри продавця; \mathfrak{R}_{sb} – уявлення продавця про наміри покупця; \mathfrak{R}_{sbs} – уявлення продавця про те, що про його наміри думає покупець; i – довільна кінцева послідовність індексів з множини учасників угоди $\{A_{seller}, A_{buyer}\}$, Ω – множина всіляких кінцевих послідовностей індексів.

Суб'єктивний розмір області компромісу може бути представлений у вигляді:

$$\Delta_i = \mathfrak{R}_{ib} - \mathfrak{R}_{is}, \quad (3)$$

причому $\forall i \in \Omega, \Delta_i \geq 0$.

Запропонована архітектура мультиагентної системи, яка складається з чотирьох рівнів як показано на рисунку 2.

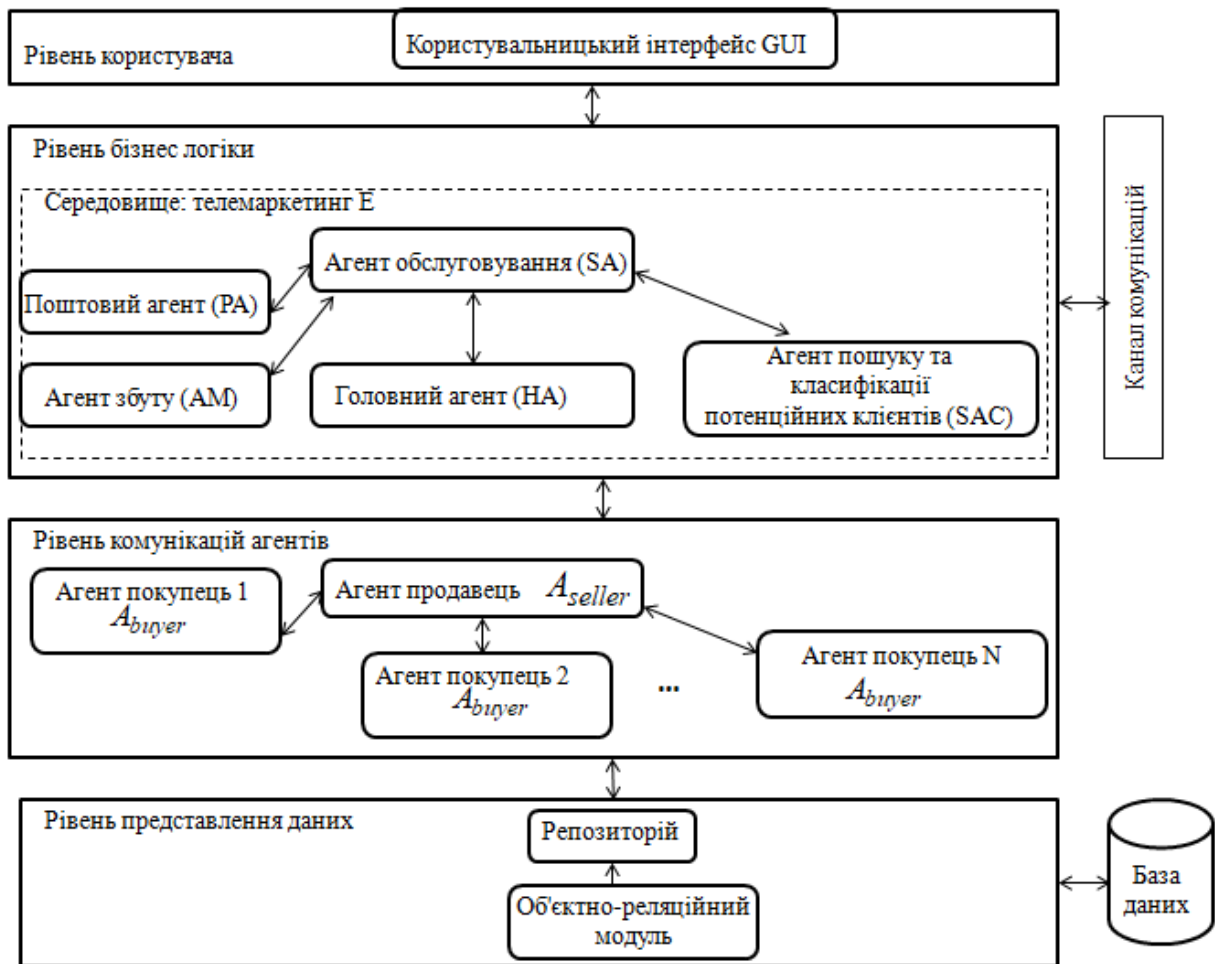


Рис. 2. Архітектура мультиагентної системи

Першим рівнем в архітектурі виступає рівень представлення даних. На цьому рівні розроблена база даних за допомогою Code First зберігає моделі, що описують використовувані сутності. Також тут розміщуються специфічні класи для роботи з різними технологіями доступу до даних, наприклад, клас контексту даних Entity Framework.

Другий рівень комунікації агентів – містить два типи агентів: агента покупця A_{buyer} , відповідального за покупку товару від імені покупця і агента продавця A_{seller} , відповідального за продаж товару від імені продавця.

Взаємодію між агентами, які спільно прагнуть досягти мети при одночасному поділі функцій та ролей, можна організувати так.

В мультиагентній системі (1) кожний агент описується таким набором

$$A_i = \{A_i^a, E_i^e, Feaa, A_i^s, Fase, Fasa\}, \quad (4)$$

де A_i^a – множина дій агентів,

E_i^e – множина станів зовнішнього середовища;

$F_{eaa}: A_i^a \times E_i^e \rightarrow 2^{A_i^a}$ – функції поведінки зовнішнього середовища;

A_i^s – множина внутрішніх станів агентів (передає, отримує або очікує повідомлення);

$F_{ase}: A_i^s \times E_i^e \rightarrow A_i^s$ – функції відновлення стану агента;

$F_{asaa}: A_i^s \rightarrow E_i^e$ – функції прийняття рішень для здійснення дій агентом за поточним внутрішнім станом.

Планувати дії агент може в такий спосіб

$$AP = \{A_i^a, P_i^{ap}, I_i^{ap}, \rho_{ap}, l_{ap,0}\}, \quad (5)$$

де P_i^{ap} – множина сприйняття агентом станів зовнішнього середовища; I_i^{ap} – підмножина внутрішніх станів агента, що є частиною множини внутрішніх станів

$$I = I_i^{ap} \times I'; \quad (6)$$

$\rho_{ap} \subseteq P_i^{ap} \times I_i^{ap} \times I_i^{ap} \times A_i^a$; – відношення переходів, що визначає за сприйняттям

P_i^{ap} поточне сприйняття зовнішнього середовища та поточний внутрішній стан плану l_i^{ap} ; $l_{ap,0} \in I_i^{ap}$ – початковий стан агента.

Результатом планування є множина ланцюжків переходів агентів з початкового стану в кінцевий, що реалізує поставлену мету.

Третій рівень бізнес-логіки містить набір компонентів, які відповідають за обробку отриманих від рівня уявлень даних. Тут реалізується вся необхідна логіка програми, всі обчислення, відбувається взаємодія з базою даних і результати обробки передаються нижньому рівню, реалізуються правила і принципи поведінки агентів продавців і агентів покупців $A = \{A_{seller}, A_{buyer}\}$, які беруть участь в аукціоні.

Зобов'язання агентів $Com(seller, buyer, an, re)$ означають, що в ході виникнення антецедента результат буде отриманий, коли продавець обслужить покупця ($seller$ – продавець, $buyer$ – покупець, an – антецедент, re – результат).

Взаємодія між агентами може бути витлумачена з погляду впливу на їхні

зобов'язання. Наприклад, пропозиція від продавця HardwareMarket про покупку навушників TWS_OPPO_W31 за ціною 1200 грн може бути витлумачена як $Com(\text{agentsHardwareMarket}, \text{buyerTWS_OPPO_W31}, \text{навушники_TWS_OPPO_W31}, \text{купити_за_1200_грн})$. Інакше кажучи, агент продавця A_{seller} та агент покупця A_{buyer} прийшли до компромісу щодо вартості навушників TWS_OPPO_W31, і у цьому випадку агент продавця $\text{agentsHardwareMarket}$ посилає повідомлення агентіві покупця buyerTWS_OPPO_W31 про покупку навушників TWS_OPPO_W31 вартістю 1200 грн (табл. 1,2).

Агенти обмінюються повідомленнями. Модель комунікації складається з таких припущень:

1. Зв'язок здійснюється так: $A_i.send(A_j, m)$ – агент A_i відправляє повідомлення m агентіві A_j ($i, j = \{seller, buyer\}$).

2. Повідомлення достовірні. Повідомлення не створюються та не руйнуються інфраструктурою.

3. Повідомлення впорядковані. Упорядковані спостереження відправляються у вигляді повідомлень. Одержувач приймає повідомлення за порядком проходження.

Спостереження агента A_i описуються послідовністю повідомлень $\langle m_0, m_1, \dots, m_n \rangle_{A_i}$, спостереження мультиагентної системи описуються вектором

$$Obs = [Obs_{A_{seller}}, Obs_{A_{buyer}}], \quad (7)$$

де Obs_{A_i} – послідовність спостережень агента A_i .

Спостереження Obs агентів $\forall A_i, A_j \in MAS$ слухні в тому випадку якщо:

– повідомлення $A_i.send(A_j, m_k)$ належить Obs_{A_j} , тоді повідомлення $A_i.send(A_j, m_k)$ належить і Obs_{A_i} ($k = \overline{0, n}$);

– повідомлення $A_i.send(A_j, m_1)$ належить Obs_{A_j} і повідомлення $A_i.send(A_j, m_0)$ передує повідомленню $A_i.send(A_j, m_1)$ в Obs_{A_i} , тоді повідомлення $A_i.send(A_j, m_0)$ передує повідомленню $A_i.send(A_j, m_1)$ в Obs_{A_j} .

Таким чином, спостереження Obs_{MAS} є множиною усіх станів системи MAS .

Система перебуває в стійкому стані, якщо ніякі повідомлення не

приймаються і не передаються. Спостереження перебувають у стійкому стані, якщо $\forall A_i, A_j \in MAS$ повідомлення $A_i.send(A_j, m_k)$ належить Obs_{A_i} , тоді повідомлення $A_i.send(A_j, m_k)$ належить Obs_{A_j} ($k = \overline{0, n}$);

Нехай Obs_{A_j} – послідовність повідомлень виду $\langle m_0, m_1, \dots, m_n \rangle_{A_j}$. Тоді для будь-якого повідомлення m' форма $\langle m_0, \dots, m_n, m' \rangle_{A_j}$ є конкатенацією Obs_{A_j} з m' . Множина пропозицій $Seq(Obs_{A_j})$, яку можна вивести з множини Obs_{A_j} , є станом A_j після спостереження повідомлень в Obs_{A_j} . Обмеження цілісності виглядає в такий спосіб: $\lfloor m[Ant2:Ant1]m' \rfloor_{A_j}$, де $Ant2$ й $Ant1$ перед- і після умови впливу на m , m' – ефект впливу на m за умови виконання перед- і після умов.

Узгодженість в мультиагентній системі виглядає в такий спосіб: $[\langle MAS \rangle]$, якщо $\forall Obs_{A_j} \in Obs_{MAS}$ спостереження Obs_{A_j} стійкі й повноцінні щодо обмежень цілісності:

$$\begin{aligned} \forall A_i, A_j \in MAS : Com(seller, buyer, an, re) \in Seq(Obs_{A_j}) \Rightarrow \\ \Rightarrow Com(seller, buyer, an, re) \in Seq(Obs_{A_i}). \end{aligned} \quad (8)$$

Таким чином, спостереження того самого антецеденту (продажу) продавцем і покупцем призводить до здійснення однієї дії для досягнення бажаного результату.

Агент покупець отримує назву товару і максимальну ціну, яку покупець може заплатити за товар. Після чого агенти починають шукати один одного, переміщаючись по ринку. На рисунках 3 і 4 показані стани агента продавця і агента покупця.

Агент продавець після отримання бажаних описів товару шукає одного або декількох агентів покупців, і запрошує їх взяти участь в аукціоні. Як тільки агент продавець отримує кілька учасників, він починає аукціон і пропонує угоду покупцям.

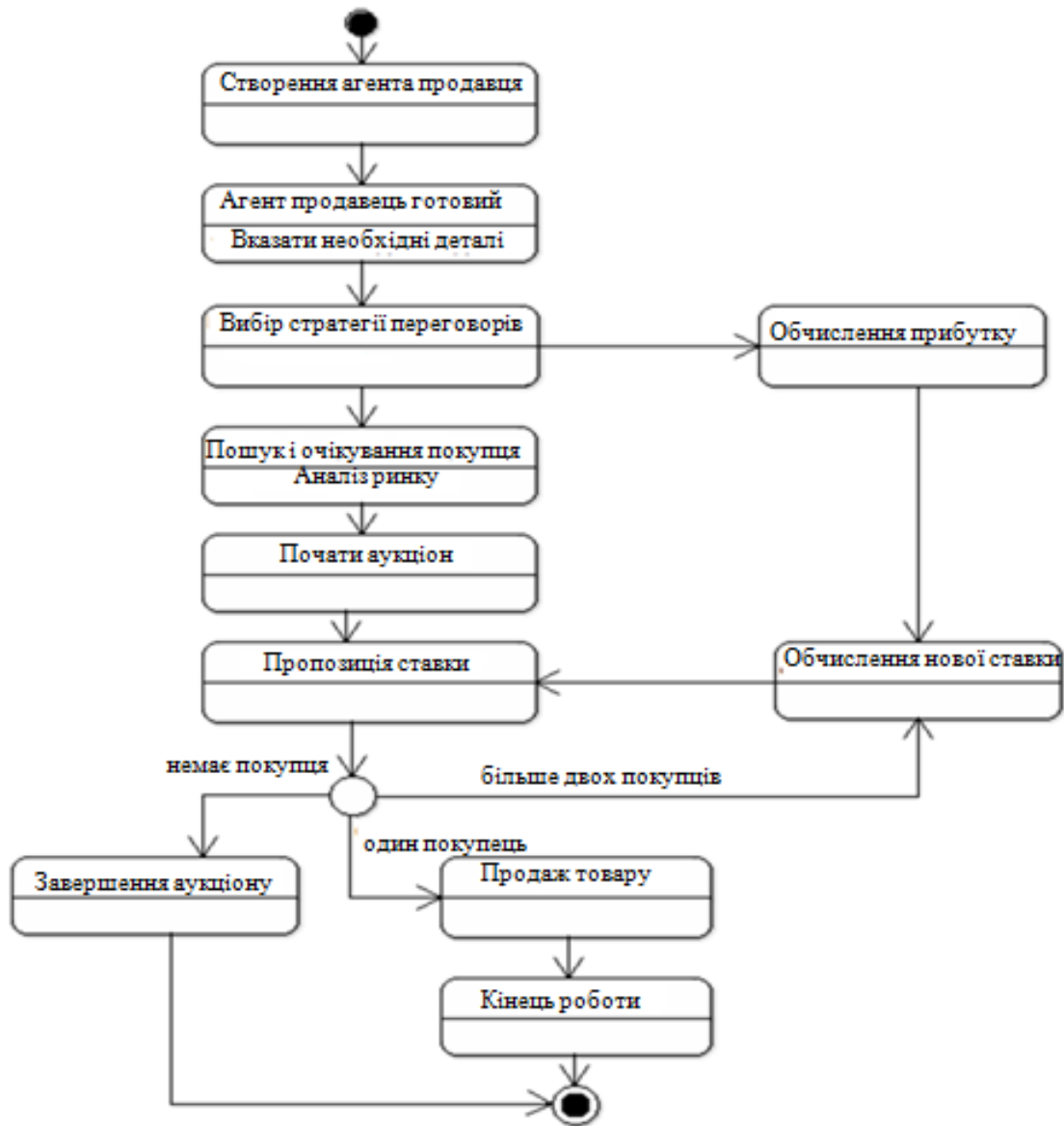


Рис.3. UML діаграма станів агента продавця

Агент покупець може відхилити або прийняти ставку. Якщо агент відхиляє ставку, він виходить з аукціону і починає пошук відповідного продавця, інакше він приймає ставку агента продавця.

Агент продавець періодично перевіряє наявність агентів покупців, які взяли ставки на товар. Якщо в аукціоні бере участь більше двох агентів, то аукціоніст пропонує нову ставку, при цьому збільшуючи значення ціни шляхом використання стратегії переговорів. Агент покупець, що виграв аукціон, може купити бажаний товар.

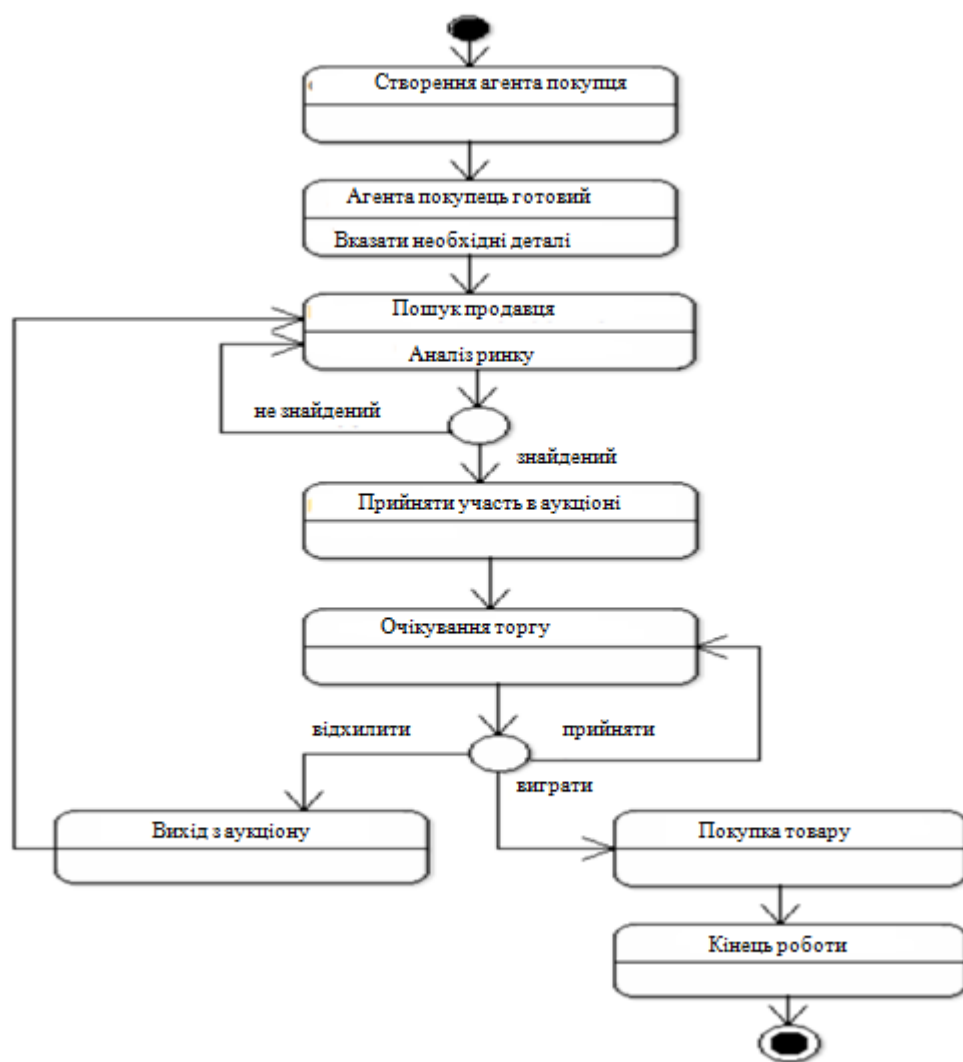


Рис.4. UML діаграма станів агента покупця

Четвертий рівень користувача - це той рівень, з яким безпосередньо взаємодіє користувач. Включає компоненти користувацького інтерфейсу, механізм введення від користувача. Стосовно до asp.net mvc на даному рівні розташовані всі компоненти, які складають призначений для користувача інтерфейс (стили, статичні сторінки html, javascript), а також моделі уявлень, контролери, об'єкти контексту запиту.

Імітаційне моделювання мультиагентної системи розподіленої торгової фірми

Реалізовано програмне забезпечення з використанням мультиагентної технології, яке виконує роботу по розподілу товарів між покупцями. Для цього

створено сцену з декількома діючими агентами продавців і агентами покупців, визначені властивості кожного агента, реалізовано запуск сцени на виконання і спостереження в динаміці матчінга між агентами продавців і агентами покупців.

Поведінка агента-продавця характеризується наступними параметрами:

- бажана дата, до якої необхідно продати товар;
- бажана ціна, по якій користувач хоче продати товар;
- мінімально допустима ціна;
- функція зниження ціни в часі (лінійна, квадратична);
- опис товару.

Агент-покупець має «симетричні» параметри:

- крайній термін покупки товару;
- бажана ціна покупки;
- максимально прийнятна ціна покупки;
- функція зростання ціни в часі;
- опис товару.

Торги ведуться за схемою закритого аукціону першої ціни. Поведінка агентів описується простою моделлю, в якій не використовуються знання і міркування. Агент-продавець, отримавши від мультиагентної системи інформацію про потенційних покупців, послідовно опитує їх усіх з метою прийняття рішення про можливість здійснення угоди. Угода укладається з першим агентом-покупцем, який готовий дати за товар запитувану ціну. Продавець не може вдруге вступити в контакт з покупцем, поки не опитає всіх потенційних покупців.

При кожному контакті агент-продавець веде переговори, пропонуючи початкову ціну, або знижуючи її. Агент-покупець діє аналогічним чином, відшукуючи продавців товару і пропонуючи їм свою ціну покупки, яку він може збільшувати в процесі переговорів. Будь-яка угода завершується тільки в разі її схвалення користувачем агента.

У сценарії діють 4 продавця і 4 покупця. Продавці можуть пропонувати за різними цінами як один, так і кілька навушників трьох найменувань: «Sony_WH-1000XM4», «TWS_OPPO_W31», «TWS_Xiaomi_Mi». Кожен покупець прагне купити один з 3-х товарів, причому двоє з них хочуть купити товар одного найменування. Кожному продавцю і кожному покупцеві зіставляється власний агент: агенти продавці продають товар з свого каталогу, агенти покупці купують

товари за дорученням своїх власників. Приклади наявності товарів у 4 компаніях із зазначенням ціни наведені в таблиці 1, індивідуальні призначення агентів вказані в таблиці 2.

Таблиця 1

Приклади наявності товарів у 4 компаніях

Найменування торгової компанії	Назва товару / ціна
WellMart	Sony_WH-1000XM4, 8000 грн TWS_OPPO_W31, 3200 грн
MiniMarket	Sony_WH-1000XM4, 8500 грн
HardwareMarket	Sony_WH-1000XM4, 8000 грн TWS_OPPO_W31, 1200 грн
АТВ	TWS_OPPO_W31, 2500 грн TWS_Xiaomi_Mi , 3000 грн

Таблиця 2

Індивідуальні призначення агентів

Найменування агента	Назва товару
BuyerSony_WH-1000XM4	Sony_WH-1000XM4
BuyerTWS_OPPO_W31	TWS_OPPO_W31
BuyerTWS_Xiaomi_Mi	TWS_Xiaomi_Mi
SecondBuyerSony_WH-1000XM4	Sony_WH-1000XM4

При розробці мультиагентної системи використана платформа JADE.

Кожен агент покупця отримує назву товару, який він повинен придбати в якості аргументу рядка і періодично запитує всіх відомих йому агентів продавців, щоб зробити запит про покупку. Як тільки пропозиція отримана, агент покупець підтверджує її та відправляє замовлення. Якщо більше ніж один агент продавець надає даний товар, агент покупець вибирає найкращу пропозицію (найкращу ціну). Купивши необхідний товар, агент покупець завершує роботу.

Кожен агент продавець має мінімальний інтерфейс, за допомогою якого користувач може додавати нові назви / ціну товарів в локальний каталог, виставлених на продаж. Агенти продавці знаходяться в стані очікування запитів від агентів покупців. Після отримання запиту на товар, вони перевіряють, чи є даний товар в їх каталозі: якщо так - то відповідають пропозицією зі

встановленою ціною, інакше - відмовляють. Коли агенти продавці отримують замовлення на покупку, вони обробляють його та видаляють запитаний товар зі свого каталогу.

Агент MarketAgent очікує запити від агентів покупців і обслуговує їх. Запити можуть бути двох типів: запит про подання пропозиції на товар і запит на виконання замовлення. Для реалізації цих запитів необхідно задати дві циклічні поведінки.

Середовище JADE надає можливість обмінюватися агентам між собою повідомленнями. Обмін повідомленнями є асинхронним і здійснюється за допомогою Message Transport System. Повідомлення, якими обмінюються між собою агенти JADE, мають формат мови ACL, визначеного FIPA як міжнародний стандарт взаємодії агентів.

Повідомлення може бути отримано з «поштової скриньки» агента і оброблено будь-яким з його режимів. Приклад обміну повідомленнями показаний на рисунку 6.

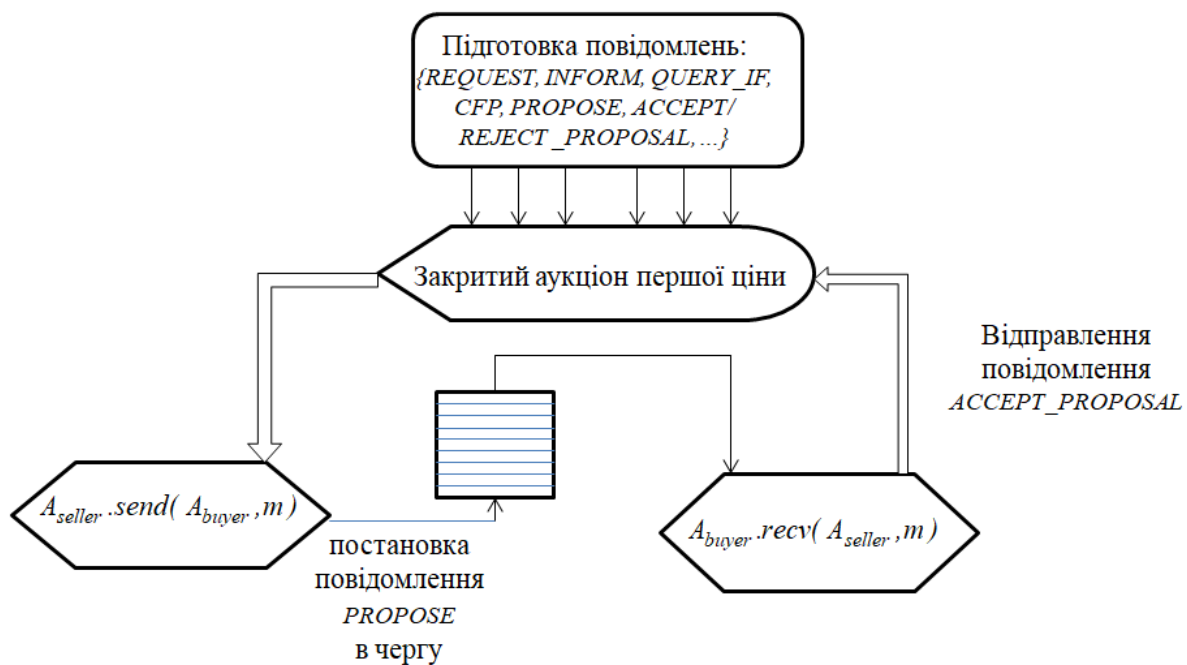


Рис.6. Приклад обміну повідомленнями в середовищі JADE

Тут закритий аукціон це послідовність повідомлень, якими обмінюються два або більше агентів з чітко визначеними причинними і тимчасовими відносинами.

Повідомлення пересилаються у вигляді Java-об'єктів. При відправці

повідомлення потрапляють в «поштову скриньку» (черга повідомлень) агента адресата, про що агент повідомляється. Повідомлення про пропозицію CFP використовується для сповіщення про те, що агент продавця HardwareMarket відправив агенту покупця *BuyerTWS_OPPO_W31* пропозицію на придбання товару. Повідомлення *PROPOSE* містить пропозицію продавця, включаючи вартість товару, а повідомлення *ACCEPT_PROPOSAL* містить згоду покупця. В обох типах повідомлень, що відправляються агенту покупця, міститься назва товару.

Агент покупця посилає запити агентам продавців періодично через певний інтервал часу. Приклад створення агента покупця наведено на рисунку 7. Після реєстрації всіх агентів покупців, вони вступають в процес матчінга. Кожен агент покупця знаходить найбільш підходящого продавця (що пропонує найменшу ціну за шуканий товар) і створює з ним зв'язок.

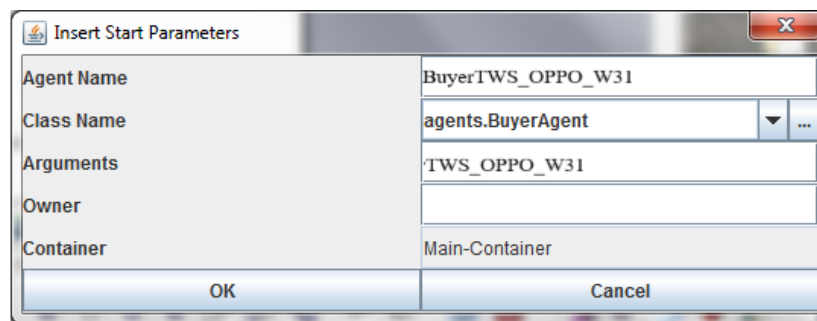


Рис. 7. Створення агента покупця **BuyerTWS_OPPO_W31**

Таким чином, нові агенти будуть і далі встановлювати зв'язки в порядку зростання ціни, на товар, поки не будуть задіяні всі варіанти. Результат матчінга представлений на рисунку 8.

```

Hello! Buyer-agent SecondBuyerSony_WH-1000XM4@192.168.1.3:1099/JADE is ready.
Target product is Sony_WH-1000XM4
Trying to buy Sony_WH-1000XM4
Found the following seller agents:
ATB@192.168.1.3:1099/JADE
WellMart@192.168.1.3:1099/JADE
MiniMarket@192.168.1.3:1099/JADE
HardwareMarket@192.168.1.3:1099/JADE
Sony_WH-1000XM4 sold to agent SecondBuyerSony_WH-1000XM4@192.168.1.3:1099/JADE
Sony_WH-1000XM4 successfully purchased from agent HardwareMarket@192.168.1.3:1099/JADE
Price = 1000
Buyer-agent SecondBuyerSony_WH-1000XM4@192.168.1.3:1099/JADE terminating.

```

Рис.8. Результат матчінга покупця **SecondBuyerSony_WH-1000XM4**

Виходячи з прикладу, наведеного на рисунку 8, можна зробити висновок, що покупець знайшов найвигіднішу пропозицію - купив необхідний товар у агента HardwareMarket, так як товар агента WellMart вже проданий.

Висновок

Пропонується багатоагентна система для планування та координації групових дій агентів, проведення переговорів та вибору оптимального рішення.

На основі запропонованої архітектури була розроблена мультиагентна система розподіленої фірми. Запропонована архітектура підвищує якість автоматизованого управління, надаючи додаткові послуги. За допомогою платформи JADE створені агенти покупця і агенти продавця, які можуть взаємодіяти між собою, що дозволяє ефективно обробляти запити і керувати продажем товарів в різних торгових точках компанії.

У програмі реалізовані основні функції агентів продавців, які полягають в продажу свого товару з каталогу, і агентів покупців - придбати товари за дорученням своїх власників за найменшою ціною.

Реалізоване програмне забезпечення підтверджує доцільність застосування розробленої МАС розподіленої фірми.

Література

1. Axak N. Cloud-fog-dew Architecture for Personalized Service-oriented Systems / N. Axak, D. Rosinskiy, O. Barkovska, I. Novoseltsev // The 9th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT'2018, Kyiv, Ukraine – 2018. – P. 80–84.

2. Axak N. G. Development of multi-agent system of neural network diagnostics and remote monitoring of patient. /N. G. Axak //Eastern-European Journal of Enterprise Technologies. – 2016. - 4/9 (82) – P. 4-11.

3. Аксак Н.Г. Концепція побудови мультиагентних систем розподіленої нейромережевої обробки великих даних / Н.Г. Аксак //ВІСНИК ХНТУ №3(66), - ТОМ 1, - 2018 р. – С. 205-212.

ГЛАВА 3

СТВОРЕННЯ ПРИСТРОЇВ КОНТРОЛЮ І ЛОКАЛІЗАЦІЇ ПОМИЛОК В РОЗПОДІЛЕНИХ СИСТЕМАХ

Вступ і постановка задачі. Дедалі більшого поширення набувають нові технології, пов'язані з обробкою і передачею інформації різного виду по каналах зв'язку. Інтеграція різних видів інформаційного обслуговування в рамках однієї мережі є закономірним наслідком розвитку цифрових технологій. Розподілений характер великої зі складною структурою мережі унеможливорює підтримання її роботи на належному рівні без системи контролю та управління, яка також є складною системою [1]. При цьому важливе значення в розвитку інформаційної технології в різних системах управління має специфіка вирішуваних завдань цими інформаційними системами. Головною особливістю систем управління, які використовують сучасні інформаційні технології, є задоволення вимог по живучості, стійкості, скритності, мобільності, гнучкості управління і якості обслуговування користувачів. Задоволення цих вимог призводить до необхідності як ускладнення архітектури мережі, так і сукупність розв'язуваних завдань. В даний час система управління працює в автоматизованому режимі, виконуючи найпростіші дії з управління мережею автоматично, а складні рішення надаючи приймати людині на основі підготовленої системою інформації [2]. Система управління мережею повинна забезпечити, з одного боку, підтримку в робочому стані як мережу в цілому, так і окремих її складових, для того, щоб вона могла виконувати свої функції; а з іншого боку, - розподіл і доставку інформаційних повідомлень за адресами з дотриманням різних вимог користувачів [3]. Для функціонування системи управління необхідна інформація про стан елементів мережі. Оскільки ця інформація надходить з тієї ж мережі, яка управляється, може виникнути ситуація, коли вона буде неповною або неточною. Аналіз робіт показує, що в даний час особливо застосовуваних систем мережевого управління є низька ефективність застосовуваних методів збору інформації, прийняття рішення і контролю виконання цього рішення. В результаті знижується якість обслуговування користувачів. Тому для виконання вимог щодо якості обслуговування розробники мереж в процесі проектування змушені передбачати певну надмірність при виконанні будь-яких операцій в

процесі управління. Це призводить до збільшення вартості як самої мережі, так і послуг, що надаються. Усунути цей недолік можна шляхом підвищення ефективності технології управління мережею на основі розробки і впровадження достовірних методів контролю і діагностики переданої інформації. Цієї мети можна досягти шляхом розробки універсальних програмних, апаратних або програмно-апаратних засобів контролю повідомлень, що передаються в розподілених системах.

В умовах зростання вимог до зменшення часу обміну даними, що передаються між об'єктами, а також підвищення достовірності доставки даних абонентам є актуальною задачею розроблення простих і ефективних засобів контролю як процесу передачі інформації, так і обладнання в розподілених системах. Основна мета роботи є удосконалення процесу створення універсальних, швидкодіючих багатоканальних пристроїв, які дозволяють локалізувати помилки в інформаційних повідомленнях і формувати дані для діагностування обладнання.

Основна частина

Процес отримання сигнатури для вхідної послідовності $\alpha(t)$ можна представити з допомогою виразу [1]:

$$\text{Sig } \alpha(t) = \sum_{i=1}^p S^i \cdot \alpha_i \cdot h_i, \quad (1)$$

де \sum – сума за модулем два; α_i – i -й елемент вхідної послідовності; p – кількість елементів вхідної послідовності; h_i – i -й стовбець матриці станів сигнатурного аналізатора, який визначається по формулі:

$$h_i = S^i \cdot h_0, \quad i=0,1,\dots,z, \quad (2)$$

де z – кількість стовбців матриці станів; $h_0 = \parallel 10\dots0 \parallel^T$ – нульовий стовбець матриці станів; S – супроводжуюча матриця, яка однозначно описує характеристичний поліном [1]:

$$P(x) = b_{1n} \cdot x^n \oplus b_{1n-1} \cdot x^{n-1} \oplus \dots \oplus b_{1i} \cdot x^i \oplus b_{12} \cdot x \oplus b_{11}, \quad (3)$$

де $b_{1i} \in \{0,1\}$ – коефіцієнти характеристичного поліному.

$$\text{Sig1 } \alpha(t) = \sum_{j=0}^z \text{Sig } \alpha_j(t) \cdot S^{j-1}, \quad (7)$$

Таким чином, отримано два перевірочних кодових слова або дві сигнатури, обчислення яких здійснюється відповідно до виразів (6) та (7).

Кожної з сигнатур потрібно мати дві еталонні сигнатури E_1 і E_2 , які складаються із сукупності еталонів згорток вхідних послідовностей по кожному входу пристрою:

$$E_1 = \sum_{j=1}^z e_j; \quad (8)$$

$$E_2 = \sum_{j=1}^z e_j^1,$$

де $e_i^1 = e_i \cdot S^{j-1}$.

У процесі аналізу вихідних реакцій пристроїв які перевіряються визначимо два синдроми помилки Φ та Ψ відповідно до виразів (7), (8), які при відсутності помилок обидва дорівнюють нулю:

$$\Phi = \sum_{j=0}^z (\text{Sig } \alpha_j(t) + e_j); \quad (9)$$

$$\Psi = \sum_{j=0}^z (S^{j-1} \cdot \text{Sig } \alpha_j(t) + e_j^1).$$

Припустимо, що виникла помилка, що виявляється, на j -м вході сигнатурного аналізатора. Тоді виникнуть зміни в сигнатурі g_i , а синдроми помилки будуть визначатися по формулах:

$$\begin{aligned} \varphi_j &= \text{Sig } \alpha_j(t) + e_j \neq 0; \\ \psi_j &= S^{j-1} \cdot \text{Sig } \alpha_j(t) + e_j^1 \neq 0; \end{aligned} \quad (10)$$

або

$$\begin{aligned}\varphi_j &= \text{Sig } \alpha_j(t) + e_j; \\ \psi_j &= S^{j-1} \cdot \text{Sig } \alpha_j(t) + S^{j-1} \cdot e_j = S^{j-1} \cdot (\text{Sig } \alpha_j(t) + e_j).\end{aligned}\tag{11}$$

З вираження (10) видно, що синдроми помилок, що виникли на i -м вході зв'язані між собою співвідношенням:

$$\psi_j = \varphi_j \cdot S^{j-1}.\tag{12}$$

Таким чином, для того, щоб обидва синдроми помилки збіглися необхідно один з них помножити на S^{i-1} або виконати $i-1$ тактів зрушення в регістрі, що зрушує, сигнатурного аналізатора. Кількість таких тактів зрушення синдрому ψ покаже номер входу сигнатурного аналізатора, до якого надходять помилки.

Розглянемо приклад. Відповідно (4) матриця станів формувача сигнатур для характеристичного поліному $P(x) = x^4 \oplus x^3 \oplus 1$ має вигляд, який наведений в табл. 1.

Таблиця 1

Матриця станів формувача сигнатур для характеристичного поліному
 $P(x) = x^4 \oplus x^3 \oplus 1$

Номер рядка	Номер стовбця														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0
2	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0
3	0	0	1	0	0	1	1	0	1	0	1	1	1	1	0
4	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1
Послідовність	1	0	0	1	1	1	0	0	1						
Послідовність із зсувом	1	0	0	0	1	1	1	0	0	0	1				

Нехай до сигнатурного аналізатора подається наступна послідовність 100111001, де молодший розряд подається першим. Еталонна сигнатура для цієї послідовності $E_1 = 0001$. Припустимо, що контролюється пристрій, який має три виходи, з кожного виходу сигнатурний аналізатор перевіряє послідовність довжиною три розряди. Тоді, з допомогою додаткових зсувів у формувачі сигнатур, визначимо послідовність 100 0 111 0 001. Еталонна сигнатура для цієї послідовності $E_2 = 0111$ (табл. 1).

Припустимо, що на другому вході пристрою у другому біті виникла помилка, тоді вхідні послідовності зміняться на наступні: 100111001; 100 0 101 0 001. Процес формування сигнатури відображений в табл.2. Реальні сигнатури, які будуть отримані для цих послідовностей відповідно дорівнюють: $\Phi = 1101$; $\Psi = 0001$.

Визначимо синдроми помилок відповідно (10):

$$\varphi = \Phi \oplus E_1 = 1101 \oplus 0001 = 1100;$$

$$\psi = \Psi \oplus E_2 = 0001 \oplus 0111 = 0110.$$

Синдроми, які отримані є відповідно п'ятим та шостим стовбцями матриці станів сигнатурного аналізатора. Для того щоб ці синдроми співпали треба виконати зсув синдрому φ у регістрі сигнатурного аналізатора або помножити його на матрицю S відповідно виразу (12).

Структурна схема пристрою, що реалізує процес локалізації помилок зображена на рис.1. Сигнатурний аналізатор включає: мультиплексор 1, три логічні схеми AND 2, 9, 13, два формувача сигнатур 3, 10, блок порівняння сигнатури з еталоном 4, комутатор адреси 5, два лічильника 6, 22, дільник частоти 7, логічну схему АБО 8, блок індикації 11, тригер 12, блок управління 14, група інформаційних входів 15, група входів завдання номера каналу 16, група входів завдання режиму роботи 17, група входів запису кількості входів пристрою що перевіряються 18, вхід пуску 19, група входів запису еталону 20 та група входів встановлення інтервалу перевірки кожного входу приладу 21.

Процес формування сигнатури при виникненні помилки

Номер рядка	Номер стовбця														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0
2	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0
3	0	0	1	0	0	1	1	0	1	0	1	1	1	1	0
4	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1
Послідовність	1	0	0	1	0	1	0	0	1						
Послідовність із зсувом	1	0	0	0	1	0	1	0	0	0	1				

Пристрій (рис.1) працює наступним чином. В початковому стані на групу входів 17 подається код завдання режиму роботи, при необхідності перевірки одного каналу на входи 16 подається код номера каналу, з групи входів 18 до лічильника 6 записується код відповідний кількості входів пристрою n, які перевіряються, з групи входів 20 до регістрів зберігання еталону 24, 27 (рис.2) блока порівняння сигнатури з еталоном 4 записується код відповідний двом еталонним сигнатурам (8), до дільника частоти 7 з групи входів 21 записується код відповідний кількості тактів перевірки кожного входу приладу, до регістрів формувачів сигнатур 3 і 10, лічильника 22, а також регістрів зберігання сигнатури 22, 26 блока порівняння сигнатури з еталоном 4 записується код 00...0, тригер 12 встановлюється в нульовий стан (ланцюги встановлення приладу в початковий стан не наведені).

Сигнал пуск з входу 19 пристрою запускає генератор 14, змінює стан тригера 12, який дозволяє збирати данні з n-го входу мультиплексора 1 (адреса входу мультиплексора встановлена в лічильнику 6) і подавати їх через перший логічний елемент AND 2 до інформаційних входів формувачів сигнатур 3, 10. Крім цього новий стан тригера 12 дозволяє проходження сигналів синхронізації

з генератора 14 до входу дільника частоти 7 через логічний елемент AND 13, а також до входу синхронізації другого формувача сигнатур 10 через логічний елемент AND 9. Інформація з n-го входу пристрою буде збиратися доки дільник частоти 7 рахує сигнали тактового генератора 14. При цьому, блоки 3 та 10 виконують згортку цих даних.

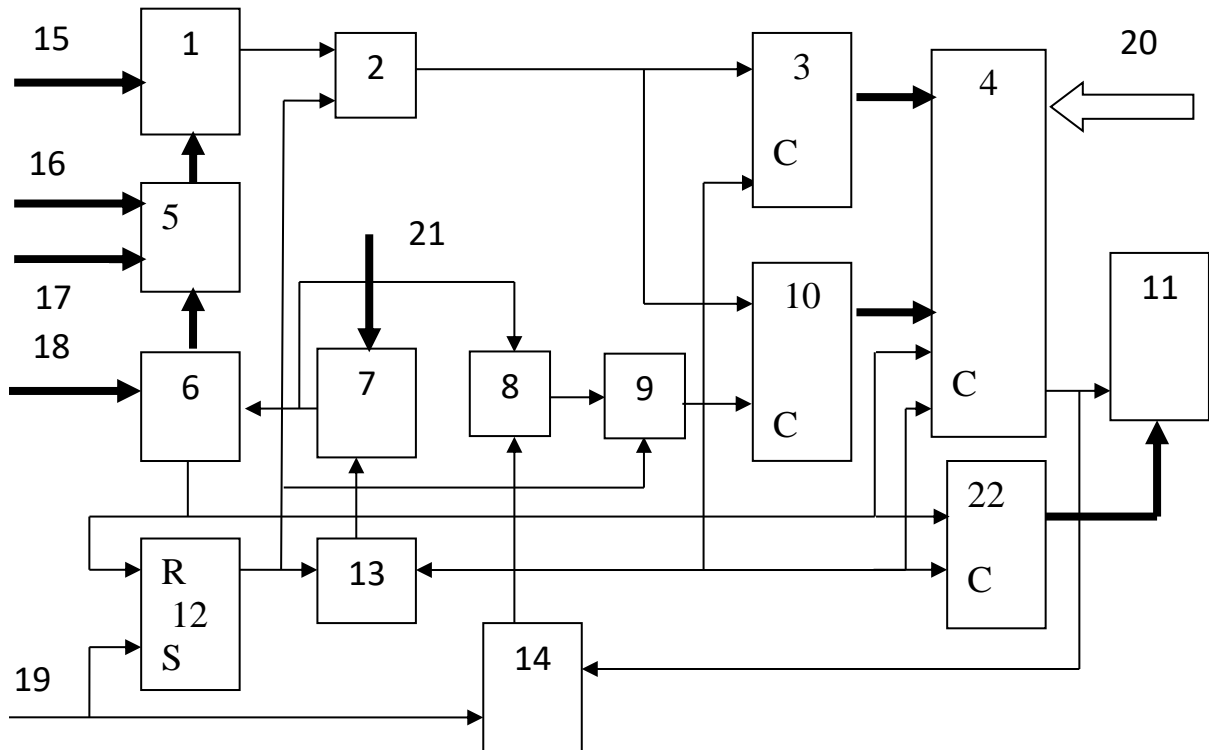


Рис.1. Структурна схема сигнатурного аналізатора

Коли кількість сигналів досягне кількості тактів перевірки кожного входу приладу (які записані з групи входів 21), блок 7 сформує імпульс, який змінить стан лічильника 6 і задасть нову адресу входу (n-1) з групи інформаційних входів 15. При цьому, дільник частоти 7 знову запише данні з групи входів 21 пристрою. Сигнал з дільника частоти 7 через логічні елементи 8, 9 виконає додатковий зсув даних, які зберігаються у формувачі сигнатур 10 відповідно виразу (7). Послідовне прочитування інформації з входів 15 виконується до появи сигналу на інверсному виході лічильника 6, який свідчить про те, що всі входи перевірені. Цей сигнал змінює стан тригера 12 на нульовий, забороняючи надходження сигналу синхронізації з блоку управління 14 на вхід дільника частоти 7, вхід синхронізації другого формувача сигнатури 10, а також закриває доступ даних

через перший логічний елемент AND 2 на інформаційні входи формувачів сигнатур 3, 10.

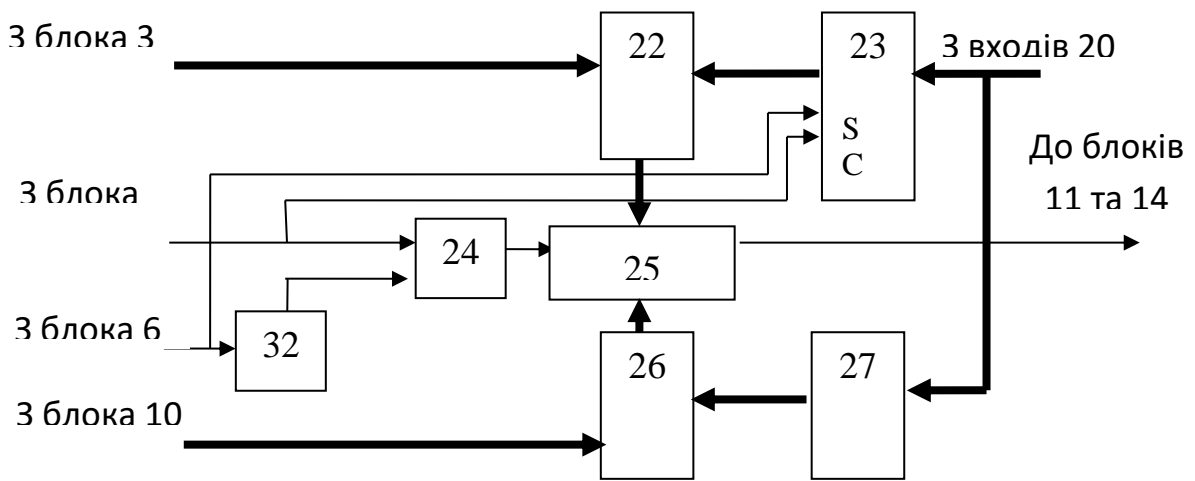


Рис.2. Структурна схема блока порівняння сигнатури з еталоном

Таким чином, прочитування даних з входів 15 завершено, при цьому в регістрах формувачів сигнатур 3, 10 зберігаються згортки, які одержані відповідно до виразів (8) і (9). Блок 4 виконує порівняння одержаних сигнатур з еталонними значеннями і, у разі їх збігу (відсутність помилок), формує на першому виході сигнал високого рівня, який дозволяє відобразити вміст лічильника 22 в блоці індикації 11 і припиняє генерацію імпульсів синхронізації блоком управління 14. В процесі прочитування даних з входів 15 пристрою лічильник 22 і формувач сигнатур 24 (рис. 2) блоку порівняння 4 знаходилися в режимі зберігання інформації, тому імпульси синхронізації з виходу блоку 14 не змінювали їх стан.

При завершенні тестування сигнал з інверсного виходу лічильника 6 переведе лічильник 22 в режим рахунку, а формувач сигнатур 24 в режим зрушення інформації, що зберігається в ньому. Таким чином, за відсутності помилок у вхідній послідовності два синдроми помилок матимуть нульове значення і блок індикації 11 відобразить код 00.0, відповідний нульовому стану лічильника 22.

В тому випадку, якщо у вхідній послідовності є помилки, які виникли на одному з входів 15 пристрою, то в результаті порівняння сигнатур з еталонами на виході блоку 4 залишається сигнал низького рівня і блок управління 14 продовжує генерацію імпульсів синхронізації, які поступають на вхід синхронізації формувача сигнатур 3 (рис.1) і вхід синхронізації формувача сигнатур 24 (рис.2) блоку порівняння 4, при цьому виконується зрушення, інформації, що зберігається в них, відповідно до виразу (12). У блоці 26 (рис. 2) виконується порівняння синдромів помилки. Лічильник 22, переведений в стан рахунку імпульсів синхронізації сигналом з інверсного виходу лічильника 6, підраховує кількість таких зрушень. Підрахунок імпульсів синхронізації триватиме до збігу синдромів помилки в блоці 4 відповідно до виразу (12). У лічильнику 22 остається номер входу, з групи входів 15 пристрою, який, за допомогою сигналу високого рівня з виходу блоку 4, буде відображений в блоці індикації 11. Якщо у вхідній послідовності є помилки, які виникли на декількох входах 15 пристрою, то блок індикації 11 відобразить число більше величини n (кількості входів пристрою, що перевіряються).

Блок порівняння сигнатури з еталонем 4 (рис.2) працює таким чином. Блоки 24 і 29 призначені для зберігання еталонів сигнатур. Суматори по модулю два 23 і 28 використовуються для визначення синдромів помилок. Блок 26 призначений для порівняння синдромів помилок. Спочатку роботи пристрою в регістр 29 і формувач сигнатур 24 записуються значення, які відповідають двом еталонним сигнатурам (8). В процесі збору інформації блок 24 знаходиться в режимі зберігання даних. При завершенні тестування сигнал з інверсного виходу лічильника 6 переведе формувач сигнатур 24 в режим зрушення, інформації, що зберігається в ньому. Крім того, цей сигнал дозволить сформувати одиночний імпульс блоком 27, який через схему АБО поступить на вхід запуску генератора одиночного імпульсу 32. Якщо на вході управління блоку 32 присутній сигнал високого рівня, який забороняє генерацію імпульсів, то на виході блоку порівняння сигнатури з еталонем 4 буде відсутній сигнал. Це означає, що у вхідній послідовності є помилки. Пошук входу пристрою, на який поступила спотворена інформація здійснюється на основі виразу (14) шляхом зрушення даних, що знаходяться у формувачах сигнатур 3 (рис.1) і 24 (рис.2). Як тільки синдроми помилок на виходах блоків 23 і 28 (або входах блоку 30) співпадуть, то це дозволить сформувати імпульс на виході блоку 32. Цей імпульс зупинить

генерацію сигналів синхронізації блоком 14 і дозволить відображення діагностичної інформації в блоці індикації 11.

За відсутності помилок у вхідній послідовності два синдроми помилок матимуть нульове значення і блок індикації 11 відобразить код 00.0, відповідний нульовому стану лічильника 22 (рис.1).

Синтез багатоканальних пристроїв на основі формул (3-12) можливо над будь-яким полем Галуа. Однак при створенні пристроїв для діагностування помилок є особливості і обмеження, які впливають на архітектуру сигнатурних аналізаторів. Розглянемо процес створення діагностичного пристрою на базі характеристичного поліному $P(x) = x^4 \oplus_3 x^3 \oplus_3 1$, де \oplus_3 – сума за модулем 3. Відповідно (1) матриця станів для перших дванадцяти стовбців поліному $P(x) = x^4 \oplus_3 x^3 \oplus_3 1$ має вигляд, який наведений в табл.. 3.

Таблиця 3

Матриця станів для характеристичного поліному $P(x) = x^4 \oplus_3 x^3 \oplus_3 1$

Номер рядка	Номер стовпця											
	0	1	2	3	4	5	6	7	8	9	10	11
1	1	0	0	1	1	0	1	2	1	1	0	0
2	0	1	0	0	1	1	0	1	2	1	1	0
3	0	0	1	0	0	1	1	0	1	2	1	1
4	0	0	0	1	0	0	1	1	0	1	2	1

Процес отримання сигнатури для вхідної послідовності $v(t)$ можна представити з допомогою виразу:

$$\text{Sig}\alpha(t) = \sum_{i=0}^n {}_3S^i \otimes_3 v_i \otimes_3 h_i, \quad (13)$$

де \sum_3 – сума за модулем три; \otimes_3 – множення за модулем 3; v_i – i -й елемент вхідної послідовності.

Матриця станів сигнатурного аналізатора [1,2] може бути побудована і другим способом. У перший тригер регістра приладу [1] записується 1, а в інші – всі нулі ($h_0 = \parallel 10\dots 0 \parallel$). Це перший стан регістра, який є першим стовпцем матриці станів. Після цього послідовно проводяться зсуви попереднього стану та їх збереження. Зсуви відбуваються з урахуванням попереднього стану завдяки зворотнім зв'язкам відповідно до ступенів утворюючого полінома.

У процесі стиснення вхідної послідовності в кожному з її елементів можуть виникати такі помилки:

замість значення елемента послідовності 0 отримана 1 (0 -> 1);

замість значення елемента послідовності 0 отримана 2 (0 -> 2);

замість значення елемента послідовності 1 отриманий 0 (1 -> 0);

замість значення елемента послідовності 1 отримана 2 (1 -> 2);

замість значення елемента послідовності 2 отримано 0 (2 -> 0);

замість значення елемента послідовності 2 отримана 1 (2 -> 1).

Перераховані помилки можна розділити на дві групи. Перша група помилок, в якій спотворений елемент вхідної послідовності відрізняється від еталонного значення на 1. До цієї групи можна віднести наступні помилки: 0 -> 1, 1 -> 0, 1 -> 2, 2 -> 1. При спотворенні і-го елемента вхідної послідовності еталонна сигнатура E буде відрізнятися від реально отриманої сигнатури Sig $\alpha(t)$ на і-й стовпець матриці станів H і є ознакою або синдромом помилки. При цьому якщо сталася помилка виду 0 -> 1 або 1 -> 2, синдром помилки можна визначити за виразом:

$$h_i = \text{Sig } \alpha(t) - E. \quad (14)$$

У формулі (14) і далі операції віднімання і складання виконуються за модулем три. Якщо сталася помилка виду 1 -> 0 або 2 -> 1, то синдром помилки визначається відповідно до виразу:

$$h_i = E - \text{Sig } \alpha(t). \quad (15)$$

Друга група помилок, в якій спотворений елемент вхідної послідовності відрізняється від еталонного значення на 2. До цієї групи можна віднести наступні помилки: 0 -> 2, 2 -> 0. При виникненні такого виду помилок синдром h_i можна визначити за формулою:

$$h_i \otimes_3 2 = \text{Sig } \alpha(t) - E \quad (16)$$

для помилки 0 -> 2 або відповідно до виразу:

$$h_i \otimes_3 2 = E - \text{Sig } \alpha(t) \quad (17)$$

для помилки 2 -> 0. Таким чином, визначивши синдром помилки за допомогою виразів (14) - (17), можна встановити номер елемента вхідної послідовності, в якому ця помилка сталася. Покажемо процес отримання сигнатури та визначення помилки на багатоканальному аналізаторі який має вісім каналів. Припустимо сигнатурний аналізатор обробляє послідовність $\alpha(t)_{GF3} = 10201002$. Процес отримання сигнатури для цієї послідовності наведений в табл. 4.

Таблиця 4

Процес отримання сигнатури послідовності

Регістр	Матриця станів								Сигнатура
9_1	1	0	0	1	1	0	1	2	0
9_2	0	1	0	0	1	1	0	1	0
9_3	0	0	1	0	0	1	1	0	2
9_4	0	0	0	1	0	0	1	1	2
$\alpha(t)_{GF3}$	1	0	2	0	1	0	0	2	

Припустимо, що помилка виникає в шостому розряді вхідної послідовності, тоді $\alpha(t)_{GF3} = 10201102$ або $\alpha(t)_{GF3} = 10201202$. Сигнатури, які отримані при виникненні таких помилок визначаються в табл. 5 та табл. 6 відповідно.

Таблиця 5

Процес отримання сигнатури послідовності з помилкою

Регістр	Матриця станів								Сигнатура
9_1	1	0	0	1	1	0	1	2	0
9_2	0	1	0	0	1	1	0	1	1
9_3	0	0	1	0	0	1	1	0	0
9_4	0	0	0	1	0	0	1	1	2
$\alpha(t)_{GF3}$	1	0	2	0	1	1	0	2	

Таблиця 6

Процес отримання сигнатури послідовності з помилкою

Регістр	Матриця станів								Сигнатура
9_1	1	0	0	1	1	0	1	2	0
9_2	0	1	0	0	1	1	0	1	2
9_3	0	0	1	0	0	1	1	0	1
9_4	0	0	0	1	0	0	1	1	2
$\alpha(t)_{GF3}$	1	0	2	0	1	2	0	2	

Визначимо синдроми можливих помилок. Синдром помилки вигляду $0 \rightarrow 1$ є стовпець $h_5 = \parallel 0110 \parallel^T$. Синдром помилки вигляду $0 \rightarrow 2$ є стовпець $h_5 \otimes_3 2 = \parallel 0220 \parallel^T$. Відповідно до виразів (5) і (7) набудемо значення еталонної сигнатури:

$$E = \text{Sig } \alpha_1(t) - h_i = \|0102\|^T - \|0110\|^T = \|0022\|^T; \quad (18)$$

$$E = \text{Sig } \alpha_2(t) - h_i \otimes_3 2 = \|0212\|^T - \|0220\|^T = \|0022\|^T, \quad (19)$$

де $\text{Sig } \alpha_1(t)$ – сигнатура, яка містить помилку, що належить першій групі помилок, $\text{Sig } \alpha_2(t)$ – сигнатура, яка містить помилку, що належить другій групі помилок.

При виникненні помилок вигляду $1 \rightarrow 0$ і $1 \rightarrow 2$ еталонною сигнатурою можна вважати результат згортки з табл. 3, а помилковими – результати з табл.2 і табл.4. Тоді, відповідно до (14) і (15), і по аналогії з (18), набудемо наступного значення еталонних сигнатур:

$$E = \text{Sig } \alpha_1(t) + h_i = \|0022\|^T + \|0110\|^T = \|0102\|^T; \quad (20)$$

$$E = \text{Sig } \alpha_2(t) - h_i = \|0212\|^T - \|0110\|^T = \|0102\|^T. \quad (21)$$

При виникненні помилок вигляду $2 \rightarrow 1$ і $2 \rightarrow 0$ еталонною сигнатурою можна вважати результат згортки з табл. 4, а помилковими - результати з табл.2 і табл.3. Тоді, відповідно до (14) і (15), і по аналогії з (18) та (19), набудемо наступного значення еталонних сигнатур:

$$E = \text{Sig } \alpha_1(t) + h_i = \|0102\|^T + \|0110\|^T = \|0212\|^T; \quad (22)$$

$$E = \text{Sig } \alpha_2(t) + h_i \otimes_3 2 = \|0022\|^T + \|0220\|^T = \|0212\|^T. \quad (23)$$

Таким чином, процес визначення елемента вхідної послідовності, що містить помилку, полягає в обчисленні еталонного значення сигнатури на основі пошуку необхідного синдрому помилки. Синдром помилки h_i може визначатися за допомогою послідовного зрушення першого стовпця $h_0 = \|10\dots0\|$ матриці станів N сигнатурного аналізатора [1]. При цьому кількість зрушень, виконаних в пристрої [1] до моменту збігу з еталоном повинно відповідати номеру помилкового елемента вхідної послідовності. Якщо кількість зрушень, виконаних в пристрої [1] у момент збігу з еталоном перевищує довжину вхідної послідовності, то ухвалюється рішення про наявність помилок в декількох її елементах або виникненні багатократної помилки.

Хай вхідна послідовність вводиться в сигнатурний аналізатор, який має K входів. Для того, щоб визначати вхід, до якого поступає помилковий елемент треба порівнювати еталонну сигнатуру з реальною та синдромом помилки відповідно виразам (14) – (21).

На рис.3 наведена структурна схема пристрою в загальному виді. Пристрій включає: K інформаційних входів 1; групу шифраторів $2_1 - 2_K$; групу логічних схем АБО $3_1 - 3_K$; групу блоків $4_1 - 4_n$ множення на два за модулем три; першу групу $5_1 - 5_n$ суматорів за модулем три; три логічні схеми AND 6, 7, 18; генератор одиночного імпульсу 8; дві групи дворозрядних регістрів $9_1 - 9_n$, $19_1 - 19_n$; блок синдрому помилки 10; схема множення на два за модулем три 11; другу групу з двох суматорів за модулем три $12_1, 12_2$; групу з двох блоків віднімання за модулем три $13_1, 13_2$; групу з чотирьох блоків порозрядного порівняння $14_1 - 14_4$; регістр 15; логічна схема АБО 16; лічильник 17; вхід діагностування 20; вхід синхронізації приладу 21; вхід запуску приладу 22; входи еталону 23.

В початковому стані в регістрах $9_1 - 9_n$, $19_1 - 19_n$, 15, регістрах блока синдрому помилки 10 (рис.3) записано код 0...0 (ланцюги встановлення в початковий стан не наведені). На входи 1 аналізатору паралельно подається перша група з K розрядів інформаційної послідовності з виходу об'єкта контролю, яка є реакцією на вхідні тестові дії, а на входах 23 пристрою встановлюється код еталону. На вхід 22 подається сигнал запуску пристрою, який дозволяє за сигналом синхронізації з виходу блоку 6 (рис.3) записати в регістри $9_1 - 9_n$ результат згортки ($\text{Sig } \alpha(t)$) вхідного паралельного потоку даних, що й є сигнатурою відповідно пристрою [4]. Той же сигнал синхронізації записує до регістрів $19_1 - 19_n$ код еталону (E). блок синдрому помилки $14_1 - 14_4$. Оскільки в регістрі блока синдрому помилки 10 записаний код 00...0, то на другі входи схем порівняння $14_1 - 14_4$ подаватиметься код згортки. Кожна з схем $14_1 - 14_4$ порівнює одержану сигнатуру з кодом еталону і формує на виході рівень логічного 0 при збігу реальної сигнатури з еталоном або логічну 1 при їх неспівпаданні. Перший сигнал синхронізації записує набутої ознаки збігу реальної і еталонної сигнатур в регістр 15. Якщо у вхідній послідовності відсутні помилки, то в регістрі 15 буде зафіксований код ознаки помилки 0000 і ухвалюється рішення про правильний прийом інформації, що перевіряється.

Обробка даних з допомогою пропонованого приладу виконується наступним чином. На першому такті сигнатурний аналізатор приймає

послідовність 2022 ($K = 4$), при цьому в регістрах $9_1 - 9_n$ буде сформована тимчасова сигнатура $\text{Sig } \alpha(1) = 1022$. На другому такті роботи приладу на входи 1 сигнатурного аналізатора подається послідовність 2101, сигнатура якої (0101) додається до сигнатури $\text{Sig } \alpha(1) \otimes_3 S^4$ з допомогою суматорів за модулем три $5_1 - 5_4$. При цьому в регістрах $9_1 - 9_n$ буде зафіксовано результат $\text{Sig } \alpha(2) = 2222$ ($1022 \otimes_3 S^4 = 2121$, $0101 \oplus_3 2121 = 2222$). На третьому такті роботи приладу на входи 1 сигнатурного аналізатора подається послідовність 1020, сигнатура якої (1020) додається до сигнатури $\text{Sig } \alpha(2) \otimes_3 S^4$. При цьому в регістрах $9_1 - 9_n$ буде зафіксовано наступний результат:

$$\text{Sig } \alpha(3) = 1101 (2222 \otimes_3 S^4 = 0111, 1020 \oplus_3 0111 = 1101).$$

Таким чином, нелінійний багатоканальний сигнатурний аналізатор здатний обробляти інформацію довжиною, яка перевищує кількість інформаційних входів пристрою та може передаватись по групах з одержанням сигнатури, яка дорівнює сигнатурі одноканального сигнатурного аналізатора при використанні одного й того утворюючого поліному. Отримана сигнатура багатоканального сигнатурного аналізатора, який пропонується завжди буде дорівнювати сигнатурі з одноканального сигнатурного аналізатору, тому, що матриця станів, а також супроводжуюча матриця залежать тільки від вигляду утворюючого поліному.

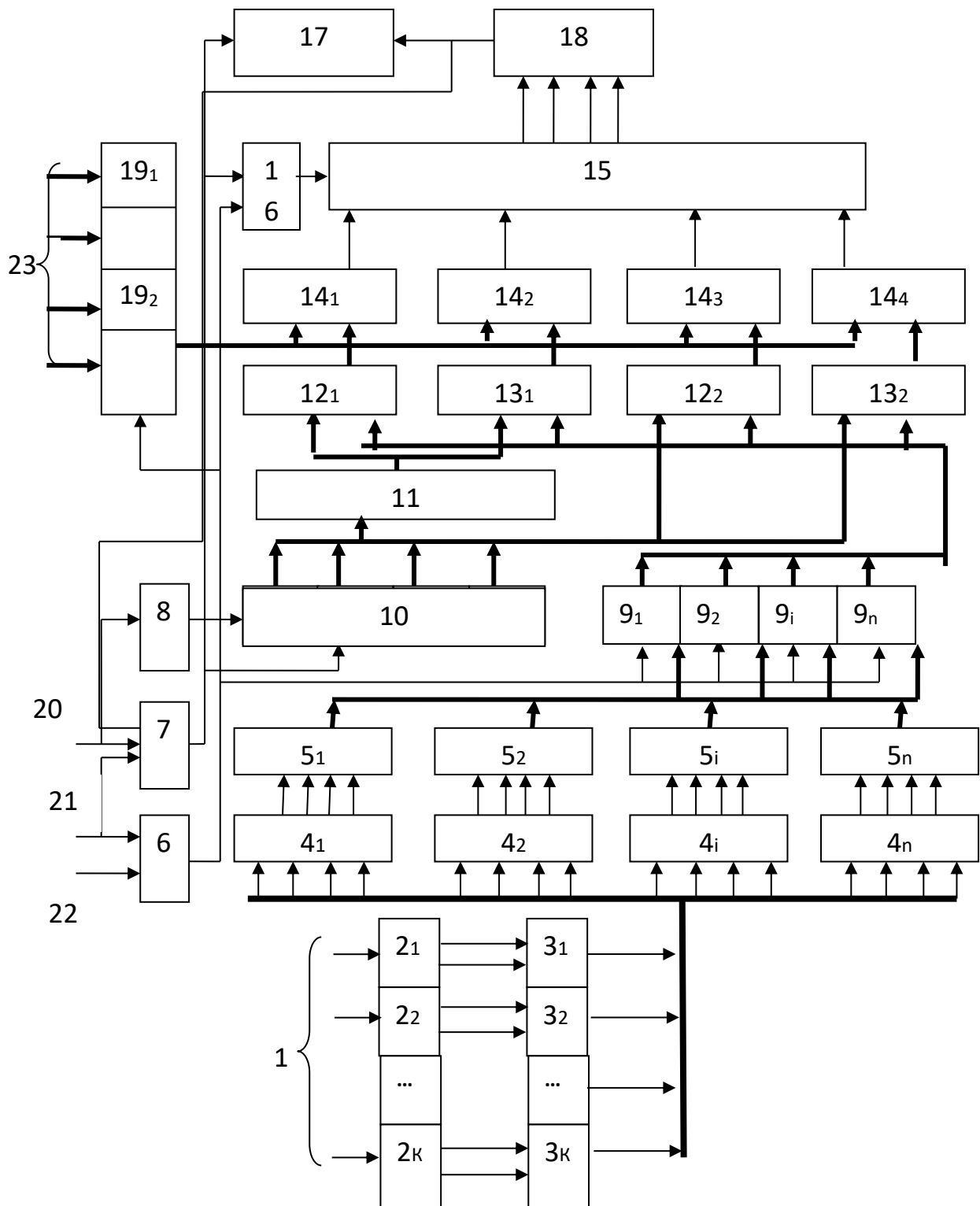


Рис.3. Структурна схема нелінійного сигнатурного аналізатора

Висновки

В проведеному дослідженні на основі формалізації процесу обробки інформації запропонована архітектура сигнатурного аналізатора. Така архітектура може використовуватись при перевірці об'єктів розподілених систем, діагностуванні складних пристроїв, а також контролю повідомлень в мережі передачі даних.

На основі моделювання процесу обробки інформації наведено методикку синтезу багатоканальних пристроїв з локалізацією помилок у вхідних інформаційних послідовностях. Такий підхід дозволяє створювати програмно-апаратні пристрої, які спроможні контролювати дані по групах, локалізувати багатократні помилки у вхідних інформаційних пакетах повідомлень. Розроблений узагальнений алгоритм обробки даних в багатоканальній сигнатурному аналізаторі

Запропоновані функціональні схеми пристроїв, що забезпечують діагностування елементів розподіленої інформаційної системи з точністю до типового елемента заміни, а також можливість визначення зіпсованих пакетів в повідомленнях. Такі пристрої пристрої контролю даних найбільш ефективні при перевірці передачі безлічі повідомлень пакетами по дейтаграммний каналу в режимі поділу часу.

Для подальшого розвитку цього напрямку роботи необхідно проведення досліджень що до впровадження вдосконалених алгоритмів пошуку помилок при використанні різноманітних протоколів обміну даними.

Література

1. Гордон Г., Надич Х. Локализация неисправностей в микропроцессорных системах при помощи шестнадцатирчных ключевых кодов // Электроника.– 1977.– №5.– С. 23 – 33.
2. Losev M. U. Analysis of ways for exchanging data in networks with package commutation / M. U. Losev, S. V. Minukhin, D. E Sitnikov // Radio Electronics Computer Science Control. – 2018. – №4. – С.196-204. <https://doi.org/10.15588/1607-3274-2018-4-19>
3. Патент на корисну модель 130723, МКИ3G06F11/00 Багатоканальний сигнатурний аналізатор з локалізацією помилок / М.Ю. Лосєв, О.В. Щербаков, Ю.М.Малишко, В.В. Федько (Україна). – № u201806022; заявл. 30.05.18; опубл.26.12.18, Бюл.№24. – 10 с.: іл.

ГЛАВА 4

РЕГУЛЯРИЗОВАНИЙ РЕКУРЕНТНИЙ МЕТОД ПОТОЧНОГО РЕГРЕСІЙНОГО АНАЛІЗУ ДЛЯ ПОБУДОВИ ЛІНІЙНИХ МОДЕЛЕЙ

Вступ

Багато задач управління, прогнозування, оцінювання тощо зводяться до побудови лінійної регресійної моделі виду

$$Y_n = X_n c^* + \Xi_n, \quad (1)$$

де $Y_n \in R^n$ – вектор вихідних сигналів, що спостерігається; $X_n \in R^{n \times N}$ – матриця нспостережень вхідних сигналів; $c^* \in R^N$ – вектор невідомих параметрів; $\Xi_n \in R^n$ – вектор завад; n – дискретний час, N – число вхідних змінних.

Визначення вектора невідомих параметрів здійснюється шляхом мінімізації деякого наперед обраного функціоналу якості (критерію ідентифікації). Найбільш широко застосовуваний при оцінюванні параметрів квадратичний функціонал призводить до оцінки МНК

$$c_n = (X_n^T X_n)^{-1} X_n^T Y_n, \quad (2)$$

яка при виконанні певних умов [1,2] є оцінкою максимальної правдоподібності і має відповідні екстремальні властивості. Володіючи рядом безперечних переваг, вона тим не менш не вільна від деяких недоліків, серед яких слід відзначити незручність її використання при вирішенні задач в реальному часі і недостатню обчислювальну стійкість. Перший недолік усувається порівняно просто шляхом переходу до рекурентного оцінювання, тобто шляхом використання рекурентного МНК [3]. При цьому, однак, необхідно відзначити, що важливою властивістю використовуваної в (2) (або обчислюється рекурентно в РМНК) матриці $(X_n^T X_n)^{-1}$ є її симетричність і позитивна визначеність. Недоліком же формули рекурентного обчислення цієї матриці є присутність в ній операції віднімання, яка при наявності численних помилок може призвести до

того, що дана матриця стане неперитивно визначеною. Для подолання цього недоліку були запропоновані спеціальні процедури, найбільш відомими з яких є метод Холеського або метод квадратного кореня. Цей метод дозволяє вдвічі зменшити помилку обчислень, однак його застосування істотно збільшує обсяг обчислень на кожному такті. У зв'язку з тим, що метод Холеського для подання зворотної матриці використовує деяку допоміжну матрицю, що є неєдиною, існують різні модифікації методу квадратного кореня, які враховують численні вимоги до швидкодії обладнання, пам'яті і програмного забезпечення. Щодо другого недоліку слід зазначити наступне.

Наявність мультиколінеарності призводить до того, що [4]:

1. Знижується точність оцінювання, що проявляється в трьох аспектах: помилки деяких оцінок стають дуже великими; ці помилки виявляються сильно корельованими одна з одною; вибіркові дисперсії різко зростають.

2. Можлива некоректність введення в аналіз деяких змінних, оскільки коефіцієнти при них виявляються незначними. Справжня причина може полягати не в тому, що ці змінні не впливають на вихідний сигнал, а в тому, що множина вибірових даних не дозволяє це вплив відобразити.

3. Оцінки параметрів дуже чутливі до особливостей множини вибірових даних (додавання незначної кількості спостережень може привести до істотної зміни оцінок параметрів).

Боротьба з мультиколінеарністю полягає в пошуку, якщо це можливо, нових даних чи іншої інформації. Однак дуже часто доводиться отримувати оцінки за допомогою лише доступною і обмеженою статистичної бази. У цьому випадку в якості основного методу боротьби з мультиколінеарністю пропонувалося виключення однієї з мультиколінеарних змінних або перетворення даних, що приводить до усунення або зменшення мультиколінеарності. Слід зазначити, що загального підходу до вирішення цього завдання немає, тому усунення другого недоліку здійснюється значно складніше. Підвищення стійкості може бути досягнуто, наприклад, використанням попередньої фільтрації вхідних сигналів,

модифікацією алгоритмів оцінювання шляхом введення в них деяких позитивних властивостей, що грають регуляризуючу роль, відмовою від вимоги незсуненості оцінок і переходом до зміщених, які мають меншу дисперсію [5].

РМНК с матричною корекцією

Ряд таких регуляризоване оцінок був запропонований в [6], зокрема, в цій роботі введена оцінка (оцінка Єрмакова)

$$c_n = \left(X_n^T X_n + k \left(X_n^T X_n \right)^{-1} \right)^{-1} X_n^T Y_n, \quad (3)$$

що представляє собою рідж-оцінку з матричною корекцією. Тут k - деякий позитивний параметр.

Оцінці (3) притаманний той же недолік, що й оцінки (2) - незручність використання в реальному часі. У зв'язку з цим в роботі [7] була розроблена і проаналізована рекурентная форма оцінки Єрмакова. Для отримання алгоритму застосовувався стандартний похід, який використовує блочне подання векторів і матриць, які входять у вираз (3) [8-10]

$$Y_n = \begin{pmatrix} Y_{n-1} \\ \text{---} \\ y_n \end{pmatrix} = \begin{pmatrix} y_1 \\ \text{---} \\ Y_n \end{pmatrix} - \text{вектор } n \times 1;$$

$$X_n = \begin{pmatrix} X_{n-1} \\ \text{---} \\ x_n^T \end{pmatrix} = \begin{pmatrix} x_{n-L+1}^T \\ \text{---} \\ X_n \end{pmatrix} - \text{матриця } n \times N;$$

При цьому рекурентні співвідношення, що описують регуляризовану оцінку, мають вигляд

$$c_n = \left(I + k \frac{S_n P_{n-1} x_n x_n^T P_{n-1}}{1 + x_n^T P_{n-1} x_n - k x_n^T P_{n-1} S_n P_{n-1} x_n} \right) \tilde{c}_n; \quad (4)$$

$$\tilde{c}_n = c_{n-1} + \frac{R_{n-1}x_n}{1 + x_n^T R_{n-1}x_n} (y_n - c_{n-1}^T x_n); \quad (5)$$

$$R_n = S_n + k \frac{S_n P_{n-1} x_n x_n^T P_{n-1} S_n}{1 + x_n^T P_{n-1} x_n - k x_n^T P_{n-1} S_n P_{n-1} x_n}; \quad (6)$$

$$S_n = R_{n-1} - \frac{R_{n-1} x_n x_n^T R_{n-1}}{1 + x_n^T R_{n-1} x_n}; \quad (7)$$

$$P_n = P_{n-1} - \frac{P_{n-1} x_n x_n^T P_{n-1}}{1 + x_n^T P_{n-1} x_n}; \quad (8)$$

$$R_0 = \alpha I; \quad 0 < \alpha_{\min} \leq \alpha \leq \alpha_{\max}, \infty;$$

$$P_0 = \beta I; \quad 0 < \beta_{\min} \leq \beta \leq \beta_{\max}, \infty,$$

де I - одинична матриця $N \times N$ (тут і далі розмірність цієї матриці не вказується і може бути визначена з контексту).

У цій роботі досліджувалися питання збіжності рекурентного алгоритму (4)-(8) і показано, що цей алгоритм збігається в середньому і середньоквадратичному, забезпечуючи отримання оцінок з меншою в порівнянні з МНК-оцінками дисперсією. Слід зазначити, що завдання ідентифікації істотно ускладнюється, якщо досліджуваній об'єкт нестационарній. У цьому випадку оцінювання нестационарних параметрів зазвичай здійснюється за допомогою модифікованих алгоритмів МНК, що мають механізм експоненційного зважування. Іншим підходом, який застосовується для оцінювання нестационарних параметрів, є використання в МНК ковзного вікна, тобто

обмеженого числа вимірів, що приводить до алгоритму метода поточного регресійного аналізу [11].

Побудова рекурентної форми алгоритму поточного регресійного аналізу

Алгоритм поточного регресійного аналізу, що має вигляд

$$c_{n|L} = (X_{n|L}^T X_{n|L})^{-1} X_{n|L}^T Y_{n|L}, \quad (9)$$

де

$$Y_{n|L} = \begin{pmatrix} Y_{n-1|L-1} \\ \text{---} \\ y_n \end{pmatrix} = \begin{pmatrix} y_{n-L+1} \\ \text{---} \\ Y_{n|L-1} \end{pmatrix} - \text{вектор } L \times 1;$$

$$X_{n|L} = \begin{pmatrix} X_{n-1|L-1} \\ \text{---} \\ x_n^T \end{pmatrix} = \begin{pmatrix} x_{n-L+1}^T \\ \text{---} \\ X_{n|L-1} \end{pmatrix} - \text{матриця } (L \times 1) \times N;$$

був запропонований в роботі [10], а в [11] розглянута модифікація цього алгоритму, що використовує механізм забування минулої інформації (зв'язування). Тут $L = \text{const} (L \geq N)$ - пам'ять алгоритму.

Особливістю алгоритмів з $L = \text{const}$ є те, що використовувані при побудові оцінок матриці і вектори спостережень на кожному кроці оцінювання формуються наступним чином: в них включається інформація про новоприбулі виміри і виключається інформація про найбільш старі. Залежно від того, як формуються ці матриці і вектори (додається спочатку нова інформація, а потім виключається застаріла, або ж спочатку виключається застаріла, а потім додається нова) можливі дві форми оцінки.

Зупинимося на цьому докладніше.

За аналогією з оцінкою (3) може бути запропонована оцінка

$$c_{n|L} = \left(X_{n|L}^T X_{n|L} + k \left(X_{n|L}^T X_{n|L} \right)^{-1} \right)^{-1} X_{n|L}^T Y_{n|L}, \quad (10)$$

де

$$Y_{n|L} = \begin{pmatrix} Y_{n-1|L-1} \\ \text{---} \\ y_n \end{pmatrix} = \begin{pmatrix} y_{n-L+1} \\ \text{---} \\ Y_{n|L-1} \end{pmatrix} - \text{вектор } (L+1) \times 1;$$

$$X_{n|L} = \begin{pmatrix} X_{n-1|L-1} \\ \text{---} \\ x_n^T \end{pmatrix} = \begin{pmatrix} x_{n-L+1}^T \\ \text{---} \\ X_{n|L-1} \end{pmatrix} - \text{матриця } (L+1) \times N;$$

Розглянемо випадок накопичення інформації.

Нехай на N -му такті отримана оцінка на основі $L+1$ спостереження.

$$c_{n+1|L+1} = \left(X_{n+1|L+1}^T X_{n+1|L+1} + k \left(X_{n+1|L+1}^T X_{n+1|L+1} \right)^{-1} \right)^{-1} X_{n+1|L+1}^T Y_{n+1|L+1}. \quad (11)$$

Позначимо

$$P_{n|L}^{-1} = X_{n|L}^T X_{n|L};$$

$$R_{n|L}^{-1} = X_{n|L}^T X_{n|L} + k \left(X_{n|L}^T X_{n|L} \right)^{-1}.$$

Тоді

$$P_{n+1|L+1}^{-1} = X_{n+1|L+1}^T X_{n+1|L+1} = X_{n|L}^T X_{n|L} + x_{n+1} x_{n+1}^T = P_{n|L}^{-1} + x_{n+1} x_{n+1}^T \quad (12)$$

$$\begin{aligned}
R_{n+1|L+1}^{-1} &= X_{n|L}^T X_{n|L} + x_{n+1} x_{n+1}^T + k \left(X_{n|L}^T X_{n|L} + x_{n+1} x_{n+1}^T \right)^{-1} = \\
&= X_{n|L}^T X_{n|L} + x_{n+1} x_{n+1}^T + k \left(P_{n|L} - \frac{P_{n|L} x_{n+1} x_{n+1}^T P_{n|L}}{1 + x_{n+1}^T P_{n-1|L-1} x_{n+1}} \right) = \\
&= R_{n|L}^{-1} + x_{n+1} x_{n+1}^T - k \frac{P_{n|L} x_{n+1} x_{n+1}^T P_{n|L}}{1 + x_{n+1}^T P_{n|L} x_{n+1}}.
\end{aligned}$$

Введемо позначення

$$S_{n+1|L+1}^{-1} = R_{n|L}^{-1} + x_{n+1} x_{n+1}^T. \quad (13)$$

З урахуванням цих позначень маємо

$$R_{n+1|L+1}^{-1} = S_{n+1|L+1}^{-1} - k \frac{P_{n|L} x_{n+1} x_{n+1}^T P_{n|L}}{1 + x_{n+1}^T P_{n|L} x_{n+1}}. \quad (14)$$

Застосовуючи до (12) - (14) лему про обертання матриць, отримуємо

$$P_{n+1|L+1} = P_{n|L} - \frac{P_{n|L} x_{n+1} x_{n+1}^T P_{n|L}}{1 + x_{n+1}^T P_{n-1|L-1} x_{n+1}}$$

$$S_{n+1|L+1} = R_{n|L} - \frac{R_{n|L} x_{n+1} x_{n+1}^T R_{n|L}}{1 + x_{n+1}^T R_{n|L} x_{n+1}};$$

$$R_{n+1|L+1} = S_{n+1|L+1} + k \frac{S_{n+1|L+1} P_{n|L} x_{n+1} x_{n+1}^T P_{n|L} S_{n+1|L+1}}{1 + x_{n+1}^T P_{n|L} x_{n+1} + k x_{n+1}^T P_{n|L} S_{n+1|L+1} P_{n|L} x_{n+1}}.$$

Таким чином, оцінка (11) має вид

$$\begin{aligned}
c_{n+1|L+1} &= R_{n+1|L+1} X_{n+1|L+1}^T Y_{n+1|L+1} = \\
&= \left(S_{n+1|L+1} + k \frac{S_{n+1|L+1} P_{n|L} x_n x_n^T P_{n|L} S_{n+1|L+1}}{1 + x_n^T P_{n|L} x_n + k x_n^T P_{n|L} S_{n+1|L+1} P_{n|L} x_n} \right) (X_{n|L}^T Y_{n|L} + x_{n+1} y_{n+1}).
\end{aligned}$$

Розглянемо добуток

$$\begin{aligned}
&S_{n+1|L+1} (X_{n|L}^T Y_{n|L} + x_{n+1} y_{n+1}) = \\
&= \left(R_{n|L} - \frac{R_{n|L} x_{n+1} x_{n+1}^T R_{n|L}}{1 + x_{n+1}^T R_{n|L} x_{n+1}} \right) (X_{n|L}^T Y_{n|L} + x_{n+1} y_{n+1}) = \\
&= R_{n|L} X_{n|L}^T Y_{n|L} + R_{n|L} x_{n+1} y_{n+1} - \frac{R_{n|L} x_{n+1} x_{n+1}^T R_{n|L} X_{n|L}^T Y_{n|L}}{1 + x_n^T R_{n-1|L-1} x_n} - \\
&\quad - \frac{R_{n|L} x_{n+1} x_{n+1}^T R_{n|L}}{1 + x_{n+1}^T R_{n|L} x_{n+1}} x_{n+1} y_{n+1} = c_{n|L} + \frac{R_{n|L} x_{n+1} x_{n+1}^T R_{n|L}}{1 + x_{n+1}^T R_{n|L} x_{n+1}} (y_{n+1} - c_{n|L}^T x_{n+1}).
\end{aligned}$$

Позначимо

$$\tilde{c}_{n+1|L+1} = c_{n|L} + \frac{R_{n|L} x_{n+1} x_{n+1}^T R_{n|L}}{1 + x_{n+1}^T R_{n|L} x_{n+1}} (y_{n+1} - c_{n|L}^T x_{n+1}).$$

Підставляючи отриманий вираз в (11), маємо

$$\begin{aligned}
c_{n+1|L+1} &= R_{n+1|L+1} X_{n+1|L+1}^T Y_{n+1|L+1} = c_{n|L} + \frac{R_{n|L} x_{n+1} x_{n+1}^T R_{n|L}}{1 + x_n^T R_{n-1|L-1} x_n} (y_{n+1} - c_{n|L}^T x_{n+1}) + \\
&+ k \frac{S_{n+1|L+1} P_{n|L} x_{n+1} x_{n+1}^T P_{n|L} S_{n+1|L+1}}{1 + x_{n+1}^T P_{n-1|L-1} x_{n+1} - k x_{n+1}^T P_{n|L} S_{n+1|L+1} P_{n|L} x_{n+1}} (X_{n|L}^T Y_{n|L} + x_{n+1} y_{n+1}) = \\
&= \left(I + k \frac{S_{n+1|L+1} P_{n|L} x_{n+1} x_{n+1}^T P_{n|L}}{1 + x_{n+1}^T P_{n|L} x_{n+1} - k x_{n+1}^T P_{n|L} S_{n+1|L+1} P_{n|L} x_{n+1}} \right) \tilde{c}_{n+1|L+1}.
\end{aligned}$$

Таким чином, отримані співвідношення описують процес накопичення інформації.

Розглянемо процедуру скидання застарілої інформації. На $n+1$ -у такті оцінка, отримана з використанням інформації про L попередні виміри, має вигляд

$$\begin{aligned} c_{n+1|L} &= (X_{n+1|L}^T X_{n+1|L} + k (X_{n+1|L}^T X_{n+1|L})^{-1})^{-1} X_{n+1|L}^T Y_{n+1|L} = \\ &= R_{n+1|L} X_{n+1|L}^T Y_{n+1|L} \end{aligned}$$

где

$$R_{n+1|L} = \left(X_{n+1|L}^T X_{n+1|L} + k (X_{n+1|L}^T X_{n+1|L})^{-1} \right)^{-1}.$$

Розглянемо зворотну матрицю

$$\begin{aligned} R_{n+1|L} &= (X_{n+1|L}^T X_{n+1|L} + k (X_{n+1|L}^T X_{n+1|L})^{-1})^{-1} = \\ &= \left(X_{n+1|L+1}^T X_{n+1|L+1} - x_{n-L+1} x_{n-L+1}^T + k (X_{n+1|L+1}^T X_{n+1|L+1} - x_{n-L+1} x_{n-L+1}^T)^{-1} \right) = \\ &= R_{n+1|L+1}^{-1} - x_{n-L+1} x_{n-L+1}^T + k \frac{P_{n+1|L+1} x_{n-L+1} x_{n-L+1}^T P_{n+1|L+1}}{1 - x_{n-L+1}^T P_{n+1|L+1} x_{n-L+1}}. \end{aligned}$$

Позначимо

$$S_{n+1|L}^{-1} = R_{n+1|L+1}^{-1} - x_{n-L+1} x_{n-L+1}^T;$$

$$P_{n+1|L}^{-1} = X_{n+1|L+1}^T X_{n+1|L+1} - x_{n-L+1} x_{n-L+1}^T = P_{n+1|L+1}^{-1} - x_{n-L+1} x_{n-L+1}^T$$

Тоді застосовуючи до отриманих співвідношенням лему про обертання матриць, маємо

$$P_{n+1|L} = P_{n+1|L+1} + \frac{P_{n+1|L+1} x_{n-L+1} x_{n-L+1}^T P_{n+1|L+1}}{1 - x_{n-L+1}^T P_{n+1|L+1} x_{n-L+1}} \quad (15)$$

$$R_{n+1|L}^{-1} = S_{n+1|L}^{-1} + k \frac{P_{n+1|L+1} x_{n-L+1} x_{n-L+1}^T P_{n+1|L+1}}{1 - x_{n-L+1}^T P_{n+1|L+1} x_{n-L+1}}. \quad (16)$$

Застосування до (15) і (16) леми про обертання матриць дає

$$S_{n+1|L} = R_{n+1|L+1} + \frac{R_{n+1|L+1} x_{n-L+1} x_{n-L+1}^T R_{n+1|L+1}}{1 - x_{n-L+1}^T R_{n+1|L+1} x_{n-L+1}};$$

$$R_{n+1|L} = S_{n+1|L} - k \frac{S_{n+1|L} P_{n+1|L+1} x_{n-L+1} x_{n-L+1}^T P_{n+1|L+1} S_{n+1|L}}{1 - x_{n-L+1}^T P_{n+1|L} x_{n-L+1} + k x_{n-L+1}^T P_{n+1|L} S_{n+1|L} P_{n+1|L} x_{n-L+1}}.$$

Для оцінки $c_{n+1|L}$ можна записати

$$\begin{aligned} c_{n+1|L} &= (X_{n+1|L}^T X_{n+1|L} + k (X_{n+1|L}^T X_{n+1|L})^{-1})^{-1} X_{n+1|L}^T Y_{n+1|L} = \\ &= R_{n+1|L} X_{n+1|L}^T Y_{n+1|L} \end{aligned}$$

або

$$\begin{aligned} c_{n+1|L} &= R_{n+1|L} X_{n+1|L}^T Y_{n+1|L} = \\ &= \left(S_{n+1|L+1} - k \frac{S_{n+1|L+1} P_{n+1|L+1} x_{n-L+1} x_{n-L+1}^T P_{n+1|L+1} S_{n+1|L+1}}{1 - x_{n-L+1}^T P_{n+1|L+1} x_{n-L+1} + k x_{n-L+1}^T P_{n+1|L+1} S_{n+1|L} P_{n+1|L+1} x_{n-L+1}} \right) \times \\ &\times \left(X_{n+1|L+1}^T Y_{n+1|L+1} - x_{n-L+1} y_{n-L+1} \right) = \\ &= c_{n+1|L+1} - \frac{R_{n+1|L} x_{n-L+1}}{1 - x_{n-L+1}^T P_{n+1|L+1} x_{n-L+1}} \left(y_{n-L+1} - c_{n|L}^T x_{n-L+1} \right). \end{aligned}$$

В дану виразі використовується інформація про n -й і $n-L+1$ -й такти. Позначимо допоміжну оцінку наступним чином:

$$\tilde{c}_{n+1|L} = c_{n+1|L+1} - \frac{R_{n+1|L} x_{n-L+1}}{1 - x_{n-L+1}^T P_{n+1|L+1} x_{n-L+1}} \left(y_{n-L+1} - c_{n|L}^T x_{n-L+1} \right).$$

З урахуванням цього отримуємо

$$\begin{aligned} c_{n+1|L} &= R_{n+1|L} X_{n+1|L}^T Y_{n+1|L} = \\ &= \left(I - k \frac{S_{n+1|L+1} P_{n+1|L+1} x_{n-L+1} x_{n-L+1}^T P_{n+1|L+1}}{1 - x_{n-L+1}^T P_{n+1|L+1} x_{n-L+1} + k x_{n-L+1}^T P_{n+1|L+1} S_{n+1|L+1} P_{n|L} x_{n-L+1}} \right) \tilde{c}_{n+1|L}. \end{aligned}$$

Підводячи підсумки, можна сказати, що отримані рекурентні співвідношення, що описують процес накопичення інформації, мають вигляд

$$c_{n+1|L+1} = \left(I + k \frac{S_{n+1|L+1} P_{n|L} x_{n+1} x_{n+1}^T P_{n|L}}{1 + x_{n+1}^T P_{n|L} x_{n+1} - k x_{n+1}^T P_{n|L} S_{n+1|L+1} P_{n|L} x_{n+1}} \right) \tilde{c}_{n+1|L+1}; \quad (17)$$

$$\tilde{c}_{n+1|L+1} = c_{n|L} + \frac{R_{n|L} x_{n+1}}{1 + x_{n+1}^T R_{n|L} x_{n+1}} \left(y_{n+1} - c_{n|L}^T x_{n+1} \right); \quad (18)$$

$$R_{n+1|L+1} = S_{n+1|L+1} + k \frac{S_{n+1|L+1} P_{n|L} x_{n+1} x_{n+1}^T P_{n|L} S_{n+1|L+1}}{1 + x_{n+1}^T P_{n|L} x_{n+1} - k x_{n+1}^T P_{n|L} S_{n+1|L+1} P_{n|L} x_{n+1}}; \quad (19)$$

$$S_{n+1|L+1} = R_{n|L} - \frac{R_{n|L} x_{n+1} x_{n+1}^T R_{n|L}}{1 + x_{n+1}^T R_{n|L} x_{n+1}}; \quad (20)$$

$$P_{n+1|L+1} = P_{n|L} - \frac{P_{n|L} x_{n+1} x_{n+1}^T P_{n|L}}{1 + x_{n+1}^T P_{n-1|L-1} x_{n+1}}. \quad (21)$$

Процес скидання застарілої інформації описується співвідношеннями

$$\begin{aligned}
c_{n+1|L} &= R_{n+1|L} X_{n+1|L}^T Y_{n+1|L} = \\
&= \left(I - k \frac{S_{n+1|L+1} P_{n+1|L+1} x_{n-L+1} x_{n-L+1}^T P_{n+1|L+1}}{1 - x_{n-L+1}^T P_{n+1|L+1} x_{n-L+1} + k x_{n-L+1}^T P_{n+1|L+1} S_{n+1|L+1} P_{n+1|L+1} x_{n-L+1}} \right) \tilde{c}_{n+1|L}; \tag{22}
\end{aligned}$$

$$\tilde{c}_{n+1|L} = c_{n+1|L+1} - \frac{R_{n+1|L} x_{n-L+1}}{1 - x_{n-L+1}^T P_{n+1|L+1} x_{n-L+1}} \left(y_{n-L+1} - c_{n+1|L+1}^T x_{n-L+1} \right); \tag{23}$$

$$R_{n+1|L} = S_{n+1|L} - k \frac{S_{n+1|L} P_{n+1|L+1} x_{n-L+1} x_{n-L+1}^T P_{n+1|L+1} S_{n+1|L}}{1 - x_{n-L+1}^T P_{n+1|L+1} x_{n-L+1} + k x_{n-L+1}^T P_{n+1|L+1} S_{n+1|L} P_{n+1|L+1} x_{n-L+1}}; \tag{24}$$

$$S_{n+1|L} = R_{n+1|L+1} - \frac{R_{n+1|L+1} x_{n-L+1} x_{n-L+1}^T R_{n+1|L+1}}{1 - x_{n-L+1}^T R_{n+1|L+1} x_{n-L+1}}; \tag{25}$$

$$P_{n+1|L} = P_{n+1|L+1} + \frac{P_{n+1|L+1} x_{n-L+1} x_{n-L+1}^T P_{n+1|L+1}}{1 - x_{n-L+1}^T P_{n+1|L+1} x_{n-L+1}}. \tag{26}$$

Вибір початкових значень для матриць, що входять в дані співвідношення, здійснюється як в методі найменших квадратів.

Моделювання.

Розглядалася задача ідентифікації лінійного об'єкта, що був представлений рівнянням (1) $N = 1$ з п'ятьма параметрами. Вивчалася задача оцінювання стаціонарних і нестаціонарних параметрів. У першому випадку задавалися наступні значення параметрів:

$c_1^* = 0.1, c_2^* = 0.2, c_3^* = 0.3, c_4^* = 0.4, c_5^* = 0.5$. В якості вхідних змінних обиралися нормально розподілені випадкові величини з $M\{x_{i,j}\} = 0$ та $M\{x_{i,j}^2\} = 1$. Залежність регресорів задавалася за допомогою коефіцієнта кореляції ρ . Проводилось усереднення по 10 реалізаціям.

В таблицях наведено значення величин F математичного сподівання $F = M \left\{ \frac{\sum_{i=1}^5 (c_{i,n}^* - c_{i,0}^*)^2}{\sum_{i=1}^5 (c_{i,0}^* - c_{i,0}^*)^2} \right\}$ після $n = 100$ тактів для різних значень коефіцієнта кореляції ρ та параметра регуляризації k . Таблиця 1 містить результати дослідження при виборі $L = 10$.

Таблиця 1

Значення критерія точності для різних значень коефіцієнта кореляції та параметра регуляризації при оцінюванні стаціонарних параметрів

Параметр регуляризації, k	Значення величин F		
	$\rho = 0,999$	$\rho = 0,995$	$\rho = 0,99$
10^0	0,94107	0,87428	0,77184
10^{-1}	0,06222	0,05079	0,05044
10^{-2}	0,05040	0,05012	0,05009
10^{-3}	0,04408	0,04205	0,04104
10^{-4}	0,04306	0,04304	0,04301
10^{-5}	0,04206	0,04205	0,04201
10^{-6}	0,04140	0,04107	0,04105
10^{-7}	0,04007	0,04004	0,04001
10^{-8}	0,04083	0,04051	0,04032
10^{-9}	0,04508	0,04621	0,04791

При дослідженні точності ідентифікації нестаціонарних параметрів приймалося, що три з п'яти параметрів змінюються у часі, причому два з них за

синусоїдальним законом, а третій - випадковим чином з заданими характеристиками

$$M \{c_3^*\} = 0.3, \quad M \{c_3^{*2}\} = 0.05.$$

В таблиці 2 наведено результати моделювання при виборі різних значень коефіцієнта кореляції, параметра регуляризації та пам'яті алгоритму L . Значення, які знаходяться над ризикою, відповідають випадку $L = 10$, а під ризикою - $L = 5$.

Таблиця 2

Значення критерія точності для різних значень коефіцієнта кореляції та параметра регуляризації при оцінюванні нестационарних параметрів

Параметр регуляризації, k	Значення величин F		
	$\rho = 0,999$	$\rho = 0,995$	$\rho = 0,99$
10^0	0,99208/0,97507	0,99102/0,96304	0,98405/0,93812
10^{-1}	0,98405/0,87108	0,98107/0,86105	0,97307/0,94904
10^{-2}	0,98301/0,76708	0,97408/0,76031	0,96750/0,68144
10^{-3}	0,98207/0,55107	0,97506/0,55093	0,96102/0,49392
10^{-4}	0,98109/0,4605	0,97405/0,43104	0,95703/0,32106
10^{-5}	0,98402/0,13404	0,97102/0,13007	0,94147/0,12931
10^{-6}	0,97506/0,09507	0,96207/0,08310	0,93561/0,076244
10^{-7}	0,98007/0,09905	0,96702/0,09316	0,95732/0,08936
10^{-8}	0,98019/0,10910	0,96809/0,09708	0,95810/0,09256
10^{-9}	0,98203/0,109316	0,96405/0,10315	0,95004/0,96103

Як видно з результатів моделювання, для кожного випадку (ідентифікація стаціонарних чи нестационарних параметрів) існує деяке оптимальне значення параметра регуляризації, вибір якого забезпечує мінімальне значення

функціонала. Крім того, при оцінюванні нестационарних параметрів можна обрати оптимальне значення пам'яті алгоритма яке, як випливає з результатів, повинно бути близьким до числа невідомих параметрів. Слід зауважити, що для оцінювання нестационарних параметрів моделі (1) з невідомим характером дрейфу більш зручними є багатокрокові проєкційні алгоритми, які мають близьку до розглянутих алгоритмів структуру та пам'ять $L < N$ [12-15]

Висновки

Таким чином, розглянуто задачу побудови рекурентної форми алгоритму поточного регресійного аналізу, що дозволяє здійснювати оцінювання невідомих параметрів за наявністю завад.

Отримано основні співвідношення, які описують процеси накопичення нової і скидання застарілої інформації. Особливістю алгоритмів, що розглядаються, є те, що в них використовується обмежене число спостережень, тобто обмежена пам'ять. При побудові формул для обчислення оцінок матриці і вектори спостережень на кожному кроці оцінювання формуються наступним чином: в них включається інформація про новоприбулі виміри і виключається інформація про найбільш старі. Залежно від того, як формуються ці матриці і вектори (додається спочатку нова інформація, а потім виключається застаріла, або ж спочатку виключається застаріла, а потім додається нова) можливі дві форми оцінки.

Показано, що алгоритми, які розглядаються, використовують деяку допоміжну оцінку. Результати імітаційного моделювання свідчать про ефективність підходу, що розвивається. Введення параметра регуляризації дозволяє отримати оцінки, які за наявності сильної кореляції вхідних змінних мають меншу середньоквадратичну похибку в порівнянні з методом найменших квадратів. Питання ж вибору оптимального значення параметра регуляризації залишається відкритим і потребує подальшого дослідження.

Література

1. Рао С.Р. Линейные статистические методы и их применения.- М.: Наука, 1968.-548 с.
2. Себер Дж. Линейный регрессионный анализ.- М.: Мир, 1980.- 456 с.
3. Алберт А. Регрессия, псевдоинверсия и рекуррентное оценивание.- М.:Наука, 1977.-305 с.
4. Джонсон Дж. Эконометрические методы. – М.: Статистика, 1980.- 444 с.
5. Демиденко Е.З. Линейная и нелинейная регрессии. - М.: Финансы и статистика, 1981.-302 с.
6. Ермаков С.М., Панкратьев Ю.Д. Смещенные оценки и метод регуляризации. / Вестник ЛГУ.-1976.- №7, вып.2.- С.27-30
7. Руденко О.Г. Рекуррентный регуляризованный алгоритм идентификации / Проблемы управления и информатики.-2001.-№2.- С. 33-41
8. Жуковский Е.Л., Липцер Р.Ш. О рекуррентном способе вычисления нормальных решений линейных алгебраических уравнений. / Журнал вычислительной математики и математической физики.-1972.-12,№4.- С.843-857
9. Руденко О.Г. Рекуррентное вычисление ридж-оценок / Докл. АН УССР.-1985.-№8.- С.72-74
10. Перельман И.И. Оперативная идентификация объектов управления.-М.: Энергоиздат,1982.-272 с.
11. Руденко О.Г., Теренковский И.Д., Штефан А., Ода Г.А. Модифицированный алгоритм текущего регрессионного анализа в задачах идентификации и прогнозирования. / Радиоэлектроника и информатика.- 1998.- № 4(05).- С.58-61.
12. Ищенко Л.А., Либероль Б.Д., Руденко О.Г. Проекционные алгоритмы идентификации линейных объектов // Докл. АН УССР. Сер. А. – 1985. – №7. – С. 62 - 64.
13. Ищенко Л.А., Либероль Б.Д., Руденко О.Г. Адаптивное оценивание параметров нестационарных объектов // Докл. АН УССР. Сер. А. – 1985. – №12. – С. 70 - 72.
14. Ищенко Л.А., Руденко О.Г. О свойствах одного класса многошаговых адаптивных алгоритмов идентификации // Кибернетика. – 1986. - №1. – С.92-96.
15. Либероль Б.Д., Руденко О.Г. О свойствах проекционных алгоритмов оценивания параметров нестационарных объектов // Докл. АН УССР. Сер. А. – 1990. – №4. – С. 71 - 74.

ГЛАВА 5

КЛАСИФІКАЦІЯ ОБ'ЄКТІВ НА ОСНОВІ МОДЕЛІ ДЕРЕВОВИДНОЇ ШТУЧНОЇ ІМУННОЇ МЕРЕЖІ

Вступ і постановка задачі

В даний час великою актуальністю характеризуються методи і моделі, що дозволяють проводити інтелектуальну обробку і систематизацію даних. До таких методів належать методи класифікації з контрольованим і неконтрольованим навчанням. Дані методи використовуються для вирішення завдання розбиття деякої множини об'єктів на класи або кластери по ряду ознак. Велике значення при класифікації об'єктів набуває організація процесу навчання. При контрольованому навчанні інформація про класи представляється явно, або за допомогою набору об'єктів навчальної вибірки, яка містить набір класифікованих об'єктів, що істотно спрощує завдання класифікації. У разі неконтрольованого навчання інформація про класи і їх кількості відсутня, а також відсутня навчальна вибірка, при цьому завдання класифікації ускладнюється і полягає у визначенні класів (кластерів) для множини об'єктів. Найбільш поширеними методами класифікації є метод з контрольованим навчанням kNN (k найближчих сусідів) і методи з неконтрольованим навчанням C-means (C-середніх), MST (мінімальне покриваюче дерево, яке формується на основі алгоритму Прима або алгоритму Краскала) і SVM (машина опорних векторів) та ін. [1, 2].

Для вирішення завдання класифікації активно використовуються моделі, які функціонують на основі біологічних принципів організації обчислень. Такими моделями є штучні нейронні мережі, генетичні алгоритми, штучні імунні системи (ШІС) та ін.. Однією з особливостей застосування ШІС є використання специфічних елементів – антигенів і антитіл, які взаємодіють між собою при вирішенні задачі класифікації. При класифікації об'єктів на основі принципів роботи ШІС множина об'єктів, що групуються, представляється набором антитіл. У разі використання контрольованого навчання навчальна вибірка представляється набором антигенів. У разі класифікації з неконтрольованим навчанням навчальної вибірки немає, тому формування кластерів відбувається в результаті аналізу зв'язків між антитілами.

Існуючі імунні моделі, що використовуються для вирішення завдань класифікації, використовують однакові підходи на етапах клонування, мутації і

відбору клонів [3], що призводить до великої кількості обчислювальних операцій. Тому імунні методи класифікації можуть використовуватися при роботі з об'єктами в багатовимірному просторі ознак і зазвичай поступаються класичним методам класифікації об'єктів, що характеризуються невеликою кількістю ознак, таким як kNN, MST або C-means.

У даній роботі пропонується новий підхід до організації роботи моделі імунної мережі при класифікації з контрольованим і неконтрольованим навчанням в n -вимірному просторі ознак. Дана модель в разі неконтрольованого навчання передбачає формування класифікатора як дерева антитіл на основі принципів, які використовуються в MST. У разі контрольованого навчання передбачається робота з класифікатором, який представляє собою дерево антитіл, яке застосовується до параметрів заданої групи об'єктів. У запропонованій імунній моделі в якості міри зв'язку між ознаками об'єктів використовується критерій афінності, а в якості міри зв'язку між об'єктами, що характеризуються великою кількістю ознак, – критерій авідності. Крім того, в запропонованій імунній моделі, в разі неконтрольованого навчання, кількість кластерів визначається в процесі взаємодії об'єктів і не є вхідним параметром, як в методах C-means та MST.

1. Аналіз імунних методів і моделей класифікації

Одним з найбільш простих методів класифікації з неконтрольованим навчанням є графовий метод MST [1, 2], який характеризується простотою реалізації і використовує математичний апарат теорії графів для угруповання об'єктів. Даний метод характеризується високою швидкодією, оскільки визначення відстаней між усіма вершинами графа об'єктів виконується тільки один раз. Наступні операції поділу графа пов'язані лише з сортуваннями, пошуком і видаленням ребер з найбільшою довжиною. Однак, при використанні методу MST кількість кластерів, що виділяються з графа, має бути відомо заздалегідь, тому що є обов'язковим вхідним параметром.

Використання методу C-means для класифікації з неконтрольованим навчанням передбачає багаторазове перевизначення заданої кількості центроїдів (центрів кластерів) [1, 2]. Слід зазначити що, незважаючи на те, що цьому методу потрібно більше часу на класифікацію, ніж методу MST, він володіє більшою точністю поділу об'єктів і частіше використовується при класифікації в

багатовимірному просторі ознак.

Серед імунних моделей, що використовуються при класифікації, можна виокремити моделі клонального відбору (BCA і CLONALG) і моделі штучної імунної мережі (aiNET і RLAIS) [3]. Незважаючи на те, що дані моделі і методи використовують різні принципи функціонування імунних систем, вони використовують багато спільних імунних операторів, що реалізують певні етапи роботи імунної системи. Робота більшості імунних операторів пов'язана з великою кількістю обчислювальних операцій, що призводить до зниження швидкості роботи імунного методу при вирішенні тієї чи іншої практичної задачі. Тому при класифікації об'єктів, що характеризуються невеликою кількістю ознак, імунні методи класифікації поступаються за швидкістю роботи класичним методам угруповання об'єктів. Рішення задачі класифікації даних з використанням модифікованої імунної моделі клонального відбору CLONALGm розглянуто у [4].

В моделях штучної імунної мережі aiNET і RLAIS після клонування і мутації досліджується мережева взаємодія всередині популяції антитіл, і після, або замість проведення відбору клонів відбувається стиснення мережі антитіл шляхом їх супресії. У [5] розглянуто рішення задачі класифікації даних з використанням модифікованої моделі штучної імунної мережі aiNETmc. Дані методи і моделі універсальні і можуть використовуватися для вирішення завдань класифікації з контрольованим і неконтрольованим навчанням, однак, вони більш складні в реалізації і модифікації.

2. Реалізація моделі деревовидної імунної мережі для класифікації

Пропонована модель імунної мережі може використовуватися для класифікації як з контрольованим, так і з неконтрольованим навчанням. У разі класифікації з контрольованим навчанням пропонована модель може використовувати заздалегідь сформований класифікатор, використання якого дозволить визначити належність досліджуваних об'єктів тому чи іншому класу, представленому множиною об'єктів класифікатора. У разі класифікації з неконтрольованим навчанням (самонавчанням) модель формує класифікатор, розділяючи об'єкти на групи (кластери), і уточнює їх кількість. Незалежно від способу навчання, вихідні об'єкти представляються множиною антитіл $Ab\{ab_1; \dots; ab_n\}$, об'єкти класифікатора (навчальна вибірка) представляються

набором антигенів $Ag\{ag_1; \dots; ag_n\}$. При цьому для опису множини об'єктів, що класифікуються, і об'єктів навчальної вибірки використовується однаковий формат представлення даних. Якщо ознаки об'єктів, що класифікуються, можуть бути розділені на групи, кожна група може мати свій ваговий коефіцієнт. У разі використання ваг для груп ознак, сума цих ваг дорівнює 1.

У більшості існуючих імунних методів класифікації в якості міри відповідності між об'єктами, представленими популяціями антитіл і антигенів, використовується значення афінності. Визначення значення афінності зазвичай ґрунтується на Евклідовій відстані між антитілами і антигенами:

$$aff(ab_i, ag_j) = \left(1 + \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \dots + (n_i - n_j)^2}\right)^{-1}, \quad (1)$$

де x_i, y_i, \dots, k_i – параметри i -го антитіла, x_j, y_j, \dots, k_j – параметри j -го антигена, $\forall i \in [1, \dots, n], \forall j \in [1, \dots, m]$.

Замість евклідової відстані у формулі афінності з метою підвищення продуктивності може використовуватися манхеттенська відстань, що може прискорити роботу ШС при вирішенні різних практичних завдань. Проте, незалежно від способу розрахунку афінності, більшість існуючих імунних моделей характеризуються низькою швидкістю. Це пов'язано з тим, що в процесі роботи більшість існуючих імунних моделей виробляють багаторазове клонування, мутацію і відбір клонів з метою підвищення афінності між об'єктами, що класифікуються, і цільовими класами або центрами кластерів. При цьому для кожної популяції антитіл в процесі відбору клонів або супресії відбувається визначення афінності до багатьох антигенів, або антитіл, що і є причиною низької швидкості навчання імунних моделей.

Робота моделі класифікації, запропонованої в даній роботі, здійснюється на основі ряду наступних принципів:

- повне відтворення ознак об'єкта за кінцеву кількість циклів клонування і мутації;
- єдиність використання ознаки при визначенні афінності;
- спрощення етапів клонування, мутації і стиснення мережі;
- використання деревовидної структури переважно для представлення мережі антитіл;
- використання властивостей біологічних антитіл добре застосувати для

рішення задачі класифікації при деревовидному поданні імунної мережі.

Під повним відтворенням ознак об'єкта за кінцеву кількість циклів клонування і мутації розуміється можливість створення такого антитіла, яке буде повністю специфічним вихідному об'єкту-антигену. Після цього для кожного клону визначаються афінності до популяції антигенів, або деякої обмеженої кількості антигенів або антитіл. Це робиться для забезпечення редагування популяції антитіл шляхом відбору клонів або супресії мережі антитіл. В результаті такого відбору зі всієї множини клонів відбирається обмежена кількість об'єктів з найкращими афінностями до антигенів. Наслідком використання такого підходу є велика кількість обчислювальних операцій, які знижують загальну швидкість роботи моделі.

У даній роботі пропонується інший принцип організації обчислень, який мінімізує кількість операцій обчислення афінності, що призводить до значного підвищення швидкості роботи імунної моделі класифікації. Оскільки за допомогою клонування і мутації ШС можуть сформувати ідентичний (по набору ознак) об'єкт для кожного антигену за фіксовану кількість циклів клонування і мутації, можна встановити цей стан як 100% відновлення ознаки. Відповідно до цього, якщо на кожному циклі мутації відбувається формування клону, ознаки якого на 1% ближче до ознак деякого цільового об'єкта, то для повного відтворення цієї ознаки потрібно 100 мутацій. Таким чином, у відповідності з твердженням про можливість повного відтворення ознак деякого об'єкта шляхом клонування і мутації, при 1% мутації кожної ознаки система за 100 мутацій з об'єкта з мінімальними значеннями ознак зможе сформувати об'єкт з максимальними значеннями ознак. Завдяки цьому, знаючи обмеження простору класифікації можна знайти значення зміни кожної ознаки на коефіцієнт мутації μ (наприклад, це 1%), яке і буде додаватися клонам при мутації. Це дозволяє уявити відстань між будь-якими двома об'єктами в системі як деяку кількість мутацій з коефіцієнтом мутації μ . Це дозволить абстрагуватися від фізичного створення клонів, а також об'єднати етапи клонування, мутації і стиснення мережі в одному операторі мутації.

Твердження про єдиність використання ознаки при визначенні афінності між імунними об'єктами має на увазі істотну зміну формули афінності. Значення афінності антитіла з антигеном в імунних системах відображає міцність зв'язку між антитілом і антигеном. В ШС значення афінності між об'єктами

визначається зазвичай на основі евклідової або манхеттенської відстані [3]. При цьому така відстань торкається тільки однієї групи ознак об'єктів. Наприклад, якщо об'єкти, що класифікуються, представлені різнокольоровими точками в двомірному просторі, то афінність не може оперувати одночасно і координатами, і кольором. В такому випадку визначаються дві різні афінності (і на основі координат, і на основі кольору), які усереднюються, або зважуються для визначення деякої загальної афінності між об'єктами. Якщо відстань між об'єктами уявити кількістю мутацій по кожній координаті і по кожній колірній ознаці, то значення афінності можна визначати на основі максимальної кількості мутацій однієї з ознак об'єкта без використання вагових коефіцієнтів.

У запропонованій імунній моделі класифікації афінність між антитілами і антигенами обчислюється в такий спосіб:

$$aff(ab_i, ag_j) = 1 - MaxMutations_k / 100, \quad (2)$$

де $MaxMutations_k$ – значення максимальної кількості мутацій для відновлення деякої k -ї ознаки.

Використання такого підходу спрощує формулу обчислення афінності між об'єктами, що призводить до підвищення швидкості класифікації.

Спрощення етапів клонування, мутації і стиснення мережі при вирішенні практичних завдань класифікації призводить до об'єднання даних етапів в один оператор мутації, що визначає серед усіх ознак антитіла одну ознаку, для якої необхідно провести максимальну кількість операцій 1% мутації (з коефіцієнтом мутації μ), щоб він відтворив відповідну ознаку деякого цільового об'єкта (антигену або антитіла). Однак слід зазначити, що при симуляції роботи ШС в дослідницьких цілях, етапи клонування, мутації і відбору клонів (або супресії мережі) повинні бути реалізовані в повній мірі.

При поданні множини антитіл або антигенів у вигляді графа або дерева всі збережені об'єкти використовуються в якості вершин, з'єднаних ребрами. При цьому ребрами даного графа є афінності між об'єктами, які визначаються згідно з (2). Слід відзначити, що не всі ребра такого графа (дерева антитіл) доцільно зберігати, оскільки його зберігання приведе до додаткових витрат оперативної пам'яті пристрою, на якому проводиться моделювання. В такому випадку критерій, який використовується для скорочення кількості ребер (зв'язків і афінностей) між вершинами (об'єктами популяції антитіл), грає дуже важливу

роль. В якості такого критерію у пропонованій моделі використовується значення мінімально допустимої афінності між об'єктами, достатнє для проведення класифікації. Дане значення визначається так:

$$aff_{\min} = 0.5 \left(1 + \frac{count}{range_{\max}} \right), \quad (3)$$

де $count$ – кількість класифікованих антитіл (кількість вершин графа), а $range_{\max}$ – значення максимальної ознаки з набору вершин графа.

Відповідно до цього при формуванні дерева антитіл, що використовується при класифікації, з усіх можливих вершин (афінностей між об'єктами) залишаються ті, які перевищують значення aff_{\min} . В іншому випадку вони видаляються з графа (придушуються при супресії мережі).

Більшість існуючих імунних моделей класифікації оперує обмеженим набором властивостей антитіл. У більшості випадків при вирішенні практичних завдань використовується тільки значення афінності між антитілом і антигеном, або афінностей між антитілами. Однак серед властивостей біологічних антитіл виділяють також авідність і валентність, які також можуть бути корисні при вирішенні багатьох практичних завдань на основі імунного підходу. Під валентністю біологічного антитіла розуміють кількість антидетермінант (клітинних центрів), здатних вступити в реакцію з антигеном [6]. Залежно від цієї характеристики антитіла поділяють на бівалентні (здатні мати два зв'язки з антигенами, або мати зв'язки з двома антигенами) і полівалентні (здатні мати більше двох зв'язків з антигенами). Валентність антитіл може використовуватися для подання зв'язків між вершинами графа, тобто якщо одна вершина графа імунних об'єктів пов'язана тільки з однією вершиною, то її валентність дорівнює 1. Якщо деяка вершина графа пов'язана афінностями з трьома вершинами графа імунних об'єктів, то вона буде мати валентність 3. Таким чином, валентність антитіла в графі імунних об'єктів прямо пропорційна кількості афінностей, що перевищують граничне значення aff_{\min} (3).

Авідність антитіла в біологічних імунних системах характеризує міцність зв'язку між антитілом і антигеном і визначається на основі афінності між ними і валентності антитіла [6]. Стосовно до задачі класифікації на основі імунного підходу, авідність може використовуватися для визначення належності деякого

об'єкта тому чи іншому класу при класифікації з контрольованим навчанням. При цьому значення авідності визначається відповідно до виразу:

$$av(ab_i, ag_j) = aff(ab_i, ag_j) \times v_i, \quad (4)$$

де $aff(ab_i, ag_j)$ – афінність між i -м антитілом і j -м антигеном, а v_i – валентність даного антитіла.

Таким чином, при класифікації з контрольованим навчанням інформація про класи представляється графом антитіл, а об'єкти, що класифікуються, – набором антигенів. В процесі роботи імунної моделі антитіла пов'язуються з антигенами і визначають до них значення авідності. Належність до того чи іншого класу визначається за максимальною авідністю з об'єктами навчальної вибірки, що класифікуються.

Слід зазначити, що описані принципи і твердження про роботу ШС при класифікації об'єктів пропонуються вперше. Вони суттєво спрощують складність моделі і скорочують кількість імунних операторів, виконуючих велику кількість складних обчислювальних операцій. Пропонована модель деревовидної штучної імунної мережі daiNet для класифікації об'єктів може функціонувати в різних режимах: класифікація з контрольованим навчанням, класифікація з неконтрольованим навчанням і класифікація з навчанням з підкріпленням. Режим роботи залежить не тільки від наявності класифікатора, що містить граф антитіл в якості вихідної інформації про класи і значення коефіцієнта мутації для області класифікації, що використовується, але і від значення допустимої афінності, яка використовується для скорочення кількості ребер графа (в разі навчання з підкріпленням). Основними особливостями пропонованої моделі класифікації daiNet є:

- поділ популяції об'єктів, що класифікуються, на дві множини: 1) об'єкти, які можуть бути класифіковані на основі існуючого класифікатора; 2) об'єкти, які не можуть бути класифіковані на основі класифікатора, для яких необхідно провести додатково неконтрольоване навчання;
- використання значення авідності при класифікації з контрольованим навчанням;
- спрощення формули афінності між імунними об'єктами;
- використання мінімально допустимої афінності для поділу об'єктів на множини;

– можливість проведення навчання з підкріпленням.

Відповідно до цього, процес класифікації об'єктів на основі моделі daiNet на формальному рівні може бути описаний в такий спосіб:

$$\begin{aligned}
 daiNet(AbTree, Obj, aff_{\min}, \mu) = & \left[\begin{array}{l} Presentation(AbTree, Obj) \rightarrow \\ \rightarrow Mutation(AbTree, Obj, \mu) \rightarrow \\ \rightarrow Selection(AbTree, Obj, aff_{\min}) \rightarrow \\ \rightarrow Classification(AbTree, Obj') \end{array} \right]^{SL} \rightarrow \\
 \rightarrow & \left[\begin{array}{l} NetPresentation(Obj'') \rightarrow \\ \rightarrow Mutation(Obj'', \mu') \rightarrow \\ \rightarrow Supression(Obj'', aff''_{\min}) \rightarrow \\ \rightarrow Classification(Obj'') \end{array} \right]^{USL} \rightarrow , \\
 \rightarrow & NetFormation(AbTree, Obj', Obj'')^{RL}
 \end{aligned} \tag{5}$$

де SL – етап класифікації з контрольованим навчанням; USL – етап класифікації з неконтрольованим навчанням; RL – етап класифікації з підкріпленням; $Presentation(AbTree, Obj)$ – оператор визначення різниці між ознаками антитіл класифікатора і об'єктами, що класифікуються; $Mutation(AbTree, Obj, \mu)$ – оператор визначення кількості мутацій, необхідних для відновлення ознак об'єктів антитілами класифікатора; $Selection(AbTree, Obj, aff_{\min})$ – оператор поділу набору об'єктів, що класифікуються, на об'єкти, які можуть бути класифіковані за допомогою чинного класифікатора, і об'єкти, для яких необхідно виділення нових класів; $Classification(AbTree, Obj')$ – оператор класифікації об'єктів за допомогою деревоподібної мережі антитіл, їх валентностей і авідностей до об'єктів; $NetPresentation(Obj'')$ – оператор подання визначення афінності між об'єктами, які не можуть бути класифіковані за допомогою мережі антитіл класифікатора; $Mutation(Obj'', \mu')$ – оператор визначення коефіцієнта мутації і кількості мутацій, які необхідні для відновлення об'єктів, для яких потрібно формування додаткових класів; $Supression(Obj'', aff''_{\min})$ – оператор визначення порогової афінності і видалення надлишкових зв'язків між об'єктами, які не можуть бути класифіковані за допомогою класифікатора; $Classification(Obj'')$ – оператор

виділення нових класів для об'єктів, які не можуть бути класифіковані при використанні класифікатора на ранніх етапах роботи імунної моделі; *NetFormation*(*AbTree, Obj', Obj''*) – оператор поповнення класифікатора новими і класифікованими об'єктами.

Етап класифікації з контрольованим навчанням *SL* є першим етапом роботи алгоритму *daiNet* при класифікації вихідних об'єктів. На даному етапі передбачається використання класифікатора, представленого деревовидною мережею антитіл з різними валентностями. На цьому етапі здійснюється поділ об'єктів, що класифікуються, на дві множини об'єктів: об'єкти, які можуть бути класифіковані за допомогою класифікатора, і об'єкти, для яких необхідно виділення нових класів. Для цього використовується значення мінімально допустимої афінності aff_{min} (3), яка визначає кількість мутацій антитіл, достатньої для класифікації об'єктів. У разі, якщо в деревовидній мережі антитіл класифікатора є хоча б одне антитіло, у якого афінність до об'єктів, що класифікуються, перевершує значення aff_{min} , то такий об'єкт може бути класифікований за допомогою класифікатора, в іншому випадку – об'єкт не може бути віднесений до жодного з вихідних класів, і для його угруповання необхідне створення додаткового класу в процесі класифікації з неконтрольованим навчанням.

На другому етапі роботи запропонованої імунної моделі класифікації *USL* відбувається групування об'єктів, які не можуть бути класифіковані за допомогою вихідної деревовидної імунної мережі класифікатора. На цьому етапі з даних об'єктів формується нова імунна мережа, для якої перевизначається значення мінімальної афінності aff_{min} , достатньої для класифікації. Після цього зі сформованої мережі виділяються дерева об'єктів, які характеризуються авідностями і належностями. Належність об'єкта до того чи іншого дерева визначає його належність до одного з виділених класів.

На завершальному етапі класифікації *RL* відбувається оновлення деревовидної імунної мережі класифікатора новими деревами об'єктів, виділеними в результаті класифікації з неконтрольованим навчанням. При цьому дерева ранжуються, тобто поділяються за черговістю застосування в разі подальшого використання. Відповідно до цього, при подальшому використанні для класифікації іншої множини об'єктів, в першу чергу, будуть використовуватися дерева антитіл вихідного класифікатора (дерева з

мінімальним рангом), які характеризуються початковим значенням порогової афінності, достатньої для класифікації. Потім, якщо об'єкти, що класифікуються, не можуть бути віднесені до жодного з класів даного рангу, вони визначають належність до дерев антитіл другого рангу, які характеризуються власним значенням порогової афінності для класифікації. Підвищення рангів класів для об'єктів, що класифікуються, відбувається до тих пір, поки об'єкт не буде віднесений до одного з класів класифікатора, або для нього не відбудеться виділення нового класу.

Слід зазначити, що в разі відсутності класифікатора, представленого деревовидною мережею антитіл, робота запропонованої імунної моделі класифікації daiNet починається з етапу класифікації на основі неконтрольованого навчання для виділення класів з множини об'єктів.

Оператор *Presentation* ($AbTree, Obj$) використовується для визначення різниці між ознаками антитіл деревовидної імунної мережі (антитілами класифікатора) і об'єктами досліджуваної множини. В результаті роботи даного оператора формується матриця відмінностей між кожним класифікованим об'єктом і кожним антитілом класифікатора.

Оператор *Mutation* ($AbTree, Obj, \mu$) застосовується для визначення кількості мутацій (на основі коефіцієнта мутації μ), необхідного для повного відновлення ознак об'єктів антитілами класифікатора. При цьому μ може визначатися, наприклад, як 1% від відстані між мінімальним і максимальним значеннями характеристик антитіл.

Оператор *Selection* ($AbTree, Obj, aff_{\min}$) застосовується для поділу множини об'єктів на дві групи: об'єкти, які можуть бути класифіковані за допомогою класифікатора, і об'єкти, для яких необхідне формування нових класів. При цьому спочатку для кожного антитіла визначаються афінності з об'єктами (на основі кількості мутацій, визначених раніше), потім при використанні мінімальної припустимої афінності aff_{\min} відбувається поділ досліджуваної множини об'єктів. В результаті цього поділу з початкової множини об'єктів Obj' виділяються об'єкти Obj'' , у яких є хоча б одна афінність з будь-яким антитілом класифікатора, яка перевищує значення мінімально допустимої афінності aff_{\min} . Такі антитіла можуть бути класифіковані за допомогою класифікатора на основі їх афінностей та авідностей з антитілами.

Для класифікації об'єктів Obj'' , що залишилися, необхідно виділення нових класів в результаті неконтрольованого навчання.

Оператор $Classification (AbTree, Obj')$ застосовується для класифікації об'єктів з множини Obj' . При цьому висновок про належність об'єктів до певного класу визначається на підставі авідності між антитілом цього класу і об'єктом, що класифікується.

Оператор $NetRepresentation (Obj'')$ застосовується для визначення відстаней між ознаками об'єктів, які не можуть бути класифіковані за допомогою класифікатора і його антитіл. На цьому етапі також визначаються значення коефіцієнта мутації μ' і мінімально допустима афінність aff''_{min} , яка використовується для формування додаткових класів і нового класифікатора.

Оператор $Mutation (Obj'', \mu')$ застосовується для визначення кількості мутацій, необхідної для відновлення ознак між об'єктами. При цьому μ' може визначатися, наприклад, як 1% від відстані між мінімальним і максимальним значеннями характеристик об'єктів, для яких формується новий клас.

Оператор $Supression (Obj'', aff''_{min})$ застосовується для виділення зв'язків між об'єктами з множини об'єктів Obj'' , сформованої на основі кількості мутацій, необхідної для взаємного відновлення їх ознак за допомогою мутації. Перед виділенням таких зв'язків між усіма об'єктами множини Obj'' визначаються афінності згідно з (2), а також значення порогової афінності aff''_{min} , необхідне для скорочення кількості зв'язків між об'єктами. Потім здійснюється видалення зв'язків між об'єктами, у яких афінність менше виділеного порогового значення.

Оператор $Classification (Obj'')$ застосовується для формування деревовидної мережі антитіл з об'єктів множини Obj'' , де вершинами є об'єкти, а ребрами – авідності між ними, визначені відповідно до виразу (4).

Оператор $NetFormation (AbTree, Obj', Obj'')$ застосовується для об'єднання результатів класифікації з контрольованим навчанням і результатів класифікації з неконтрольованим навчанням в одну ранжовану мережу. При цьому деревоподібна імунна мережа першого рівня формується з антитіл вихідного класифікатора і об'єктів, класифікованих за допомогою даного класифікатора, а мережа другого рівня формується на основі об'єктів, для яких потрібно виділення нових класів.

3. Експериментальні результати

Тестування запропонованого імунного методу класифікації daiNET виконувалось на декількох наборах вихідних об'єктів. При цьому набори об'єктів, що класифікуються, різнилися між собою як кількістю об'єктів, так і розмірністю ознак, що характеризують кожен об'єкт. Характеристики наборів об'єктів, що класифікуються, представлені в табл. 1.

Таблиця 1

Набори об'єктів, що класифікуються

Ідентифікатор	Кількість		Кількість ознак
	об'єктів	класів	
Набір 1	100	2	2×3
Набір 2	500	5	5×3
Набір 3	1000	20	8×8
Набір 4	10000	50	10×10

Кожен набір об'єктів використовується в якості контрольної вибірки для оцінки точності класифікації за допомогою різних алгоритмів. Основні відмінності між цими наборами – це кількість класів, розмірність матриці об'єктів та кількість об'єктів у наборі.

Для порівняння запропонованого методу з іншими імунними і не імунними методами, застосовуваними для вирішення задачі класифікації наборів вихідних об'єктів, наведених у табл. 1, використовувалися алгоритми MST Prima і C-means [1, 2]. Крім того, для класифікації також використовувалися найпоширеніші імунні методи ВСА, CLONALG, aiNET [3] та метод aiNETmc [5]. Результати класифікації наборів даних наведені в табл. 2, в якій використовується декілька умовних ознак: «S» характеризує загальний час, витрачений на класифікацію об'єктів; «C» характеризує кількість класів, виділених в результаті класифікації; «A» характеризує точність класифікації.

Результати класифікації

Алгоритм		Набор 1	Набор 2	Набор 3	Набор 4
MST Prima	S	31.2 %	31.3 %	31.8 %	31.7 %
	C	2	5	20	50
	A	98.7 %	98.3 %	97.5 %	96.8 %
C-means	S	54.3 %	54.8 %	54.8 %	55.2 %
	C	2	5	20	50
	A	99.8 %	99.7 %	99.3 %	99.1 %
CLONALG	S	67.3 %	67.6 %	67.8 %	67.9 %
	C	2	5	22	48
	A	70.6 %	70.5 %	70.4 %	70.5 %
BCA	S	100 %	100 %	100 %	100 %
	C	2	5	20	50
	A	99.8 %	99.6 %	99.2 %	98.9 %
aiNET	S	62.5 %	62.5 %	62.9 %	62.9 %
	C	2	5	24	54
	A	65.5 %	65.2 %	64.2 %	63.9 %
aiNETmc	S	49.1 %	49.2 %	49.2 %	49.3 %
	C	2	5	20	50
	A	99.2 %	99.2 %	98.7 %	98.1 %
daiNET	S	29.8 %	29.8 %	30.1 %	30.1 %
	C	2	5	20	50
	A	99.3 %	99.3 %	99.1 %	98.9 %

Відповідно до отриманих результатів класифікації можна зробити висновок, що імунний метод BCA характеризується найгіршою ефективністю. Але за точністю класифікації він не поступається методам MST Prima та C-means. Отже, час класифікації наборів даних за методом BCA приймається як максимальне значення (100%). Метод C-means є еталонним алгоритмом не тільки з точки зору часу роботи, але і з точки зору точності класифікації. Алгоритм MST Prima за швидкістю перевищує метод C-means в середньому на 20-25%, але поступається йому за точністю класифікації на 3-5%.

Слід зазначити, що більшість імунних методів BCA, CLONALG, aiNET характеризуються низькою швидкістю та відносно низькою точністю

класифікації (за винятком методу ВСА). Метод класифікації aiNETmc за точністю класифікації порівняний з методом C-means (на 1-1,5% гірше), але перевершує його за швидкістю на 8-10%. Метод MST Prima має найкращу швидкість – за цією характеристикою він перевершує майже всі інші методи, крім запропонованого методу daiNET. Крім того, метод MST Prima характеризується високою точністю групування і на 1-3% гірший C-means за цією характеристикою, тоді як зниження точності групування спостерігається лише при роботі в багатовимірному просторі символів, де класифіковані об'єкти характеризуються декількома десятками чи сотнями ознак.

Слід зазначити, що daiNET характеризується найбільшою швидкістю і за цим параметром перевершує метод MST Prima в середньому на 2%. За точністю групування daiNET можна порівняти з методами C-means, ВСА та aiNETmc. Таким чином, daiNET є найбільш збалансованим порівняно з еталонними методами MST Prima та C-means. Крім того, що є дуже важливим, використання цього методу дозволяє автоматизувати процес визначення кількості класів, що неможливо при використанні MST Prima і C-means.

Висновки

У статті розглядається вирішення нагальної задачі класифікації об'єктів на основі імунного підходу. Запропонований метод класифікації є модифікацією aiNET і використовує особливості моделі штучної імунної мережі та математичного апарату теорії графів, характерних для методів MST. Цей метод відрізняється від більшості існуючих імунних методів (aiNET, CLONALG, ВСА) тим, що застосовуються для класифікації, такими ознаками:

- метод визначення спорідненості між об'єктами;
- використання характеристик авідності та валентності;
- спрощення етапів клонування та мутації;
- використання моделі теорії графів для виділення нових класів.

Спрощення методу визначення спорідненості між об'єктами призводить до збільшення швидкості алгоритму класифікації, що працює на основі daiNET. Використання математичної моделі теорії графів при формуванні мережі антитіл дозволяє зменшити кількість обчислювальних операцій на етапах відбору клонів і супресії, що також призводить до збільшення швидкості класифікації. Використання принципу досяжності будь-якого мережевого об'єкта за

допомогою фіксованої кількості операцій клонування та мутації зменшує вартість пам'яті для зберігання популяції клонів.

Для підвищення продуктивності daiNET розбиває набір класифікованих об'єктів на кілька наборів – набір об'єктів, які можна класифікувати за допомогою класифікатора, та багато об'єктів, які вимагають формування нових класів. Це дозволяє скоротити час, витрачений на імунну підготовку для об'єктів, спорідненість яких дозволяє їх класифікувати за допомогою початкового набору класів. Щоб збільшити швидкість ізоляції кластера від багатьох об'єктів, які неможливо призначити жодному з вихідних класів, daiNET використовує критерій мінімально допустимої спорідненості, який використовується для зменшення кількості з'єднань між об'єктами імунної мережі. Через це всі сформовані кластери не зливаються, тобто не містять спільних об'єктів між собою та з вихідними класами.

З результатів експериментів, проведених під час тестування daiNET, випливає, що запропонований метод характеризується високою швидкістю та високою точністю класифікації.

Література

1. Duda R.O. Pattern classification / R.O. Duda, P.E. Hart, D.G. Stork. – Wiley & Sons, 2010. – 738 p.
2. Lars Eldén. Matrix Methods in Data Mining and Pattern Recognition, Second Edition, SIAM, 2019. – 243 p.
3. Dasgupta D. Immunological computation, Theory and applications / D. Dasgupta, L.F. Nino – Taylor & Francis Group, 2009. – 278 p.
4. Кораблев Н.М. Классификация данных с использованием иммунной модели клонального отбора / Н.М. Кораблев, А.А. Фомичев // Інформаційні системи та технології: монографія / за заг. ред. В.С. Пономаренка. – Х.: ФОП Бровін О.В., 2019. – С. 185-199.
5. Кораблев Н.М. Классификация данных с использованием модели искусственной иммунной сети / Н.М. Кораблев, А.А. Фомичев // Інформаційні технології: сучасний стан та перспективи: монографія. За заг. ред. В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – С. 86-101.
6. Abul Abbas, Andrew Lichtman, Shiv Pillai. Basic Immunology 4th Edition, Functions and Disorders of the Immune System – Saunders, 2014. – 336 p.

ГЛАВА 6

ДОСЛІДЖЕННЯ ПОСИЛАЛЬНИХ МЕТОДІВ РАНЖУВАННЯ ДОКУМЕНТІВ ЗА РЕЗУЛЬТАТАМИ ІНФОРМАЦІЙНОГО ПОШУКУ

Вступ та постановка задачі

Під інформаційним пошуком зазвичай розуміють процес реалізації операцій, необхідних для отримання результатів (текстових або текстографічних документів, аудіофайлів, відеофайлів тощо), що повністю (або в значній мірі) відповідають запитам користувачів інформаційно-пошукових систем (ІПС) [1, 2]. Ефективний пошук інформації в ресурсах мережі Інтернет в міру збільшення її обсягу і розосередження її джерел стає все складнішим. При цьому критичним є не стільки час пошуку, скільки відбір релевантної запитам користувачів інформації. Динамічність інформаційного середовища мережі Інтернет, що вимагає постійного оновлення відомостей про наявність і місце розташування інформації, ускладнює завдання.

Інформаційний пошук дозволяє здійснити зіставлення запиту користувача відомостями про ІР, якими ІПС, до якої надійшов цей запит. Процес пошуку базується на використанні певної моделі пошуку. Модель пошуку характеризується наступними параметрами:

- форма подання документів і запитів;
- критерій змістовної відповідності;
- методи ранжування результатів запитів;
- механізм зворотного зв'язку для оцінки релевантності документів.

Існує багато способів оцінити якість результатів пошуку за допомогою ІПС документів, що відповідають запиту користувача. Поняття ступеня відповідності знайдених документів запиту (релевантності) є суб'єктивним поняттям, а ступінь такої відповідності залежить від конкретного методу оцінювання результатів виконання запиту.

Актуальною є задача дослідження сучасних методів ранжування текстових документів за рівнем релевантності запитам користувачів.

Метою статті є аналіз та експериментальне дослідження посилальних методів ранжування за результатами пошуку текстових документів, що відповідають запиту користувача, а також розробка та програмна реалізація багатоетапного методу ранжування, що має певні переваги в порівнянні з

існуючими методами.

Показники оцінювання якості пошуку документів, релевантних запитам користувачів

Ефективність пошуку представляє собою оцінки різних показників результатів пошуку [3]. Однак ефективність, як і оцінки, багато в чому залежать від релевантності. Релевантність визначають документи, зміст яких відповідає запиту користувача. Другою оцінкою показників результатів пошуку є пертинентність знайдених документів, яка визначає відповідність їх змісту інформаційній потребі.

ПС всередині результатів пошуку намагається розташувати елементи колекції документів в порядку їх відповідності до обробленого користувачем запиту, тобто в порядку їх релевантності. При цьому ймовірність відсутності в пошуковій видачі більш відповідного запиту користувача документа, або його значної віддаленості в упорядкованому списку результатів обернено пропорційна рівню ефективності пошуку.

Однак співвідношення між запитом і документом може бути як між безпосередньо запитом і документом, так і між їх формалізованими поданнями (для документа даним поданням є пошуковий образ документа (ПОД), а для запиту – пошуковий образ запиту (ПОЗ)). Релевантність поділяють на два види, в залежності від того, співвідношення між якими об'єктами необхідно встановити, релевантність поділяють на два види: якщо мова йде безпосередньо про запити і документи, то така релевантність буде змістовою, якщо аналізуються ПОЗ і ПОД, то релевантність буде формальною.

В автоматизованих ПС пошук заснований на формальній релевантності. Якщо запит від користувача створений точно і детально, то релевантні результати, швидше за все, будуть пертинентними. В ідеальному випадку ПС має видавати тільки релевантні документи, однак, на практиці спостерігається або недостача, або видача не повністю релевантних документів (при цьому за результатами пошуку користувач може отримати і нерелевантні і непертинентні колекції документів).

Найбільш поширеними показниками оцінювання якості пошуку документів, релевантних запитам користувачів, є точність, повнота, випадання

та F-міра

Під точністю (precision) зазвичай розуміють відношення числа релевантних документів, знайдених ПС, до загальної кількості знайдених документів:

$$\text{Precision} = \frac{|D_{rel} \cap D_{retr}|}{|D_{retr}|}, \quad (1)$$

де $|D_{rel}|$ – кількість релевантних документів; $|D_{retr}|$ – загальна кількість знайдених документів.

Повнота (recall) оцінюється як відношення числа знайдених релевантних документів, до загальної кількості релевантних документів в базі:

$$\text{Recall} = \frac{|D_{rel} \cap D_{retr}|}{|D_{rel}|}. \quad (2)$$

Випадання (fall-out) характеризує ймовірність знаходження нерелевантного ресурсу і визначається як відношення числа знайдених нерелевантних документів до загальної кількості нерелевантних документів в базі:

$$\text{Fall-out} = \frac{|D_{nrel} \cap D_{retr}|}{|D_{nrel}|}, \quad (3)$$

де $|D_{nrel}|$ – кількість знайдених нерелевантних документів.

Іноді буває корисно об'єднати точність і повноту в одному усередненому показнику. Для цього не доцільним є використання середнього арифметичного значення (наприклад, якщо пошуковій системі повернути взагалі всі документи, щоб забезпечити рівну одиниці повноту при близькій до нуля точності, то середнє арифметичне точності і повноти буде не менше 0,5). Середнє гармонійне

значення не має такого недоліку, оскільки при суттєвій різниці усереднених значень наближається до мінімального з них.

Тому прийнятною мірою спільної оцінки точності і повноти є F-міра, яка визначається як зважене гармонійне середнє точності P і повноти R:

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}}, \alpha \in [0,1]. \quad (4)$$

Слід зазначити, що інформація в науково-технічних документах підпорядковується законам розсіювання, внаслідок чого повнота і точність залежать одна від одної, і при збільшенні однієї зменшується інша. У реальних системах коефіцієнт повноти може досягати 70%, а коефіцієнт точності пошуку буде в дуже широких межах, іноді знижуючись до 10%. Подібні характеристики залежать від внутрішніх властивостей ІПС, специфічності запитів, повноти запиту та ступіню відповідності запиту інформаційній потребі.

Оцінка релевантності визначає ступінь відповідності документів запиту, а їх подальше сортування реалізується різними пошуковими системами по-різному, і здійснюється з використанням різних методів ранжування.

Методи ранжування результатів пошуку

Кожна ІПС має набір правил та інструкцій, за якими визначається релевантність сторінки або документа. При цьому алгоритми ранжування постійно змінюються. Якщо раніше в ранжованому документі або сторінці достатньо було вказати тільки тематику і опис, то тепер це не дає потрібного результату через занадто велику кількість документів, які можуть повторюватися за даними параметрами. Додання до документів ключових слів також не підвищує суттєво ефективність пошуку та ранжування. Це спричинило розвиток досліджень у розвитку методів ранжування, і сучасні пошукові системи при визначенні найбільш релевантних документів розглядають додаткові фактори. До таких факторів слід віднести:

- кількість ключових слів та частота їх використання;
- відношення ключових слів до загального обсягу вмісту ;
- вміст документа;
- індекс цитованості документа;

– обсяг переходів по даному запиту.

Ранжування знайдених ІПС документів передбачає реалізацію таких етапів [4]: визначення вихідної колекції документів; обчислення релевантності кожного документа; сортування за зменшенням оцінок релевантності; видача необхідної кількості відсортованих за оцінкою релевантності документів (з її найвищим значенням).

Особливість використання зазначених вище факторів для оцінки релевантності залежить від обраного методу ранжування. Розглянемо найбільш поширені в ІПС класи методів ранжування, а саме: *текстові*, які використовують при виконанні алгоритму ранжування вміст самого документа, та *посилальні*, що використовують посилання, які знаходяться в документі або вказують на документ.

В текстових методах ранжування використовується поняття «терм» документа (зазвичай, це кожне слово в ньому). Ці методи спираються на такі властивості термів, як частота появи, місце розташування, процентне співвідношення терма з іншим вмістом, наявність в особливих елементах підвищеної важливості (наприклад, в заголовках).

Найбільш популярним текстовим методом ранжування є метод TF-IDF. Для його використання документ представляється у вигляді векторної моделі, що містить в собі всі терми. При цьому, чим більше локальна частота терма, що містить в собі запит (TF – Term Frequency), в певному документі, і більше «рідкість» терма у всій колекції документів, тим вище вага даного документа по відношенню до терму. За таким принципом документи, що мають найбільшу вагу по відношенню до конкретного терму, будуть видані першими в результатах пошуку.

Посилальні методи ранжування припускають аналіз позиції документа в мережі Інтернет. Позиція документа полягає в його значимості, яка обчислюється за кількістю посилань на нього з інших документів, що можна описати як «індекс цитованості». Основна посилальних методів ідея полягає в тому, що якщо документ найчастіше цитується, і на нього найчастіше посилаються, що він є більш значимим джерелом інформації. При цьому також необхідно враховувати посилання всередині документа, які ведуть на його окремі сторінки, а також посилання на документ авторитетних джерел, що автоматично також підвищить його ранг. Основою використання подібних методів ранжування є подання

документів і посилань між ними у вигляді посилального графа, в якому документи є вершинами, а посилання між ними – ребрами.

Алгоритми посилального ранжування аналізують цей граф, приділяючи особливу увагу зв'язкам між сторінками та таким властивостям сторінок-документів, як індекс цитованості, ймовірність читання користувачем даної сторінки при перегляді масиву сторінок, авторитетність. Алгоритми обробляють посилальний граф, обчислюючи ваги сторінок (числові значення властивостей, що беруться до уваги). Отримані результати сортуються за спаданням, і кожній сторінці призначається рейтинг у вигляді числового значення. Слід зазначити, що посилальне ранжування в більшості своїй не враховує запит користувача, не залежить від нього, і не звужує вихідну колекцію документів, виходячи з його вмісту. На практиці, визначення рейтингу сторінки проводиться при безпосередньому додаванні її в загальну мережу (при індексації) і змінюється або при зміні самої сторінки (додаванні нових посилань на інші ресурси), або при додаванні/зміні іншої сторінки (додаванні нових посилань на вихідний ресурс). В результаті рейтинги посилального ранжування створюють відносно статичний масив оцінок, існуючий як до призначеного для користувача запиту, так і після. Тому в ІПС, орієнтованих на подібні запити, спочатку проводиться текстовий пошук, використовуючи при цьому текстові методи ранжування (це дозволяє звужити кількість сторінок до найбільш релевантних за вмістом до запиту), після чого здійснюється сортування сторінок за рейтингом, визначеним за допомогою посилань (за методами посилального ранжування).

Розглянемо найбільш відомі методи посилального ранжування: PageRank і HITS [5]. Метод PageRank аналізує показник авторитетності (значущості) окремих сторінок відносно інших і використовує їх значення в якості ваг. Авторитетність можна пояснити наступними припущеннями: гіперпосилання зі сторінки А на сторінку В є визнанням авторитетності сторінки В з боку сторінки А (чим більше гіперпосилань веде на сторінку В, тим більше вона важлива або авторитетна). При цьому слід зазначити, що важливість сторінки В також визначається важливістю тих сторінок, які посилаються на неї. Таким чином, обчислення ваги сторінки при посилальному ранжуванні з використанням методу PageRank проводиться шляхом підрахунку важливості посилань (безпосередніх сторінок) на неї. На рис.1 наведено приклад посилального графу з зазначенням авторитетності кожної вершини у процентах.

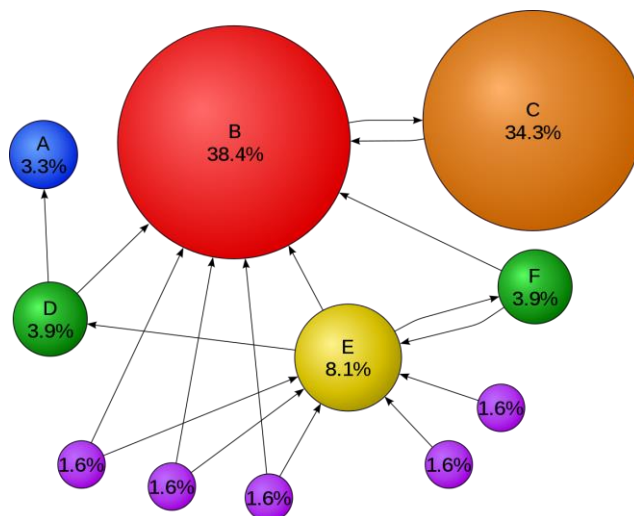


Рис. 1. Приклад посилального графу з зазначенням авторитетності вершин

У загальному випадку, підрахунок значення показника важливості PageRank кожної сторінки здійснюється таким чином:

$$PR(P_i) = \sum_{P_j \in V_i} \frac{PR(P_j)}{L_j}, \quad (5)$$

де PR – рівень важливості; P_i – аналізована сторінка; V_i – набір сторінок, що посилаються на сторінку P_i ; P_j – сторінка з набору V_i ; L_j – кількість посилань на сторінці P_j .

Показник PageRank може бути розрахований як ітеративно, так і алгебраїчно. Ітеративний метод передбачає обчислення важливості кожної сторінки t раз поспіль, використовуючи при цьому попередні значення, і закінчення виконання підрахунку важливості тоді, коли різниця між поточним значенням і минулим буде в межах деякої заданої похибки (числа від 0 до 1).

Іноді до класичної формули обчислення PageRank додають також коефіцієнт загасання d , значення якого визначає ймовірність того, що користувач, що зайшов на сторінку, все ж перейде по одному з посилань, що містяться на цій сторінці, а не вийде з посилального графу. Стандартним значенням коефіцієнта загасання є 0.85.

З урахуванням коефіцієнту згасання формула (5) приймає такий вигляд:

$$PR_t(P_i) = \frac{1-d}{N} + d \sum_{P_j \in B_i} \frac{PR_{t-1}(P_j)}{L_j}, \quad (6)$$

де N – загальна кількість сторінок.

Модифікації методу PageRank широко використовуються в багатьох сучасних пошукових системах (Google, Yandex, Bing).

Основними перевагами методу PageRank є: враховування множини інших сторінок при підрахунку важливості певної сторінки; динамічна оцінка (в результаті будь-яких подій значення PageRank може змінитися).

Недоліками метода PageRank є: існування різних шляхів штучного збільшення показника важливості, якщо PageRank не модифікований для запобігання цьому; відносно повільна швидкість отримання актуального значення оцінки.

Метод HITS, як і PageRank, використовує поняття авторитетності документа-сторінки (чим більше документів посилаються на певний документ, тим вище його показник авторитетності, і тим вищий рейтинг він отримає).

Разом з цим тут вводиться поняття посередницької оцінки, згідно з якою (одночасно з авторитетністю) документи також можна оцінити за кількістю посилань, що містяться в них, на авторитетні документи. Таким чином, метод HITS реалізує дворівневу систему ранжування, де у визначенні авторитетності сторінки задіяні також і посередницькі оцінки, і навпаки – для кожної сторінки обчислюється як оцінка авторитетності, так і посередницька оцінка. При цьому слід зазначити, що даний алгоритм, будучи посилальним, все ж залежить від запиту користувача, і визначення авторитетності сторінок здійснюється для кожного користувачького пошукового запиту.

Першим кроком методу HITS є обов'язкове звуження вихідного масиву сторінок до найбільш релевантного пошуковому запиту кількості. Даний крок реалізується за допомогою текстових методів ранжування, наприклад, за векторною моделлю, що використовує для підрахунку ваг термів оцінку TF-IDF. Результуюча відібрана множина сторінок називається кореневим набором (root set). Наступною дією є розширення кореневого набору до базового шляхом додавання в кореневий набір: всіх сторінок, на які вказують посилання зі сторінок кореневого набору; деяких сторінок, які посилаються на сторінки кореневого набору.

Приклад базового набору сторінок наведено на рис 2.

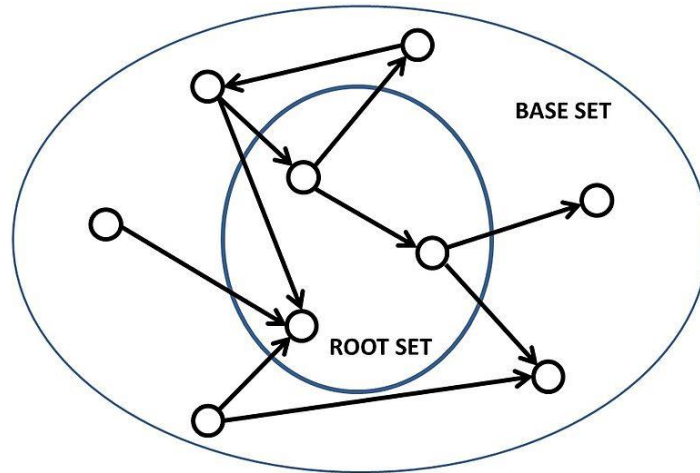


Рис. 2 Приклад базового набору сторінок для метода HITS

Даний набір утворює підграф, а обчислення всіх необхідних показників проводиться саме в цьому підграфі. Далі відбувається безпосереднє обчислення показників авторитетності і посередництва. Як і в методі PageRank, даний крок виконується тут ітеративно, використовуючи при цьому значення з попередніх ітерацій. Після останнього кроку початкові значення кожного з показників сторінки встановлюються самою системою без розрахунків, і дорівнюють одиниці для кожного показника.

Першим етапом кожної ітерації є оновлення авторитетності шляхом підсумовування посередницьких оцінок кожної зі сторінок, що вказують на них:

$$\text{auth}_{t+1}(P_i) = \sum_{j \in B_i} \text{hub}_t(P_j), \quad (7)$$

де t – поточна ітерація; auth – оцінка авторитетності; P_i – поточна сторінка; B_i – множина сторінок, що вказують на сторінку P_i ; hub – посередницька оцінка; P_j – сторінка з множини сторінок B_i .

Далі відбувається оновлення посередницької оцінки шляхом підсумовування авторитетних оцінок кожної зі сторінок, на яку вона посилається:

$$\text{hub}_{t+1}(P_i) = \sum_{j \in B_i} \text{auth}_t(P_j), \quad (8)$$

P_i – аналізована сторінка; B_i – набір сторінок, що посилаються на сторінку P_i ; P_j – сторінка з набору B_i ; L_j – кількість посилань на сторінці P_j .

Останнім кроком кожної ітерації є нормалізація кожної з отриманих оцінок. Як і у методу PageRank, ітеративні обчислення припиняються, коли різниця між поточними значеннями і минулими буде в межах деякої граничної похибки (від 0 до 1), після чого сторінки сортуються за показником авторитетності і ранжування вважається виконаним.

Методу HITS часто використовується для знаходження тематичних документів в мережі Інтернет (на тематичних форумах при потребі в пошуку всіх тематично зв'язаних документів).

Основними перевагами HITS є: врахування не тільки вхідних у вершину посилань, як у PageRank, а й вихідних, що дає можливість обчислення і використання її в подальшому додаткового показника (посередницької оцінки); старіші сторінки не будуть пріоритетними в результуючому списку; можливість ранжувати сайти на основі посилань, релевантних запитам користувача.

Основними недоліками HITS є: недостатня швидкість (якщо метод PageRank обчислює необхідні показники вже на етапі створення сторінки, то HITS робить це після визначення базового набору); зсув тематики (у випадку, коли деякі документи сильно зв'язані між собою, та частина з них не релевантна до запиту, вони все одно можуть зсувати релевантні документи донизу списку результатів, так як при формуванні базового набору все ж попадають у граф та мають більший авторитет за рахунок тематичних зв'язків).

Багатоетапний метод посилального ранжування

Суть пропонованого багатоетапного методу полягає в проходженні вихідної колекції документів або масивів сторінок через кілька послідовних етапів ранжування. Дані етапи незалежні один від одного, а списки результатів стають багато в чому більш точними і релевантними, адже при кожній подібній ітерації показники відповідності контенту, властивостей документів і пошукового запиту уточнюються. Також при цьому з'являється можливість більш широкого охоплення понять і мети, закладених безпосередньо в пошуковий запит, враховуючи нових факторів.

Ключовим компонентом пошуку є користувальницький запит. Так як даний метод є залежним від нього, а етапи ранжування – незалежними один від одного, кожна ітерація проходження етапів ранжування проводиться з використанням цілого ряду запитів, що відрізняються деякими змінами один від одного (додаванням, перестановкою, заміною слів). Незважаючи на це, для отримання більш коректних результатів, передбачається, що кожна зміна буде синонімічною до початкового запиту.

Першими кроками багатоетапного методу ранжування є визначення пошукового запиту, і його відправка в пошукову систему. Більшість пошукових систем використовує декілька методів ранжування, кожен з яких обробляє певну властивість документа-сторінки. Після отримання першого результуючого списку, відсортованого пошуковою системою, в неї знову надсилається запит, подібний по сенсу до попереднього, але з дещо зміненим змістом, з метою отримання нового ранжованого списку від пошукової системи, в якому будуть як елементи з попереднього списку, так і зовсім нові.

Кожен результуючий список документів, виданий як відповідь на певний запит з їх послідовності, буде порівнюватися з попереднім на предмет однакових елементів. При позитивному випадку даний елемент фіксується в наступних результуючих списках, і буде промаркований – йому буде присвоєна деяка мітка, наприклад, номер. При виявленні даного елемента в наступних результатах, у відповідь на нові змінені запити з аналогічним змістом, він буде отримувати додаткову мітку, або присвоєне йому число буде збільшено на одиницю. При зупинці введення запитів, результуючий список буде містити множину промаркованих документів, і документи з найбільшою кількістю міток або найбільшим доданим числом будуть найбільш релевантними безпосередньо суті змісту запиту.

Для вимірювання показників якості даного методу ранжування можна вимірювати як точність метода, так і повноту (якщо у використанні є готовий набір даних). Однак результати ми отримуємо послідовно, використовуючи декілька запитів та отримуючи декілька списків в якості результатів, які пізніше обробляються, і створюють новий список тих посилань, які знаходилися в декількох пошукових видачах. У подібному випадку не зовсім правильно використовувати звичайну точність з двох причин. По-перше, список повторюваних результатів на кожній ітерації буде лише збільшуватися. По-друге,

в подібному списку велике значення має місце документа серед інших – тому що поява даного документа в черговій пошуковій видачі буде інкрементувати кількість його міток, і він буде поміщений вище, ніж був до цього, що показує його збільшену важливість. Алгоритм роботи багатоетапного методу ранжування наведено на рис. 3.



Рис. 3. Алгоритм багатоетапного методу ранжування

Середня точність на k елементах у результуючому списку (average precision at K , $ap@k$), розраховується таким чином:

$$ap@k = \frac{1}{n} \sum_{i=1}^k P(i) \times rel(i), \quad (9)$$

де n – кількість релевантних документів; k – кількість документів у результуючому списку; $P(i)$ – точність на k елементах; $rel(i)$ – функція-індикатор, яка дорівнює 1, якщо i -тий документ списку є релевантним, та 0, якщо ні.

Дана формула передбачає підсумовування точності на певній кількості документів, помножену на функцію-індикатор. Показник точності дозволяє враховувати положення документів в підсумковому списку. Для того, щоб враховувати також кількість запитів, посланих в систему, використовується ще одна метрика точності під назвою mAP (mean average precision):

$$mAP = \frac{\sum_{q=1}^Q ap(q)}{Q}, \quad (10)$$

де Q – кількість запитів; q – номер певного запиту.

Метрика (10) використовує показники середньої точності результатів кожного виконаного запиту, що дозволяє виявити точність багатоетапного методу ранжування.

Подібний метод ранжування доцільно застосовувати, коли велику важливість має релевантність отриманих результатів запиту, яка, в даному випадку, залежить від кількості міток. Перевагами багатоетапного методу ранжування можна назвати: збільшення показників релевантності завдяки ітеративному проходженню ранжування; пошук відповідності документів не до простого текстового вмісту запиту, а до його змісту.

Основним недоліком даного методу є відсутність те, що користувач має самостійно вводити і змінювати запити, трансформуючи кожен з них в синонімічний до вихідного.

Експериментальне дослідження методів ранжування

Для програмної реалізації і моделювання обраних для дослідження методів ранжування були використані:

- мови програмування Python (для реалізації методів PageRank та HITS), C# (для реалізації багатоетапного методу);
- зовнішні бібліотеки NetworkX; Scikit-learn; Google Custom Search API.
- набори даних – «Friendship» та «Enron» (кількість вершин – 29, кількість ребер – 377).

Для ітераційного обчислення показника PR (за методом PageRank) кожної з вершин за формулою (8) досліджено два варіанти – з використанням готового набору даних і з використанням випадкового графа.

У першому випадку використовувалася запрограмована функція `iterate_PageRank()`, параметрами якої є сама матриця суміжності, значення коефіцієнта загасання, значення межі похибки і максимальна кількість ітерацій (рис. 4).

```
class SparsePageRank(SparsePageRank):
    def iterate_PageRank(self, b=0.15, epsilon=1e-9, maxIters=100):
        q = np.ones(self.n) / self.n
        old_p = q
        residuals = []
        for t in range(maxIters):
            p = (1 - b) * (self.nAT.dot(old_p)) + (b * q)
            residual = np.linalg.norm(p - old_p, 1)
            residuals.append(residual)
            old_p = p
            if residual < epsilon:
                break
        return p, residuals
```

Рис. 4. Реалізація функції `iterate_PageRank()`

У разі випадкового графа, створеного за допомогою бібліотеки `NetworkX`, здійснювалося ітераційне обчислення показників `PageRank` кожної вершини за допомогою функції `pagerank()` самої бібліотеки. Фрагмент отриманих під час тестування ранжованих результатів обробки посилального графа «Friendship» методом `PageRank` наведено на рис. 5


```

PageRank vector
[0.02182012 0.03968736 0.03936064 0.01384059 0.03088527 0.04461635
 0.01657996 0.05707779 0.05576658 0.03271251 0.05580039 0.04697743
 0.05088893 0.04899874 0.04520136 0.02710907 0.01760596 0.01563257
 0.01497605 0.04701234 0.05267241 0.01979954 0.04876799 0.02708614
 0.03558235 0.03635852 0.02315413 0.01553116 0.01849764]
Top-5 rankings based on PageRank vector:
  node_id  score
0         7 0.057078
1         10 0.055800
2          8 0.055767
3         20 0.052672
4         12 0.050889

```

Рис. 5. Ранжовані результати обробки посилального графа «Friendship» методом PageRank

Для ітераційного обчислення показників авторитетності і посередницької оцінки PR (за методом HITS) за наборами даних «Friendship» та «Enron» використовувалась запрограмована функція `iterate_HITS()` (рис.6).

```

class SparseHITS(SparseHITS):
    def iterate_HITS(self, epsilon=1e-9, maxIters=100):
        old_h = np.ones(self.n) / self.n
        old_a = np.ones(self.n) / self.n
        h_residuals = []
        a_residuals = []
        for t in range(maxIters):
            h = self.A.dot(old_a)
            a = self.AT.dot(h)
            h = h / np.linalg.norm(h, 2)
            a = a / np.linalg.norm(a, 2)
            h_residual = np.linalg.norm(h - old_h, 1)
            a_residual = np.linalg.norm(a - old_a, 1)
            h_residuals.append(h_residual)
            a_residuals.append(a_residual)
            old_h = h
            old_a = a
            if h_residual < epsilon and a_residual < epsilon:
                break
        return h, a, h_residuals, a_residuals

```

Рис. 6. Реалізація функції `iterate_HITS()`

Для реалізації багатоетапного методу ранжування було використано бібліотеку Google Custom Search AP, що містить крім назви і посилання на

сторінку, дані про ідентифікатор кеша, зображення, та формат [6]. Після виведення всіх знайдених в поточній ітерації результатів пошукового формувалися ранжовані результати. Після цього цикл повторювався, і користувач знову мав ввести новий запит. В кінцевому підсумку це дозволяло знайти такі результати, які є найбільш релевантні до суті самого пошукового запиту, а не до його текстового вмісту, для чого здійснювалася послідовна відправка синонімічно схожих до початкового запитів. В якості значення k були обрані 10 перших елементів списку повторних результатів. Результати значень середньої точності багатоетапного методу для k елементів наведено в табл. 1.

Таблиця 1

Значення середньої точності після виконання запиту

Порядковий номер запиту	Середня точність
1	0.574
2	0.665
3	0.87
4	0.942
5	0.966
6	0.992

Результати тестування підтвердили ефективність методу.

Висновки та перспективи подальших досліджень

В роботі були досліджені посилальні методи ранжування документів за запитами користувачів PageRank, HITS, а також багатоетапний метод ранжування. Для реалізації обчислювальних алгоритмів досліджуваних методів були здійснені практичні реалізації кожного з методів ранжування, використовуючи зовнішні бібліотеки. Для ітераційного обчислення показників авторитетності і посередницької оцінки за наборами даних «Friendship» та «Enron» використовувалися додатково розроблені функції `iterate_PageRank()` та `iterate_HITS()`. Для реалізації багатоетапного методу ранжування було використано бібліотеку Google Custom Search AP, що містить різні технічні дані результатів пошуку. Результати тестування кожного з методів були проаналізовані на релевантність до запиту або коректності показань, а також

оброблені для отримання значень метрик якості ранжування, зокрема, показників mAr і точності на k елементах результатів.

Перспективним продовженням досліджень слід вважати розробку та тестування процедур комбінованого використання методів PageRank, HITS та багатоетапного методу ранжування.

Література

1. Turnbull D. Relevant Search: With applications for Solr and Elasticsearch / D. Turnbull. // Manning Publications. – 2016. – 360 с.
2. Long B. Relevance Ranking in Vertical Search Engines/ B. Long , Y. Chang. // Morgan Kaufmann. – 2014. – 264 с.
3. Чала Л.Е. Модифікований метод пошуку релевантних веб-документів з використанням подвійного зважування / Л.Е. Чала, Ю.Ю. Харитонova // IES-2014. Proceedings of the ninth international scientific-practical conference. Vinnytsia, VNTU, 2014. – С. 18-20.
4. Ward E. Ultimate guide to link building: build backlinks, earn a higher search engine rank, increase the authority and popularity of your site / E. Ward, G. French // Entrepreneur Press. – 2013. – 215 с.
5. Patel P. A Review of PageRank and HITS Algorithms / P. Patel, K. Patel. // International Journal of Advance Research in Engineering, Science & Technology (IJAREST). – 2015. – 254 с.
6. Google Custom Search API for Developers. [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/custom-search/v1/overview>.

ГЛАВА 7

МЕТОДОЛОГІЯ SAFE ТА ЇЇ ВИКОРИСТАННЯ ДЛЯ ВПРОВАДЖЕННЯ ВИКОНАННЯ ЦІЛЕЙ КІБЕРБЕЗПЕКИ У ГНУЧКОМУ ПІДХОДІ ДО РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вступ

В епоху програмного забезпечення – взаємопов’язаного світу реального часу, в якому кожна галузь залежить від технологій, а кожна організація (принаймні частково) є програмною компанією. Щоб залишатися конкурентоспроможними, підприємствам потрібно буде цифрово трансформувати свою діяльність, бізнес-рішення та досвід клієнтів. Більшою проблемою на багатьох підприємствах є те, що їхні поточні бізнес-моделі, організаційна ієрархія та технологічна інфраструктура не можуть йти в ногу з необхідними швидкими змінами.

Швидкий розвиток забезпечив значні покращення для багатьох організацій. Однак самого по собі навіть швидкого розвитку недостатньо. Початок розробки програмного забезпечення тепер повинен розширитися, щоб охопити все підприємство, змінивши спосіб роботи людей та функціонування кожного аспекту бізнесу. Простіше кажучи, підприємствам потрібна спритність бізнесу, визначальний фактор, який визначатиме переможців та переможених у цифровій економіці.

Надаючи людям можливість швидко приймати рішення, включаючи розподіл ресурсів та вирівнювання потрібних людей навколо правильної роботи, спритність бізнесу дозволяє компаніям використовувати свої нові можливості. Однак досягнення такого рівня спритності вимагає освоєння не однієї, а двох ділових операційних систем

SAFE може допомогти підприємству поліпшити результати бізнесу, прискоривши продуктивність, час виходу на ринок, якість, залучення працівників тощо. Результати можуть бути суттєвими, як це проілюстровано на рис. 1.



Рис. 1. Бізнес–переваги SAFe

SAFe призначений для того, щоб допомогти організаціям синхронізувати гнучку розробку програмного забезпечення. Зокрема, на великих проектах SAFe допомагає командам працювати над досягненням спільної мети, скорочуючи час виконання поставленого завдання і допомагаючи їм швидше реагувати на зміни на ринку.

Масштабування проекту

Гнучка методологія розробки – корисна річ, але її використання не вирішує проблем зі структурою команди або бізнес–завданнями, такими як комунікації між продуктивним і торговим відділом. Розуміння Agile як частини бізнес–процесу, а не просто теорії розробки програмного забезпечення дуже важливо для досягнення успіху.

Ще одна важлива помилка в масштабуванні організації – наявність зайвої кількості людей, які працюють з одним кодом. Коли команда складає більш ніж 15–20 інженерів, зв'язку між ними і координація починають страждати. Розбіжності виникають в плануванні ресурсів, субординації і прийнятті рішень. Ці конфлікти займають відведений на виробництво нового функціоналу час, що знижує цінність продукту для клієнтів.

Оскільки потреба в масштабуванні Agile–практик зростає, зростає плутанина щодо того, які з них є кращими, а які дійсно відповідають потребам.

На перший погляд, більшість масштабованих підходів можуть виглядати однаково, але насправді вибір правильного – ключ до успішного застосування. Згідно з 14th Annual State of Agile Report [1] SAFe є найбільш використовуваною методологією розробки програмного забезпечення. Розподіл зображено на рис. 2.

SCALING AGILE

SCALING METHODS AND APPROACHES

The Scaled Agile Framework* continues to be the most popular scaling method cited by respondents (35% this year compared to 30% last year). As a percentage of all responses, SAFe* outdistances the next nearest response, Scrum of Scrums, by 19%.



Рис. 2. Розподіл застосування методологій розробки ПО для великих проектів

В даний час у версії 5.0 SAFe інтерпретує безпеку як нефункціональну вимогу поряд з іншими «якостями», такими як ремонтпридатність і продуктивність. Незважаючи на те, що це традиційний засіб забезпечення безпеки в формальних моделях розробки програмного забезпечення, нефункціональні вимоги часто важко задовольнити, оскільки виконання робіт постійно відкладають. Виконавчі цілі і безпека не відповідають стратегії короткострокового прибутку, яка, як правило, не узгоджується з боротьбою проти гіпотетичних втрат в більш віддаленому майбутньому.

SAFe – це комбінація Lean, Agile і DevOps. SAFe розшифровується як Scaled Agile Framework, чи зі змінним розміром.

Компетентність Lean Portfolio Management (LPM) поєднує стратегію та виконання, застосовуючи Lean, Agile та системне мислення. Для досягнення цього необхідні три співпраці – стратегія та інвестиційне фінансування, оперативні портфельні операції та економне управління [2].

Принципи SAFe та мислення Lean–Agile сприяють децентралізації реалізації стратегії на розширені можливості АРТ та поїздів рішень. Однак навіть тоді потрібно застосовувати системне мислення, щоб забезпечити узгодженість АРТ та поїздів рішень та функціонування в більш широкому контексті підприємства.

Бережливе управління забезпечує нагляд за витратами, аудит та дотримання вимог, прогнозування витрат та вимірювання. Управління портфелем вимагає співпраці між власниками бізнесу та архітекторами проєктів, щоб динамічно прогнозувати та складати бюджети, координувати постійне дотримання та вимірювати ефективність портфеля за допомогою Lean метрик.

Фреймворк містить чотири конфігурації. Чим більше людина працює в організації, чим складніше продукт, тим більше потрібно інструментів для ефективної організації роботи та, відповідно, вибирається більш складна конфігурація.

Essential SAFe

Це основа фреймворка, яка є мінімальним набором інструментів, необхідний для отримання результату. На цій базовій конфігурації засновані всі інші. Підходить організаціям, які працюють над одним продуктом середньої або високої складності.

Portfolio SAFe

В рамках цієї конфігурації йде розробка кількох продуктів середньої або високої складності. Виникає необхідність в управлінні портфелем, де приймаються рішення щодо розподілу бюджету між потоками, рішення про покупку або злиття з іншими компаніями, створення нових напрямків бізнесу та закриття старих.

Large Solution SAFe

Підходить організаціям, які займаються розробкою одного великого, комплексного вирішення декількома командами команд. Створюються плани робіт на 12–36 місяців, проводиться аналіз економічної доцільності змін.

Full SAFe

В рамках цієї конфігурації йде розробка кількох комплексних і складних рішень. Залучаються всі рівні.

Базова конфігурація складається з двох рівнів – рівня команди і рівня програми. На рівні команди робота здійснюється по Scrum, Kanban, XP.

На уровне программы вводятся новые роли.

Ролі SAFe на рівні програми

Роль	Опис
Менеджмент продукту	Один або кілька людей, що визначають напрямок розвитку продукту, відповідають за беклог продукту
RTE (Release Train Engineer)	Аналог ролі скрам-майстра.
ART (Agile Release Train)	Відповідає за координацію і організацію процесу роботи не окремої команди, а програми
System Architect / Engineer	Команда команд (50–125 осіб), яка поступово розробляє і поставляє рішення в потоці цінності

Використовується багато термінології, пов'язаної з поїздами: ART (Agile Release Train), RTE (Release Train Engineer). Це пов'язано з тим, що робота команд

в якійсь мірі схожа на роботу поїзда – є стабільне розклад. Якщо ви не встигаєте на один поїзд, то завжди можна сісти на наступний. Якщо в поточний інкремент не виходить вмістити якісь цілі, то їх можна помістити в наступний.

Організація роботи на рівні програми в SAFe нагадує організацію роботи по скрам, тільки в більшому масштабі, з плануванням на довший період.

Таблиця 2

Порівняння етапів роботи в Scrum і SAFe на рівні програми

Scrum	SAFe Program Level
Спринт (1–4 тижні)	Інкремент програми (8–12 тижнів)
Планування спринту	Планування інкремента програми
Щоденний скрам	Додаткові зустрічі для синхронізації команд, власників і менеджменту продукту
Огляд спринту	Демонстрація системи
Ретроспектива спринту	Інспекція та адаптація

SAFe забезпечує безпеку завдяки моделювання загроз в процесі розробки архітектури додатку. Моделювання загроз можна розуміти як форму аналізу ризиків та аналізу проекту. Залежно від потреб організації і процесів її розвитку можуть застосовуватися різні методології. Приклади включають в себе систематизацію слабких місць, наприклад, Microsoft STRIDE.

Моделювання загроз, розгортається в рамках циклу взаємодії з клієнтом і потоків взаємодії з користувачем, зазвичай використовується неформальний, творчий процес, що приводить до «випадків неправильного використання» і «історій зловмисників» – ризикам для бізнесу.

Моделювання загроз для технічного проекту і його потоків даних призводить до більш механічного підходу, що призводить до конкретних змін дизайну і технічних заходів безпеки.

Передбачувані перешкоди для моделювання загроз:

- Час. Моделювання загроз – це процес, що вимагає часу, який є найдорожчим продуктом в команді розробників програмного забезпечення.
- Пріоритезація. Безпека і конфіденційність – два аспекти, які розробники повинні враховувати. Також необхідно вирішити питання управління архітектурними рішеннями і технічним боргом, продуктивністю, оптимізацією експлуатаційних витрат, дизайном послуг та виправленням помилок.

На щастя, є спосіб вирішити обидві проблеми.

Команди розробників програмного забезпечення керують своєю роботою через тікети (або елементи невиконаної роботи). Якщо завдання оформлена у вигляді тикета, значить її виконання стало очевидною і явною необхідністю. Потім цю потребу можна врівноважити і розставити за пріоритетами в порівнянні з усією іншою оплачуваною роботою.

Роботи щодо забезпечення безпеки традиційно не виносяться в окремі тікети. Швидше, вони залишилися набором вимог та настанов з безпеки, загальних критеріїв прийнятності або набором розпливчастих не функціональних обмежень. Це стає проблемою, коли команди стикаються з жорсткими термінами і тиском функцій, тому що – хоча правила і існували – явного розподілу часу не було.

Зробивши моделювання загроз (і всі інші неавтоматизовані дії щодо забезпечення безпеки) видимими у вигляді тикетів на розробку, можна відстежити розподіл часу. Аналіз і пропоновані заходи щодо усунення наслідків самі по собі також стають тикетами. Отже, безпека стає пріоритетнішою, займаючи місце за столом під час обговорення бізнес-пріоритетів.

Моделювання загроз в SAFe

Моделювання загроз – це діяльність, що відноситься до архітектурних етапах проектування SDLC. При використанні методології Agile для розробки програмного забезпечення етап проектування може бути не так чітко визначено, як при використанні інших методологій розробки, таких як Waterfall. Деякі з проблем можуть включати спроби пристосувати моделювання загроз до відносно коротким спринті.

Пастка може полягати в спробі створити моделі загроз для кожного спринту, а потім спробувати об'єднати їх, щоб змоделювати загрози для всієї системи. На початку набору спринтів відомо недостатньо для моделювання загроз для всієї системи, оскільки кожен спринт може вплинути на модель загроз.

DevOps зазвичай включає процеси розробки, які дозволяють часті зміни програмного забезпечення, аналогічні процесам гнучкої і безперервної інтеграції (CI), з упором на автоматизацію. Моделювання загроз в DevOps відрізняється від того, що можна побачити в Agile і CI, тим, що воно також включає в собі можливість частих змін в роботі і інфраструктурі, які зазвичай безпосередньо пов'язані з процесом розробки. Більшість процесів DevOps зводять до мінімуму або повністю виключають традиційні процеси аналізу змін і управління, які історично були типовими для інфраструктури послуг. Без надійного процесу управління змінами, що забезпечує шлюз для виявлення змін, що знижують рівень безпеки інфраструктури, моделювання загроз стає логічним механізмом для проведення аналогічної перевірки.

Оскільки DevOps стирає грань між змінами програмного забезпечення і змінами інфраструктури, з'являються додаткові тригери для моделювання загроз, які історично були б охоплені процесом управління змінами інфраструктури. Двама найбільш поширеними додатковими тригерами в DevOps є:

1. Зміна процесу розгортання ПО. Більшість середовищ DevOps включають інструменти, які автоматично створюють компоненти на основі активності в репозиторії вихідного коду, а потім розгортають ці компоненти у виробничому середовищі. Компрометація будь-якій частині цього процесу може привести до повної компрометації служби, тому будь-яка зміна аутентифікації / авторизації, потоку даних і елементів, задіяних в автоматизації розгортання, вимагає перегляду моделі загроз.

2. Зміни в середовищі виконання служб, включаючи нові порти / протоколи, зміни в сервері додатків або конфігурації ОС, зміни облікового запису і дозволів і т. д. Існує безліч ресурсів, що описують, які події повинні запускати

перевірку в процесі управління змінами. і ці тригери однаково добре відповідають моделювання загроз.

Розробка на каденцію, реліз за вимогою.

Розробка рішень у SAFe здійснюється з фіксованою частотою. Cadence запевняє, що важливі події, такі як планування збільшення програм (PI), демонстраційні демонстрації системи та рішення, а також перевірка та адаптація, відбуваються за регулярним, передбачуваним графіком.

Проекти, орієнтовані на клієнта, прагнуть створити постійний потік вартості для своїх клієнтів. Відмежування випуску вартості з графіка розробки дозволяє споживачам отримувати те, що їм потрібно, коли їм це потрібно. Терміни цих випусків визначаються потребами ринку та споживачів, а також власною мотивацією підприємства забезпечити цінність.

DevOps і конвеєр безперервної доставки дозволяють підприємствам випускати вартість на запит і є важливими елементами гнучкої доставки продуктів. DevOps – це мислення, культура та набір технічних практик. Він забезпечує зв'язок, інтеграцію, автоматизацію та тісну співпрацю, необхідну для планування, розробки, тестування, розгортання, випуску та підтримки рішення.

Конвеєр безперервної доставки представляє робочі процеси, дії та автоматизацію, необхідні для створення нової функції, від ідеї до випуску цінності. Він складається з чотирьох аспектів: безперервна розвідка, безперервна інтеграція, постійне розгортання та випуск на вимогу. Кожна ART будує та підтримує конвеєр, щоб максимально незалежно доставити цінність рішення.

Для організацій, що використовують SAFe, моделювання загроз може бути введено на двох окремих етапах циклу розробки: уточнення епіка (відоме в SAFe як рівень портфолію) і уточнення функцій (програмний рівень).

Портфолію SAFe погоджує стратегію з її реалізацією і організовує розробку рішень навколо потоку цінності через один або кілька потоків створення цінності.

Конфігурація портфолію, яка включає Essential SAFe, є найменшою конфігурацією, яка може бути використана для досягнення ділової спритності, і складається з наступного.

Portfolio SAFe працює на трьох рівнях для менеджерів портфолію, менеджерів продуктів та членів команди. На рівні портфолію менеджери портфолію визначати пріоритети епіка у централізованому відставанні та відстежувати хід епіки на візуальній дошці.

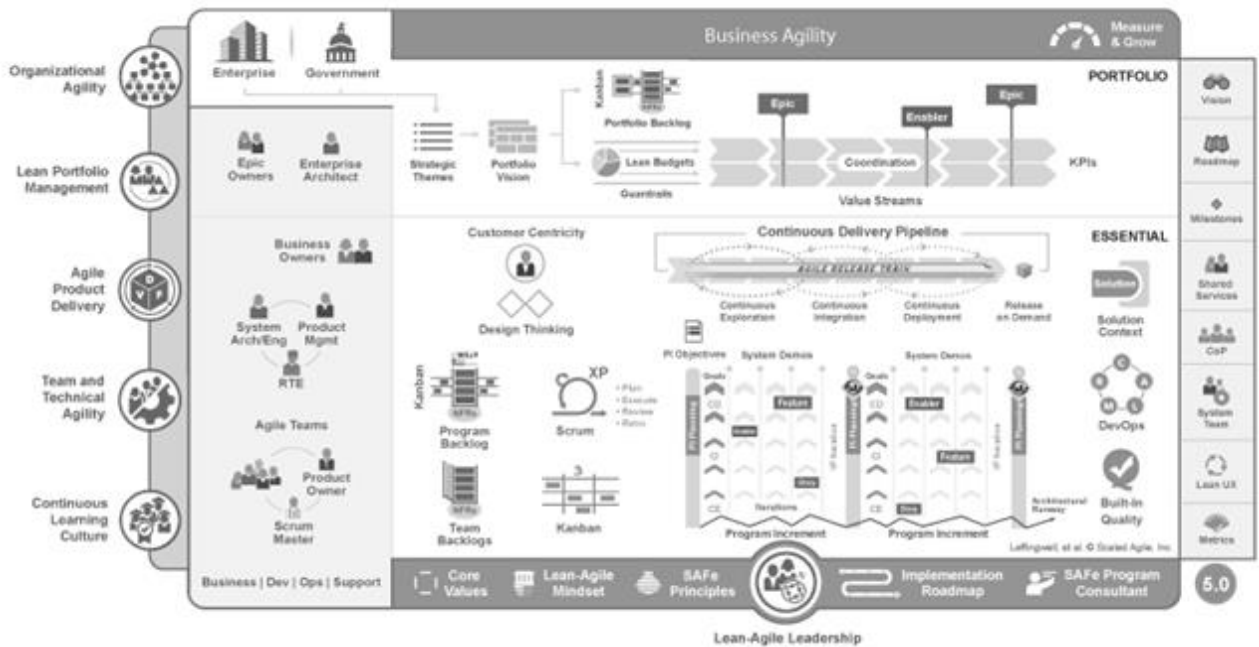


Рис. 3. Портфоліо SAFe

Конфігурація Portfolio SAFe погоджує стратегію підприємства з виконанням портфеля, організовуючи економне і гнучке підприємство навколо потоку цінності через один або кілька потоків створення цінності. Надаючи базові механізми бюджетування і необхідні механізми управління, в тому числі Lean Budget Guardrails, він допомагає гарантувати, що потоки створення цінності спрямовані на створення правильних речей з відповідним рівнем інвестицій в рішення, щоб портфель досяг своїх стратегічних цілей [3]. Основні моменти, ролі, події та артефакти, присутні на рівні портфоліо зображено на рис. 4.

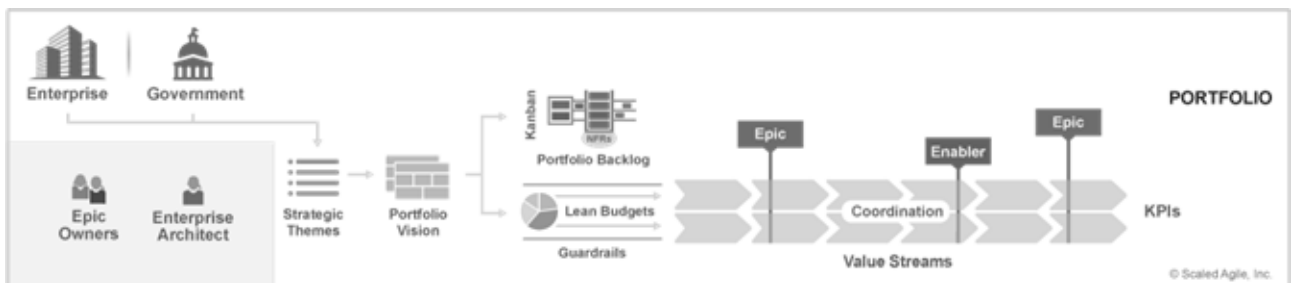


Рис. 4. Рівень портфоліо

Рівень програми містить ролі і дії, необхідні для безперервної доставки рішень через Agile Release Train (ART). На програмному рівні групи розробників, зацікавлені сторони та інші ресурси присвячують себе виконанню важливої постійної місії розробки системи. Метафора ART описує команди, ролі і дії на

рівні програми, які поступово забезпечують безперервний потік цінності. ART – це віртуальні організації, створені для подолання функціональних кордонів, виключення непотрібних передач і кроків, а також для прискорення надання цінності за рахунок впровадження принципів і практик SAFe Lean–Agile.

Хоча це називається програмним рівнем, ART довговічні і, отже, мають більш стійку самоорганізацію, структуру і місію, ніж традиційні програми, які зазвичай мають остаточні дати початку і закінчення, а також тимчасово призначені ресурси.

ART, що працюють на рівні програми, в кінцевому підсумку несуть відповідальність за створення і випуск цінності в потоці з частотою, необхідною підприємству для задоволення ринкового і споживчого попиту. Спосіб мислення і практики на цьому рівні вносять свій внесок в корпоративну компетенцію DevOps і Release on Demand, що робить можливим цей потік цінності.

На програмному рівні ART поставляють одну або кілька систем або, в деяких випадках, все рішення. ART – це довгоживучий, заснований на потоках, що самоорганізується і його характер – ось що рухає силою SAFe. Рівень програми в SAFe - це місце знаходження Agile Release Train (ART). Детальніше зображено на рис. 5.

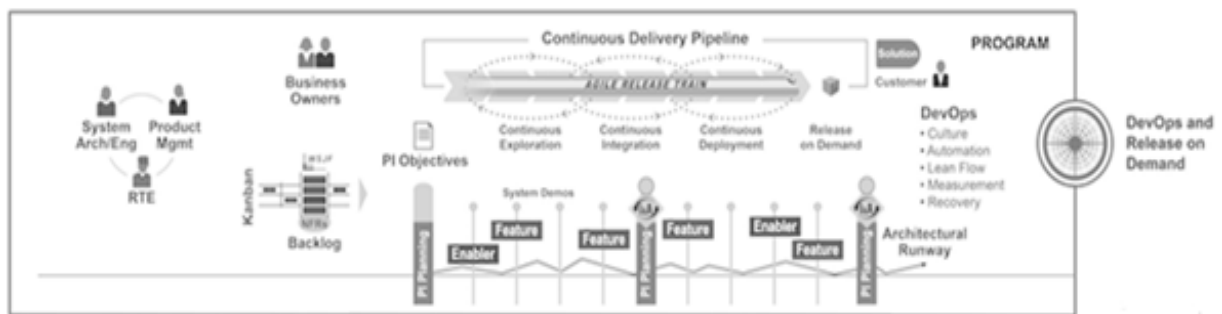


Рис. 5. Рівень програми

Під час уточнення епіка моделювання загроз може бути введено за допомогою методу аналізу ризику для виявлення можливих негативних бізнес-результатів епіків SAFe. Аналіз ризику, швидше за все, буде проводитися щодо існуючих епіків на етапі «аналізу» портфельного канбана [4]. Детальніше моделювання загроз зображено на рис. 6.

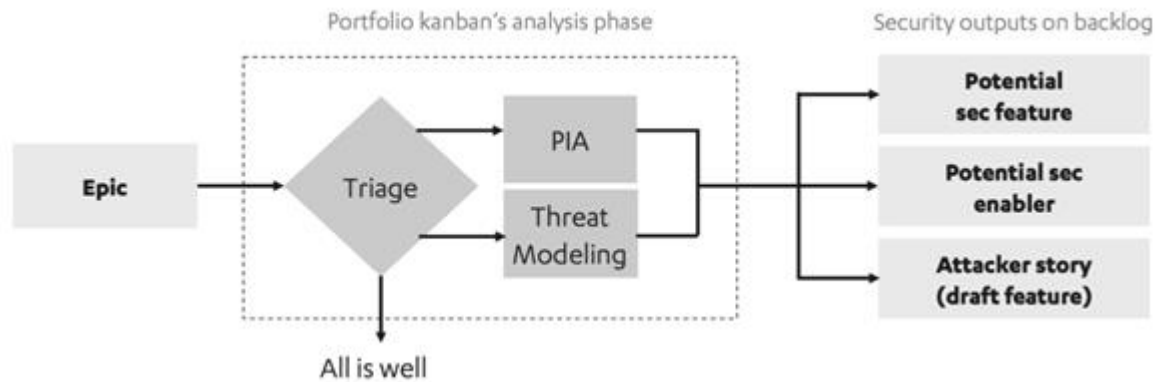


Рис. 6. Моделювання загроз після сортування в епіку, доопрацювання з висновком вимог безпеки

Оцінка впливу на конфіденційність (PIA) або затверджена GDPR, оцінка впливу на захист даних (DPIA) є частиною цього аналізу ризиків. Вони також допомагають продемонструвати відповідність вимогам GDPR «Конфіденційність за замовчуванням і за замовчуванням» (PbDD).

Конфіденційність і безпека мають природну взаємозалежність; якщо з'ясується, що основне економічне обґрунтування саме по собі суперечить правилам конфіденційності, зміна цього правила пізніше при реалізації може вбити проект через брак часу або через витрати на виправлення.

Малоймовірно, що бізнес–модель продукту може бути змінена до повного запобігання ризику. У цих ситуаціях необхідно задокументувати виявлені ризики. Мета полягає в тому, щоб перетворити цей ризик в позитивні (реалізовані) елементи невиконаної роботи, які, як правило, є функціями, які визначають функціональні можливості безпеки та зміни конструкції матеріалів. Вони також можуть виявитися інструментами підтримки, особливо тими, які SAFe називає інструментами підтримки інфраструктури та відповідності.

Документування ризиків, для яких в даний час немає рішень

Особливо в доопрацюванні епіків може бути не відразу зрозуміло, які функції безпеки або засоби підтримки потрібні. Цю прогалину можна заповнити, додавши історії зловмисників або вектор атаки, в список невиконаних робіт.

Історії зловмисників це небажані події або вектор атаки. Історії зловмисників технічно працюють як функції SAFe, які зазвичай перетворюються в функціональні можливості при подальшому аналізі. Поки деталі конструкції не опрацьовані, вони діють як засіб повідомлення про небажані ризики.

Хоча може виникнути спокуса створити епік безпеки, в який можна додати всю роботу по забезпеченню безпеки, цього слід уникати. Це призведе до того, що робота не отримає належного пріоритету, а потреби в безпеці явно не будуть

пов'язані з бізнес–цінністю. Можливо, що вимоги безпеки, поховані під їх власним епіком безпеки, ніколи більше не будуть зустрінуті ніким серед бізнес–аналітиків.

Це перша точка, де робота по забезпеченню безпеки і конфіденційності стає очевидною в процесі розробки. Робота по забезпеченню безпеки відображається як елемент невиконаної роботи і може бути безпосередньо використана як свідчення виконаної роботи, а також незавершеної роботи.

Доопрацювання функцій

При уточненні функцій до них застосовується моделювання загроз. Це фактичні робочі завдання, які повинні бути виконані розробниками. Замість того щоб розглядати аспекти безпеки і конфіденційності як нефункціональні вимоги, мета полягає в тому, щоб зробити їх власними функціями і інструментами. Це змушує керівництво продукту приймати чіткі рішення при розподілі часу розробника на функціональність або безпеку. В результаті безпеку витрат знаходиться під контролем, в той час як прямі докази роботи по забезпеченню безпеки підтверджуються її квитками.

Якщо моделювання загроз вже було виконано шляхом доопрацювання епіка, етап доопрацювання функцій може успадкувати історії зловмисників, які вимагають перетворення в позитивні заходи безпеки: або функції, як засоби реалізації. У будь–якому випадку, підвищена прозорість технічних деталей дозволяє проводити тут подальше моделювання загроз.

Моделювання загроз при уточненні функцій, буде в основному пов'язано з конкретними проблемами технічного проектування і реалізації. Отже, найбільш підходящі люди для моделювання загроз – це самі розробники.

Виконання сортування і моделювання загроз для функцій можна зробити критерієм визначення готовності (DoR). Це означає, що функція не розвиватиметься в змісті інкремента програми, тобто не буде передана групі на реалізацію, до тих пір, поки не буде виконано моделювання загроз. Детальніше моделювання загроз за умови уточнення загроз зображено на рис. 7.

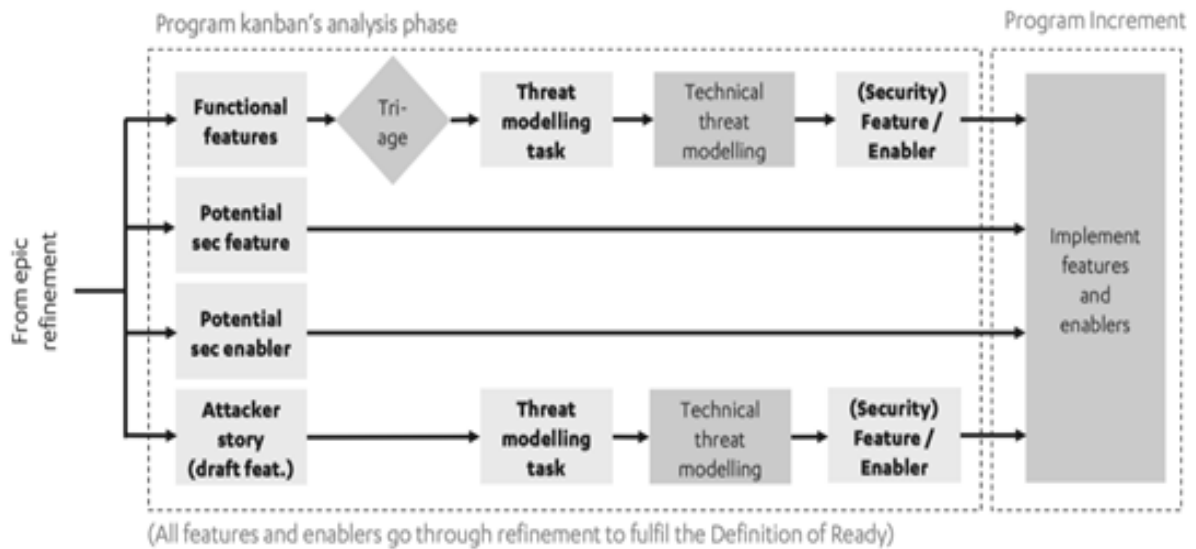


Рис. 7. Моделювання загроз при уточненні ознак загроз

Існує три варіанти запуску моделювання загроз для будь-якого функціонального елемента:

Моделювання загроз з'їдає тимчасові квоти, які розробники використовують для роботи з уточнення. Проблема тут в тому, щоб мати достатньо часу для моделювання загроз для складних функцій за відведений час. Моделювання загроз можна помістити в той же інкремент програми, де буде реалізована функція.

Моделювання загроз потребує додаткової роботи, що може привести до розростання вмісту Program Increment. Моделювання загроз також може бути виконано в попередньому інкременті програми, і будь-які вихідні дані моделювання просто будуть поміщені в чергу.

Наступний етап програми може працювати на основі повного розуміння загроз функції. Однак це може відсунути план реалізації функції на пару місяців вперед. При використанні простого Scrum, де короткі спринти йдуть один за іншим, моделювання загроз в рамках попереднього спринту зазвичай є кращою альтернативою. Програмний інкремент зображено на рис. 8.

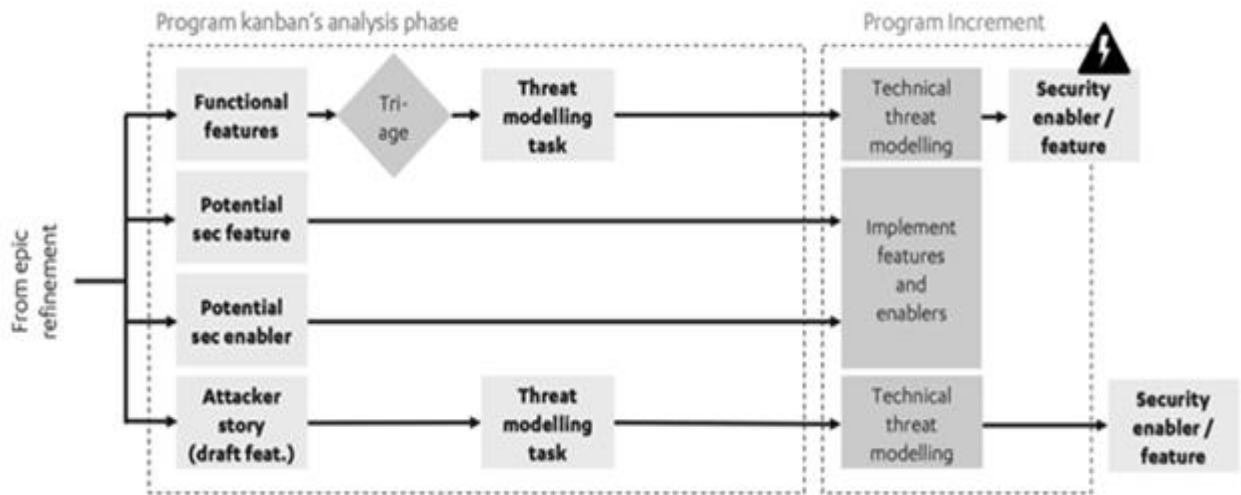


Рис. 8. Program Increment з усіма розробками

Результати моделювання загроз можуть викликати зміни в інкрементного змісті, або результати можуть бути просто поміщені в чергу для подальшої реалізації.

Навіть якщо моделювання загроз виконується під час уточнення функцій, все одно важливо зробити задачу моделювання загроз видимою в невиконаній роботі. Це гарантує наявність свідочств виконаної роботи і їх не можна випускати з уваги.

Вибір правильної методології моделювання загроз

Використовувана методологія моделювання загроз залежить від того, чи є вихідна інформація пропозицією цінності для бізнесу (епік) або функціональною вимогою (функція).

Що стосується епіка, може бути неможливо глибоко зануритися в деталі дизайну, оскільки вони можуть бути навіть не відомі. Замість цього успіху може бути досягнуто за рахунок використання розкадровки зловмисника, яка б показала розгортання векторів атак, які можуть бути використані для надання технічного впливу, що призведе до наслідків або для бізнесу (безпека), або для суб'єкта даних (конфіденційність). Можна полегшити обговорення, в якому зацікавлені сторони зможуть придумати всі необхідні частини історії зловмисника. Розуміння векторів атак є найскладнішим.

Що стосується функцій, важливо уточнити технічні припущення і спрогнозувати будь-які проблеми проектування і реалізації. При моделюванні загроз зазвичай використовується методологія, керована потоком даних,

наприклад STRIDE від Microsoft. У системах, що обробляють персональні дані, які зазвичай використовується TRIM, яке є легкою контрольною точкою.

Висновки

SAFe - це ефективний інструмент для управління масштабними проектами, що складаються з декількох команд та які працюють над великими епіками. Він не стосується деяких складностей, таких як географічний розподіл та дотримання вимог, і SAFe має деякі незвичні терміни, що відрізняє від Scrum. Але терміни відповідають ідеям здорового глузду та їх легко прийняти. Шаблони базової команди Scrum збігаються на рівень програми і дещо на рівень портфолію. Більшість організацій приймають SAFe, починаючи з програмного рівня. Також часто використовують додаткові практики портфолію, щоб доповнити практики SAFe на цьому рівні. Командам, які вже опанували Scrum, легше застосовувати SAFe.

SAFe 5.0 містить важливі оновлення та вдосконалення, що відображають поточні потреби сучасних підприємств та постійно мінливі вимоги ринку. Спрощена, більш впорядкована презентація концентрується на спритності бізнесу та кроках, необхідних для його досягнення, пропонує кращу структурну чіткість та робить акцент на орієнтованості на клієнта як одній із своїх основних цінностей. Автори вказують на важливість таких компетенцій, як безперервне навчання та організаційна спритність, для успішної трансформації та сталого розвитку підприємства, що підтримує безпеку.

Платформа SAFe надає компаніям економічно прийнятний спосіб ефективного масштабування методики agile в організаціях і досягнення кінцевих бізнес-результатів. Але не менш важливі і інструменти, які обирають для зміцнення існуючих методів роботи і реалізації всіх переваг цих методів.

Виділяють 10 принципів SAFe це:

1. Економічний погляд
2. Застосовуйте Системне Мислення
3. Припускайте Мінливість, Зберігайте Опції
4. Інкрементальний поставки з швидкими циклами навчання, вбудованими в процес
5. Віхи визначаються тільки об'єктивною оцінкою діючих систем
6. Визуалізуйте і обмежуйте WIP, зменшуйте розмір порцій робіт і керуйте довжиною черг
7. Застосовуйте каденції, синхронізує крос-доменних планування

8. Задіюйте внутрішню мотивацію співробітників
9. Децентралізоване прийняття рішень
10. Організувати процеси навколо вартості

Працюючи по фреймворку SAFe необхідно завжди дотримуватися «економічної точки зору». Це означає, що лідери повинні чітко сформулювати мету змін в термінах, зрозумілих кожному. Впроваджуючи SAFe, компанія може очікувати побачити вигоди в чотирьох основних сферах: швидкість виходу на ринок (Time To Market), якість, продуктивність, залучення співробітників. Керівники повинні повідомляти про ці передбачувані переваги як частини бачення змін. Крім того, потрібно описати будь-які інші конкретні, відчутні цілі, які сподіваються досягти. Це повинно включати базові метрики, які ілюструють поточний стан, чому поточний стан є нестійким, цільове майбутній стану для тих самих метрик, і стратегію того, як зміна досягне цих цілей. Передбачуване поліпшення цих ключових показників ефективності стане свого роду «паливом», необхідним для подолання інерції поточної системи і вже існуючих процесів.

SAFe не зосереджується на одному підході і заснований на Agile методології в цілому. Він адаптує цінності Agile з середовища невеликої команди в велику організацію і дозволяє кожній команді вибирати, який метод вони хочуть використовувати, будь то Scrum, Kanban, Scrumban або інший. Однак при масштабізації SAFe, незалежно від того, яка практика Agile використовувалася раніше, виникне необхідність у реструктуризації та переосмисленні організації. Це означає, що перехід буде більш дорогим і, швидше за все, займе більше часу, але є відмінним варіантом для команд, які не хочуть бути прив'язаними до Scrum і мають можливість вибирати.

Список використаних джерел

1. State of Agile [Електронний ресурс] – Режим доступу : <https://stateofagile.com/>
2. Leffingwell, Dean. Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and Portfolio //Dean Leffingwell – Addison–Wesley, 2011.
3. Scaled Agile Framework [Електронний ресурс] – Режим доступу : <https://www.scaledagileframework.com/portfolio-safe/>
4. Scaled Agile Framework [Електронний ресурс] – Режим доступу : <https://v46.scaledagileframework.com/program-level/>

ГЛАВА 8

ПІДХОДИ ДО ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вступ і постановка задачі

Забезпечення якості програмного забезпечення (ПЗ) є невід'ємною складовою процесу розроблення і використовуються в ІТ-індустрії фахівцями з тестування і фахівцями щодо забезпечення якості. Як забезпечення якості, так і тестування програмного застосунку покликані гарантувати якість програмного продукту, яка відповідає вимогам замовника. Однак ці два поняття мають принципову відмінність. Тестування проводиться після того, як продукт був створений або для статичного тестування після того, як були визначені і зафіксовані у відповідному документі вимоги до програмного забезпечення. Забезпечення якості передбачає діяльність, яка забезпечує якість продукту на всіх етапах його створення.

Для розуміння відмінностей між цими складовими процесу розроблення програмного забезпечення потрібно дати чітке визначення цих понять, провести співвідношення між їх характеристиками, визначити методи оцінювання якості програмного забезпечення.

Успішне вирішення проблем якості програмного забезпечення можливе лише за умов відповідної організації роботи команд з забезпечення якості і тестувальників, тому в роботі будуть визначені відмінності між обов'язками цих команд, відмінності в плануванні випробувань і документації, а також розроблені рекомендації для поліпшення організації роботи команд.

1. Поняття забезпечення якості, контролю якості та тестування програмного забезпечення

Для з'ясування відмінностей між поняттями тестування і забезпечення якості програмного забезпечення, розглянемо пов'язані з ними поняття "тестування", "контроль якості" та "забезпечення якості".

Тестування програмного забезпечення (Testing) відповідно до ISO/IEC TR 19759: 2005 – це процес дослідження, випробування програмного забезпечення, який має на меті перевірку відповідності між реальною поведінкою програми і її очікуваним поведінкою на кінцевому наборі тестів, обраних певним чином.

Контроль якості (Quality Control, QC) відповідно до ISO 9000 - це частина менеджменту якості, зосереджена на виконанні вимог по відношенню до оцінки кількості дефектів, багів (при їх наявності) в продукті. Контроль якості являє собою набір процесів (дій), спрямованих на оцінку розробленого продукту (проекту документа, системи розвитку тощо) і відповідності вимогам замовника. Виконання цих процесів гарантує перевірку якості поставленого продукту і визначає, наскільки добре він спроектований і виконаний. Мета контролю якості полягає в пошуках дефектів і забезпеченні їх виправлення. Таким чином, тестування є невід'ємною частиною контролю якості (рис. 1).

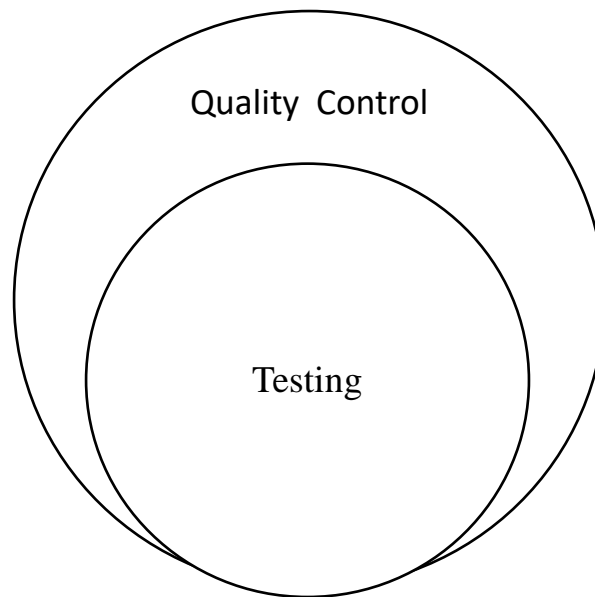


Рис. 1. Співвідношення понять "тестування" і "контроль якості"

Забезпечення якості (Quality Assurance, QA) в стандарті ISO 9000 визначається як частина менеджменту якості, орієнтовану на створення впевненості в тому, що вимоги до усунення багів будуть виконані. Метою забезпечення якості є гарантія того, що продукт буде відповідати вимогам замовника. Забезпечення якості складається з процесів, спрямованих на забезпечення якості розробки продукту на кожному з етапів життєвого циклу. Ці дії, як правило, передують розвитку продукту і тривають, поки процес перебуває в стані розвитку. На забезпечення якості лежить відповідальність за розробку і впровадження процесів і стандартів для поліпшення життєвого циклу розробки,

і забезпечення впевненості в тому, що ці процеси виконуються [1, 2]. Основне призначення забезпечення якості - запобігання дефектів на всіх етапах створення програмного забезпечення і постійне його вдосконалення. У той час, як забезпечення якості є діяльністю, спрямованою на забезпечення розробки якісного програмного забезпечення, контроль якості - діяльністю, яка фіксує і оцінює якість вже створеного продукту. На рис. 2. показано співвідношення між забезпеченням якості, контролем якості та тестуванням. Діяльність із забезпечення якості включає встановлення стандартів і процесів, перевірки якості, а також вибір відповідних інструментів.

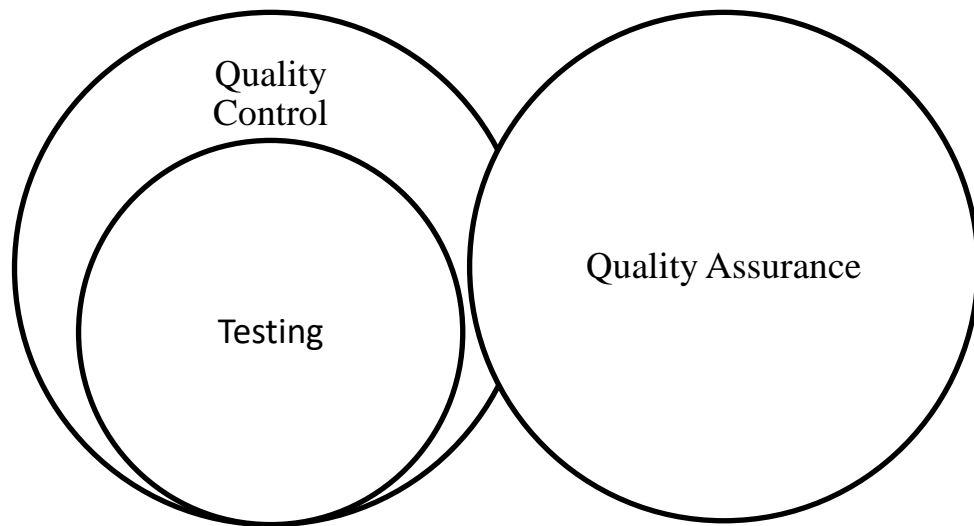


Рис. 2. Співвідношення між QA, QC і Testing

2. Багаторівнева модель якості програмного забезпечення

Якість програмного забезпечення визначається в стандарті ISO 9126 як вся сукупність його характеристик, що належать до можливості задовольняти висловлені або ті, що мають на увазі потреби всіх зацікавлених осіб.

Виділяють наступні аспекти якості ПЗ [3 – 5]:

1. Якість технологічних процесів розробки ПЗ, яка впливає на створення якісного ПЗ;
2. Внутрішня якість ПЗ, пов'язана з його характеристиками, без урахування поведінки програмного продукту;
3. Зовнішня якість, що характеризує ПЗ із точки зору його поведінки;

4. Якість ПЗ при використанні у різних контекстах, тобто якість програмного продукту, яка проявляється при його використанні користувачами в різних конкретних сценаріях роботи.

Для всіх цих аспектів якості створені метрики, які дозволяють їх оцінити

На рис. 3 показаний зв'язок різних аспектів якості ПЗ.

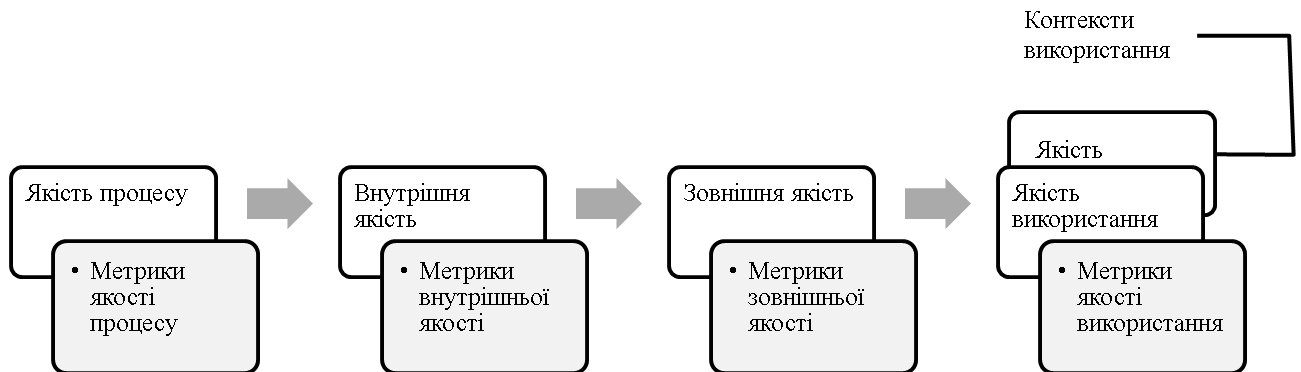


Рис. 3. Зв'язок різних аспектів якості ПЗ згідно ISO 9126

Крім того, в стандарті описана багаторівнева модель якості програмного забезпечення, яку можна використовувати для опису як внутрішньої, так і зовнішньої якості ПЗ (рис. 4). На верхньому рівні моделі виділено 6 основних характеристик якості ПЗ, кожна з яких має свої атрибути:

функціональність: здатність до взаємодії, функціональна придатність, відповідність стандартам і правилам, захищеність, точність;

надійність: завершеність, здатність до відновлення, відповідність стандартам, стійкість до відмов;

зручність використання: зрозумілість, зручність навчання, зручність роботи, привабливість, відповідність стандартам;

продуктивність: часова ефективність, ефективність використання ресурсів, відповідність стандартам;

зручність супроводу: аналіз, зручність внесення змін, стабільність, зручність перевірки, відповідність стандартам;

перенесення: адаптованість, зручність установки, здатність до співіснування, зручність заміни, відповідність стандартам.



Рис. 4. Багаторівнева модель якості ПЗ згідно ISO 9126

Для кожного атрибута визначається набір метрик, які дозволяють його оцінити. Метрики повинні мати такі властивості:

1) надійність; яка пов'язана з випадковою помилкою; метрика вільна від випадкової помилки, якщо випадкові зміни не впливають на результати метрики;

2) повторюваність; повторне використання метрики для того ж продукту і тими ж фахівцями з оцінювання при використанні тієї ж специфікації оцінювання (включаючи навколишнє середовище), того самого типу користувачів і оточення, має привести до тих же результатів з відповідними допусками; відповідні допуски повинні враховувати такі компоненти, як втома і результат накопичених знань;

3) однотипність; застосування метрики для того ж продукту різними фахівцями з оцінювання при використанні тієї ж специфікації оцінювання (включаючи навколишнє середовище), того самого типу користувачів і оточення, має привести до тих же результатів з відповідними допусками;

4) можливість застосування; метрика повинна чітко вказувати умови (наприклад, наявність певних атрибутів), які обмежують її вживання;

5) показовість; це здатність метрики ідентифікувати частини або елементи програми, які повинні бути поліпшені, на підставі порівняння вимірних і очікуваних результатів;

б) коректність; метрика повинна мати наступні властивості:

об'єктивність; результати метрики і її вхідні дані повинні бути засновані на фактах і бути не підвладні почуттям або думкам фахівців з оцінювання або тестування (виключаючи метрики задоволеності або привабливості, за допомогою яких вимірюються почуття і думки користувача);

неупередженість; вимірювання не повинно бути направлено на отримання будь-якого специфічного результату;

адекватність точності; точність визначається при проектуванні метрики і особливо при виборі описів фактів, які використовуються як основа для метрики; розробник метрики повинен описати точність і чутливість метрики;

7) значимість; вимірювання має давати значущі результати, які стосуються поведінки програми або характеристик якості.

Метрика повинна також бути ефективною по відношенню до вартості. Це означає, що більш дорогі метрики повинні забезпечувати кращі результати оцінки.

Розробник метрики повинен довести її обґрунтованість. Метрика повинна задовольняти хоча б одному з наступних критеріїв обґрунтованості метрики:

1) кореляція; зміна в значеннях параметрів якості (оперативно визначених за результатами вимірювання основних метрик), обумовлена зміною в значеннях метрики, має визначатися лінійною залежністю;

2) трасування; якщо метрика M безпосередньо пов'язана з величиною характеристики якості Q , то зміна величини $Q(T_1)$, наявної в момент часу T_1 , до величини $Q(T_2)$, отриманої в момент часу T_2 , має супроводжуватися зміною значення метрики від $M(T_1)$ до $M(T_2)$ в тому ж напрямку (наприклад, якщо збільшується Q , то M теж збільшується);

3) несуперечливість; якщо значення характеристик якості (оперативно отриманих за результатами вимірювання основних метрик) Q_1, Q_2, \dots, Q_n , пов'язані з продуктами або процесами $1, 2, \dots, n$, визначаються співвідношенням $Q_1 > Q_2 > \dots > Q_n$, то відповідні значення метрики повинні задовольняти співвідношенню $M_1 > M_2 > \dots > M_n$;

4) передбачуваність; якщо метрика використовується в момент часу T_1 для прогнозу значення (оперативно отриманого за результатами вимірювання основних метрик) характеристики якості Q в момент часу T_2 , то помилка прогнозування

$$(Q_p(T_2) - Q_f(T_2)) / Q_f(T_2),$$

де $Q_p(T_2)$ – прогнозне значення характеристики якості в момент T_2 ,
 $Q_f(T_2)$ – фактичне значення характеристики якості в момент T_2 ,

повинна потрапляти в допустимий діапазон помилок прогнозування;

5) селективність; метрика повинна бути здатною розрізняти високу і низьку якість програмного засобу.

3. Методи визначення показників якості програмного забезпечення

Методи визначення показників якості програмного забезпечення можна класифікувати виходячи з ряду факторів, які представлені в табл. 1.

Таблиця 1

Класифікація методів визначення показників якості програмного забезпечення

Ознака	Метод
Спосіб отримання інформації	Вимірювальний
	Реєстраційний
	Органолептичний
	Розрахунковий
Джерело отримання інформації	Експертний
	Соціологічний
	Традиційний

Вимірювальний метод визначення показників якості програмного забезпечення – це метод отримання інформації про його властивості та характеристики шляхом вимірювань за допомогою інструментальних засобів (наприклад, кількість операторів в програмі, кількість виконаних операторів, кількість операндів, час виконання програми за певних наборах вхідних даних тощо).

Реєстраційний метод визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики ПС під час його випробування або функціонування, коли реєструються певні події, наприклад, кількість збоїв, відмов.

Органолептичний метод визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики програмного забезпечення, заснований на сприйнятті таких органів почуттів людини, як зір і слух, наприклад, зручність використання.

Розрахунковий метод визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики програмного забезпечення, заснований на використанні емпіричних і теоретичних залежностей на ранніх етапах його розроблення, накопичених при випробуваннях, експлуатації та супроводі програмного забезпечення статистичних даних,. Таким чином може визначатися, наприклад, точність обчислень.

Експертний метод визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики програмного забезпечення на підставі думок групи експертів-фахівців, компетентних у вирішенні даного завдання на основі їх досвіду і інтуїції. Експертний метод застосовується, коли виконати оцінку показників якості за допомогою інших методів неможливо або потребує значних трудовитрат. За допомогою даного методу рекомендується визначати, наприклад, показники зрозумілості і опанування програмного забезпечення.

Соціологічний метод визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики програмного забезпечення на підставі обробки спеціальних анкет-опитувальників. Цим методом, наприклад, можуть визначатися окремі показники зручності використання.

Традиційний метод визначення показників якості програмного забезпечення - це метод отримання інформації про властивості та характеристики програмного забезпечення на підставі безпосереднього спостереження за їх функціонуванням в процесі роботи. Цим методом, наприклад, можуть визначатися деякі з показників функціональності і зручності використання.

4. Оцінювання якості програмного забезпечення

Існують різні методи оцінювання якості програмних продуктів [3 – 5]. Дані методи дозволяють отримати підсумкову інтегральну величину якості програмного забезпечення в цілому або окремих його характеристик, виражену в певних кількісних показниках. Найбільш поширеними є методи інтегральної оцінки якості програмного забезпечення, заснований на витратах та оцінки якості на основі ієрархічної моделі

Метод інтегральної оцінки якості програмного забезпечення, заснований на витратах відноситься до групи розрахункових методів. Відповідно до даного методу формулюється кількісний критерій якості програмного забезпечення T , орієнтований на його життєвий цикл (ЖЦ).

Витрати на розроблення, експлуатацію та супровід програмного забезпечення включають:

R – одноразові витрати на розроблення програмного забезпечення;

V – одноразові витрати на впровадження програмного забезпечення.

E – повторювані витрати S на експлуатацію програмного забезпечення за період часу експлуатації t_e протягом життєвого циклу T :

$$E = (T / t_e) * S,$$

C – повторювані через випадкові проміжки часу витрати на супровід, які складають в середньому n -ту частину від витрат R і m -ту частину від витрат V і здійснюються протягом життєвого циклу T в середньому через періоди часу t_c :

$$C = (n * R + m * V) * T / t_c,$$

B – випадкові втрати через недостовірність або несвоєчасність результату:

$$B = S_e * T / t_e,$$

Тут S_e – середня сума втрат, яка припадає на одноразову експлуатацію програмного забезпечення протягом його ЖЦ.

Таким чином загальна сума витрат Z в процесі життєвого циклу програмного забезпечення на програмного забезпечення, буде визначатися таким чином:

$$Z = R + V + (S_e + S) * T / t_e + (n * R + m * V) * T / t_c.$$

Як критерій якості пропонується використовувати мінімізацію сумарних витрат на розроблення, експлуатацію та супровід програмного забезпечення. Критерієм якості програмного забезпечення є мінімізація загальної суми витрат Z :

$$Z \rightarrow \min.$$

Основним недоліком даного методу є те, що фактичні значення витрат, які входять в формулу, можна визначити після завершення розробки програмного застосунку, і отже його не можна використовувати як інструмент в процесі його розроблення для досягнення заданого рівня якості.

Відповідно до **методу оцінки якості на основі ієрархічної моделі** вибір номенклатури показників якості для конкретного програмного застосунку здійснюється з урахуванням його призначення і вимог до сфери застосування в залежності від приналежності програмного забезпечення до того чи іншого підкласу, який визначається класифікатором програмних продуктів:

- операційні системи і засоби їх розширення;
- програмні засоби управління базами даних;
- інструментально-технологічні засоби програмування;
- програмні продукти для інтерфейсу і управління комунікаціями;
- програмні продукти організації обчислювального процесу (планування, контролю);
- сервісні програми;
- програмні продукти обслуговування обчислювальної техніки;
- прикладні програми для наукових досліджень;
- прикладні програми для проектування;
- прикладні програми для управління технічними пристроями і технологічними процесами;

прикладні програми для вирішення економічних завдань;
інші програмного продукту.

Оцінювання якості ПЗ полягає у виборі номенклатури показників, їх оцінюванні і зіставленні з базовими значеннями. Основу описуваного методу оцінювання якості становить чотирирівнева ієрархічна модель якості, яка включає:

- 1 рівень - характеристики якості ;
- 2 рівень - атрибути якості;
- 3 рівень - метрики;
- 4 рівень – оціночні показники (атрибути ПЗ).

Для кожної з обраних характеристик якості складається чотирирівнева ієрархічна модель, яка відображає взаємозв'язок характеристик, атрибутів, метрик і показників. Вид даної моделі залежить від фази ЖЦ.

Для практичного застосування моделі використовуються таблиці, які створюються для кожної характеристики. Так для оцінювання характеристики "Надійність" можна використовувати показники, які є в табл. 2.

Таблиця 2

Показники оцінки характеристики "Надійність"

Показник	Метод оцінювання	Форма оцінювання
Вплив помилок в вхідних даних на стійкість функціонування	Експертний	Шкала 0-1
Можливість оброблення помилок	Експертний	Шкала 0-1
Імовірність безвідмовної роботи, P	Реєстраційний, розрахунковий	$P=1 - q/n$, де n – кількість випробувань, q- кількість число зареєстрованих відмов

5. Вдосконалення роботи QA команд і команд тестувальників

Потреба в команді з забезпечення якості ПЗ зростає з розміром організації та рівнем її політики в області якості. Там, де потрібно така команда, дуже важливо, щоб функція QA залишалася незалежною від менеджменту проектом і операційної діяльності. Але зв'язки між QA команди з командою проекту повинні надавати їм потужну підтримку.

Деякі організації мають функцію QA, вбудовану в офіс управління проектами. Така модель також відповідає критеріям незалежності. Однак, за такої організації треба бути впевненим, що QA команда складається з кваліфікованих аналітиків з забезпечення якості.

З огляду на відмінності розглянутих понять тестування програмного забезпечення, контролю якості і забезпечення якості, є також відмінності між обов'язками QA команд і команд тестувальників.

Обов'язки команд. В обов'язки команд тестувальників входить:
планування тестування,
написання тестових сценаріїв і тест кейсів, перевірку тестів,
виконання тестів,
аналіз результатів тестування,
створення і аналіз звітності про результати тестування для різних рівнів випробувань.

В рамках своєї ролі з контролю якості тестувальники можуть висувати вимоги щодо:

перевірки зразків документів проекту,
діяльності з управління конфігураціями програмного забезпечення, дизайну, коду тощо.

В той же час QA команди виконують такі функції:
формування організаційної політики щодо якості, стандартів і процесів розроблення;

надання допомоги з підготовкою в області забезпечення якості та планів забезпечення якості проекту;

перевірка відповідності між процесами проекту і планами якості;
проведення регулярних перевірок проектних продуктів і процесів, а також регулярне представлення результатів аналізу оцінок якості для вищого керівництва;

загострення ситуації з відхиленням від керівних принципів або стандартів. В рамках своєї ролі щодо забезпечення якості QA-команда виконує контроль:

- незалежних оглядів;
- наявності процедур управління змінами проектів;
- наявності процедур управління конфігураціями проектів;
- наявності ретроспективи планування і виконання процесів життєвого циклу розробки;
- забезпечення якості на основі розвитку системи життєвого циклу;
- проведення безперервного поліпшення в процесі контролю якості і втілення рекомендацій, заснованих на отриманому раніше досвіді.

Виконання обов'язків QA команди ще означає їх розвиток командою, а тільки забезпечення їх реалізації.

Планування випробувань і документація. При плануванні випробувань тестувальники готують тестові стратегії і плани, засновані на базових тестових документах, таких як вимоги до програмного продукту і проектні рішення. Ці документи планування тестування є основою випробування процесів на різних запланованих випробувальних рівнях. Для кожного рівня випробувань складаються тести, набори вхідних даних і очікуваних результатів, докладні графіки випробувань, екологічні вимоги, управління дефектами, управління тестуванням і ведення звітності. На відміну від цього, документація з забезпечення якості програмного продукту або плани якості включають більш широкий набір дій протягом усіх етапів розроблення. Це відіграє свою роль при розробленні методології управління проектами.

Типовий проект плану якості включає в себе очікування клієнтів, критерії приймання, плановий контроль якості і аудиту процесів, плани управління конфігураціями і процедури управління змінами. При складанні планів щодо якості виходять з політики самої організації в цій сфері, стандартів або керівних принципів, які формують основи забезпечення якості. В ході створення проекту моніторинг плану забезпечення якості проекту здійснюється безперервно і на його основі оновлюються заплановані показники якості діяльності. Існують різні точки перетину між управлінням ризиками і якістю, і тому реєстр ризиків може зробити вагомий внесок в підготовку планів якості.

Рекомендації щодо поліпшення роботи команд. Перед QA командами встає цілий ряд проблем щодо організації їх роботи, а саме:

незалежність. Для успіху QA команди повинні бути незалежними від команди проекту. Це забезпечує QA команді можливість проведення об'єктивної оцінки проектів. В невеликих організаціях тестувальники і QA фахівці можуть перебувати в одній команді. Однак з'являється можливість створення конфлікту інтересів при моніторингу діяльності тестування. Рішенням проблеми в залежності від політики самої організації в сфері якості може бути створення окремої команди для формування звітності;

відносини всередині команди. Якщо аналітики забезпечення якості занадто процесно-орієнтовані і наполягають на процесах або документації, які можуть не мати особливого значення для проекту, це може погіршувати відносини з керівниками проекту. QA команді буде набагато легше працювати з проектними групами, якщо вони будуть працювати за принципом врахування цілей проекту. Крім того надання допомоги і сприяння з боку проектних команд формує основу для підтримки хороших відносин, що є важливим аспектом успішного тестування;

залучення потрібних фахівців. Для успішної діяльності QA команд провідну роль відіграє якісна кадрова політика. Люди з досвідом в області розробки ЖЦ програмного забезпечення є хорошими кандидатами для QA команди. Деякі знання в рамках стандартів ISO і принципів СММІ є додатковими корисними знаннями;

контрольні списки вимог. Стандартні контрольні списки є корисним механізмом для проведення аудиту проектів, особливо якщо вони розроблені відповідно до фазами ЖЦ. Для плідної співпраці з менеджерами проекту важливим є забезпечення участі в проекті зацікавлених сторін. Це дає можливість отримати від них зворотній зв'язок у відповідь на пропозиції щодо внесення змін до списків;

зв'язок і звітність. Хоча регулярна звітність для вищого керівництва дуже важлива, розроблення правильних шаблонів і показників для забезпечення топ-менеджерів необхідною інформацією гарантує, що цими звітами приділяється належна увага. Це найкраще досягається шляхом проведення зустрічей з відповідними представниками вищого керівництва, надання їм звітів і отримання від них відгуків і коментарів. Крім того, QA командам необхідно постійно

отримувати схвалення на внесення змін до процесів контролю якості та стандартів і забезпечувати ефективну взаємодію з зацікавленими сторонами;

постійне вдосконалення. Врахування попереднього досвіду забезпечує QA-команду основою для оцінювання процесів і рекомендаціями щодо забезпечення якості, включаючи постійні поліпшення. QA-команда повинна проявляти гнучкість, підтримувані хороші відносини із зацікавленими сторонами при внесенні поліпшення в управлінську звітність. Безперервні удосконалення можуть також вимагати внесення змін до методології розроблення системи, тому QA командам рекомендується зберігати методологію розробки IT-відділу.

Висновки

Зіставлення відмінностей між поняттями "тестування", "контроль якості" та "забезпечення якості" показало, що тестування є частиною контролю якості, а контроль якості співпадає з забезпеченням якості в площині перевірки якості.

Оцінювання якості програмного забезпечення повинно враховувати міжнародні стандарти в цій галузі, які визначають різні аспекти якості, такі як якість процесу, внутрішня якість, зовнішня якість та якість використання. Для визначення якості використовується багаторівнева модель, яка включає такі характеристики, функціональність, надійність, зручність використання, продуктивність, зручність супроводу, перенесення. Визначення якості програмного забезпечення здійснюється з використанням методів інтегральної оцінки якості та окремих показників якості.

Для оцінювання відмінностей між процесами з контролю якості та забезпечення якості був проведений аналіз їх обов'язків відповідних команд, планування їх роботи та документації, що дало можливість порівняти виконувані функції й умови роботи. Так QC підтверджує, що конкретні результати відповідають стандартам і специфікаціям, а QA є ширшою функцією, яка охоплює планування і контроль на протязі всього життєвого циклу розроблення. Тестування ж є невід'ємною частиною контролю якості. Для того, щоб в IT-компанії процеси управління якістю були ефективними команди QA і QC повинні співпрацювати.

Успішна QA команда може додати значну цінність для організації, а саме:
підвищення якості програмних продуктів;
послідовність в процесах постачання програмних продуктів;

вдосконалення організації процесів;
зменшення загальних витрат на поставку;
використання застосунків для документації щодо підтримки продукту.

В той же час потрібно враховувати, що для QA-команди потрібні додаткові витрати в штатному розкладі для аналітиків забезпечення якості ПЗ та ускладнення процесів, які можуть викликати розчарування в деяких співробітників

Література

1. Ушакова І.О. Проектування інформаційних систем : практикум / І. О. Ушакова. – Х. : ХНЕУ ім. С. Кузнеця, 2015. – 236 с. (Укр. мов.)
2. Ушакова І.О. Методика управління вимогами в гнучких методологіях / І.О. Ушакова. // Збірник наукових праць ХНУПС. – 2018.– Вип. 2(56). – С. 93 – 98.
3. Foidl H., Felderer M. Integrating software quality models into risk-based testing // Software Quality Journal. –V 26. -2018. – P. 809 – 847.
4. Jacob P.M., Mani P. A framework for evaluating performance of software testing tools // International Journal of Scientific and Technology Research. –V. 9. – Issue 2. – 2020. P. 2175–2180.
5. Pietrantuono R. On the testing resource allocation problem: Research trends and perspectives //Journal of Systems and Software. –V. 161. – 2020. – 42 p..

ГЛАВА 9

МОДЕЛЮВАННЯ ВІДТВОРЕННЯ ПЛОСКИХ КРИВИХ ЛАМАНИМИ ЗА АСИМПТОТИЧНО - ОПТИМАЛЬНИМ АЛГОРИТМОМ ІНТЕРПОЛЯЦІЇ

Вступ і постановка задачі. Відтворення безперервних геометричних об'єктів, таких як криві лінії та поверхні, на комп'ютерах, станках чи принтерах потребує дискретизації, тобто представлення у вигляді кінцевої множини точок (вузлів). Ця множина надалі перетворюється у пікселі, або з'єднується простішими лініями (найчастіше, прямими) – [1]. Звичайно, при цьому виникає проблема оптимізації кількості вузлів та їх розташування уздовж кривої чи поверхні, що відтворюється. Таким чином, розробка ефективних алгоритмів відтворення кривих ламаними з урахуванням точності апроксимації та оптимізації кількості вузлів ланок становить актуальну наукову проблему.

Метою цієї роботи є вдосконалення методів використовуваних алгоритмом асимптотично оптимального алгоритму кусково - лінійної інтерполяції плоских параметричних кривих на основі дослідження впливу параметрів при моделюванні похибок апроксимації реальних кривих ліній.

Розробці алгоритмів шматко-лінійного відтворення кривих присвячене багаточисельна множина досліджень. Ці дослідження можливо поділити на роботи з інтерполяції кривих, коли вузли ламаної знаходяться на первісній кривій, та дослідження з апроксимації кривих, під якими будемо розуміти розташування цих вузлів і самої ламаної в межах границь «допуску», що визначається точністю відтворення.

Загальні алгоритми асимптотичної оптимальної інтерполяції та апроксимації кривих технологічними лініями (в т. ч. ламаними) було розглянуто в роботі [2]. Особливістю такого підходу є те, що він дає ламану з найменшою кількістю ланок. Проаналізувавши ці дослідження можливо виділити ряд невирішених в цих дослідженнях питань, а, саме: обґрунтування кількості точок попередньої дискретизації кривої, обрання методу чисельного інтегрування при обчисленні значень функції регулятора вузлів та методу інтерполяції значень цієї функції, обґрунтування обчислювальних формул дискретних аналогів похідних при обчисленні значень інтегральної функції та ін.

Основна частина

Асимптотичні оптимальні модель інтерполяції кривих ламаними в хаусдорфовій метриці.

Теоретичною базою досліджень, викладених в роботі, є монографія [2], де в формі теорем було доведено асимптотичну оптимальність моделей для загальних випадків подання кривих.

Далі будемо вважати, що криві, які відтворюються, мають представлення в параметричній векторній формі у вигляді єдиної дуги:

$$\mathbf{p} = \mathbf{p}(t) (x = x(t), y = y(t)), \quad (1)$$

або складаються з декількох дуг зі спільною параметризацією, наприклад, мають форму сплайна:

$$S(t) = \sum_{j=1}^k a_j B_j(t). \quad (2)$$

Асимптотично оптимальною в хаусдорфовій метриці, згідно доведеному в [2], вважається інтерполяція плоскої кривої ламаною, що базується на виборі вузлів на основі репараметризації кривої за наступною схемою:

1) Визначається кількість ланок інтерполяції за формулою:

$$m = \left[\frac{1}{\sqrt{8\varepsilon}} \int_0^T \Phi(t) dt \right] + 1, \quad (3)$$

де: $[a]$ – ціла частина числа a ;

ε – похибка інтерполяції, якій відповідає допустима хаусдорфова відстань між кривою та ламаною;

t – параметр кривої з областю зміни $[0, T]$;

$\Phi(t)$ – функцію розподілу, що має такий вигляд:

$$\Phi(t) = \sqrt{\frac{|x''(t)y'(t) - x'(t)y''(t)|}{\sqrt{x'(t)^2 + y'(t)^2}}}. \quad (4)$$

2) Значення параметру кривої - t_i , яким відповідають вузли інтерполяції, визначається з умов:

- при $\Phi(t) \neq 0, t \in [0, T]$

$$\int_0^{t_j} \Phi(t) dt = \frac{j}{m} \int_0^T \Phi(t) dt, (j = 0, 1, \dots, m) \quad (5)$$

- при наявності нульових значень функції $\Phi(t)$ до неї додається величина $m^{-\alpha}$ і рівняння розподілу перетворюється на

$$\int_0^{t_j} (\Phi(t) + m^{-\alpha}) dt = \frac{j}{m} \int_0^T (\Phi(t) + m^{-\alpha}) dt, (j = 0, 1, \dots, m), \quad (6)$$

де α – стала, що згідно [2] може приймати значення в межах від 0 до 2/3.

Оскільки, права частина рівняння (5) або (6) являє собою монотонну зростаючу дискретну послідовність значень, то у граничному випадку при цю послідовність можливо розглядати як змінну s , а вираз (5) або (6) - як рівняння репараметризації. Якщо пожити, наприклад

$$\Psi(t) = \int_0^t (\Phi(u) + m^{-\alpha}) du, \quad (7)$$

залежність між параметрами матиме такий вигляд

$$t = \Psi^{-1}(s), \quad (8)$$

де Ψ^{-1} функція, зворотна до Ψ .

Природньо, такий підхід можливий тільки, коли Ψ монотонно зростає. До того ж найчастіше інтеграл правої частини (7) не має кінцевого виразу. Це призводить до необхідності дискретизації значень інтегральної функції на основі чисельного інтегрування правої частини (7) та подальшої інтерполяції цих значень.

Отримання вузлів асимптотично оптимального розбиття кривої

Для інтерполяції значень інтегральної функції розподілу може бути запропонований наступний алгоритм.

Поперед усього, визначим кількість інтервалів первісної дискретизації інтеграла правої частини рівнянь (5) або (6), яку позначимо через N .

Далі визначимо шаг рівномірної сітки дискретизації $h = \frac{T}{N}$ та отримаємо набір точок кривої

$$P_i = \mathbf{p} \left(\frac{i}{N} \right), \quad (i = 0, 1, \dots, N), \quad (12)$$

що відповідають дискретному ряду значень параметру кривої $\Delta_N^t \left\{ t_i = 0, \frac{1}{N}, \frac{2}{N}, \dots, T \right\}$.

Для отримання значень послідовності вузлів $\Delta_m^t = \{t_{j,m}^*\} (j = 0, 1, \dots, m)$ асимптотично - оптимального розбиття кривої одним із відомих чисельних методів інтегрування одержимо послідовність значень $s_i = \Psi(t_i)$, $t_i \in \Delta_N^t$, $(i = 0, 1, \dots, N)$ функції (7) у вузлах первісної рівномірної сітки по t - Δ_N . І, далі, інтерполюючи ці значення між вузлами, знайдемо значення зворотної функції $t_j = \Psi^{-1}(s_j)$ при рівномірному вже, згідно правій частині (5) або (6), розподілу значень s_j від 0 до $s_N = \Psi(t_N)$.

При цьому кількість ланок m асимптотично - оптимального розбиття, згідно (3), набуде значення

$$m = \left[\frac{s_N}{\sqrt{8\varepsilon}} \right] + 1. \quad (13)$$

Значення вузлів рівномірної сітки по s - $\Delta_m^s \{s_j, j = 0, 1, \dots, m\}$ обчислимо за виразом

$$s_j = \frac{j s_N}{m}. \quad (14)$$

Визначимо значення параметру кривої - $t_{j,m}^*$, які утворюють асимптотично оптимальну послідовність вузлів. Найпростішим таким підходом є лінійна

інтерполяції функції $s=\Psi(t)$ за її значеннями на первісній рівномірній сітці Δ_N^t . Знаходячи перетин ліній сітки $s=s_j$ із побудованим на сітці Δ_N графіком $\Psi(t)$, отримаємо відповідні значення послідовності Δ_m^t з наступного виразу оберненої залежності

$$t_j^* = t_i + \frac{T(s_j - s_i)}{N(s_{i+1} - s_i)}, \text{ при } s_i \leq s_j \leq s_{i+1} \quad (t_i \in \Delta_N^t). \quad (15)$$

Оцінювання результатів відтворення кривих

Перед тим, як перейти до обговорення результатів побудови апроксимуючих ламаних для конкретних кривих ліній за різними варіантами алгоритму, розглянемо питання оцінки забезпечення необхідної точності наближення ламаної до кривої.

Для можливості оцінювання результатів відтворення та порівняння їх із максимальною допустимою відстанню ε між кривою на ламаною, що її апроксимує, може бути застосований метод еквідистант до кривої - [2]. Сутність цього методу полягає в побудові разом із ламаною внутрішньої та зовнішньої еквідистант до відтворюваної кривої за їх загальними параметричними рівняннями.

Метод еквідистант надає можливості візуального контролю за поведінкою ламаною відносно меж ε – допустимого коридору вихідної кривої. Однак, суттєвими недоліками такого підходу є відсутність кількісної оцінки якості наближення, складність такого контролю зі зменшенням похибки i , як слідство, із зростанням кількості ланок апроксимуючої ламаної.

Щоб усунути вказані недоліки, розглянемо можливості оцінювання на основі кількісної характеристики якості відтворення. Природньо за таку характеристику прийняти максимальну відстань між ланкою ламаної l_j та точками відповідної їй дуги кривої - $\Gamma_{t_{j-1}^*}^{t_j^*}$, яку апроксимує ланка, та порівнювати її з заданою похибкою наближення. Отже, склавши послідовність таких відстаней для усієї ламаної $\{d_{j,m}^{\max}\} (j=1, \dots, m)$, де $d_j^{\max} = \max\{d(\Gamma_{t_{j-1}^*}^{t_j^*}, l_j)\}$ можливо представити похибки відтворення кривої для всіх ланок ламаної графічно у вигляді точкової або стовпчикової діаграми, де по осі абсцис відкладати номери ланок ламаної, а по осі ординат – відповідну відстань від кривої до ламаної.

У випадку інтерполяційної ламаної початкова та кінцеві точки її довільної ланки збігаються з точками вхідної кривої при $t = t_{j-1}^*$ та $t = t_j^*$. Таким чином, дуга кривої, яку інтерполуює поточна ланка, визначається інтервалом зміни параметру кривої - (t_{j-1}^*, t_j^*) .

Відстань між прямою j -ї ланки інтерполяційної ламаної та точкою $M (x_M(t), y_M(t))$ при $t \in (t_{j-1}^*, t_j^*)$ кривої, визначається рівністю

$$d_j(t) = \left| \frac{A_j x_M + B_j y_M + C_j}{\sqrt{A_j^2 + B_j^2}} \right|, \quad (22)$$

де коефіцієнти $A_j = y_{j-1} - y_j$, $B_j = x_j - x_{j-1}$, $C_j = y_j x_{j-1} - x_j y_{j-1}$ залежать від координат кінцевих точок ланки.

Щоб отримати локальні екстремуми функції (22), диференціюємо її праву частину по t та дорівнявши її нулеві, отримаємо

$$A_j x'(t) + B_j y'(t) = 0.$$

Отримавши корені цього рівняння, що лежать в досліджуваному інтервалі значень, обираємо з них максимальне значення - d_j^{\max} .

У випадку, якщо вхідна крива є складеної із декількох елементарних дуг, вузлові точки інтерполяційної ламаної можуть належати до різних складових дуг. Це, значить, що відстань між кривою та ламаною потрібно визначати серед усіх дуг кривої, що знаходяться в межах вузлів інтерполяції такої ланки. Наприклад, для B - сплайну з відкритим вузловим вектором \mathbf{T} ($t_i \in [0, T]$) знаходимо наступні значення вузлового вектору $t_i \geq t_j$ (при $t_{i-1} < t_j$) та $t_{i-k} \geq t_{j-1}$ (при $t_{i-k-1} < t_{j-1}$), які відповідають j -ї ланці ламаної. І, далі, для кожної дуги сплайну в межах значень вузлового вектору $[t_{i-k-1}, t_i]$ знаходимо відстані за наведеним вище алгоритмом - $\{d_k^{\max}\}$. Після чого обираємо максимальне з них - $d_j^{\max} = \max\{d_k^{\max}\}$.

Результати моделювання кривих без точок перегину.

Моделювання відтворення ламаними за асимптотично - оптимальним алгоритмом інтерполяції конкретних кривих ставило за мету дослідити вплив параметрів алгоритму на якість результатів апроксимації.

Перед початком розгляду результатів моделювання зупинимось на питанні їх оцінювання в залежності від заданої (яку необхідно забезпечити) точності наближення. Розподіл значень похибок або відстаней від кривої до ланок ламаної звісно залежить та порівнюється із значенням допустимої похибки наближення. Щоб спростити це порівняння та мати змогу аналізувати результати отримані при різних значеннях точності, введемо поняття відносної похибки наближення ланки ламаної до кривої, за яку візьмемо відношення максимальної відстані від ланки ламаної до дуги кривої, яку вона апроксимує, до значення заданої допустимої похибки наближення. Це означає, що значення відносної похибки в межах від 0 до 1 задовольняють заданій точності апроксимації, а значення більші за одиницю виходять за її межі.

Обговорення результатів моделювання почнімо з дослідження чинників, що пов'язані з інтерполяцією інтегральної функції та її впливом на точність відтворення кривої. До цих чинників можна віднести: кількість вузлів первісної сітки дискретизації інтерпольованої інтегральної функції - N , похибка

обчислення значень функції Ψ у вузлах цієї сітки - δ^Ψ , метод інтерполяції. В рамках даної роботи обмежимося вивченням впливу на результати відтворення кількісної характеристики N при визначеній рівнянням (15) лінійної інтерполяції функції – регулятора вузлів.

В роботі [2] при визначенні значення N використовувалась наступна залежність

$$N = \left\lceil \frac{1}{\varepsilon} \right\rceil + 1, \quad (23)$$

яка враховує тільки допустиму похибку відтворення та не залежить від властивостей самої кривої. До того ж цей вираз не залежить від того, яким чином відбувається інтерполяція значень інтегральної функції на проміжках між вузлами її дискретизації – прямими лініями, як в даній роботі, або іншими методами. З метою подальшого порівняння наведемо ряд значень похибок апроксимації та відповідний їм значень кількості вузлів дискретизації, які зведемо до таблиці 1.

Таблиця 1

Ряд значень кількості вузлів дискретизації, отриманий за виразом (23)

Похибка ε	0.0001	0.001	0.005	0.0075	0.01	0.025	0.05	0.1
Кількість вузлів - N	10001	1001	201	134	101	41	21	11

Для визначення необхідної кількості вузлів за розглядуваним методом інтерполяції і подальшого порівняння з результатами табл. 1, візьмемо за основу оцінку похибки лінійної інтерполяції та відштовхуючись від нею будемо визначати кількість точок дискретизації, одночасно фіксуючи вплив на розподіл похибок відтворення. Згідно з [3, 4] оцінка максимальної похибки лінійної інтерполяції на відрізку $[t_{i-1}, t_i]$ має вигляд

$$\delta_i = \max_{[t_{i-1}, t_i]} |f(t) - L_1(t)| \leq h^2 \frac{M_{2,i}}{8}, \quad (24)$$

де $M_{2,i} = \max_{[t_{i-1}, t_i]} |f''(t)|$.

Враховуючи, що при сталому кроці дискретизації $h = \frac{T}{N}$ та розповсюджуючи оцінку на всі інтервали $M_2 = \max_T |f''(t)|$ знайдемо

$$N \geq T \sqrt{\frac{M_2}{8\delta}}.$$

Для визначеності прийmemo

$$N = \left\lceil T \sqrt{\frac{M_2}{8\delta}} \right\rceil + 1. \quad (25)$$

Далі постає питання вибору значення похибки інтерполяції інтегральної функції. Оскільки від цього значення залежить отримана точність наближення кривої, почнемо з оцінки $\delta = \varepsilon$, далі будемо зменшувати цю величину.

Для моделювання відтворення було обрано дві криві лінії (замкнута та незамкнута). Перша крива (рис. 2 а) є кривою Без'є шостого порядку з координатами точок опорного полігону (0.0; 0.0), (0.55; 6.9), (11.8; 9.2), (16.5; 6.25), (22.0; 3.2), (24.5; 0.0), (18.0; -1.0). Друга крива (рис. 2 б) носить назву равлика Паскаля та має параметри: радіус вихідного кола – 2.5, зміщення – 14. Обидві криві мають опуклу форму без точок перегину, що забезпечує відсутність точок розриву першої похідної функції (4), яка визначатиме значення M_2 в рівнянні (25), та виключає необхідність розгляду питання оптимізації параметру α в рівнянні (6).

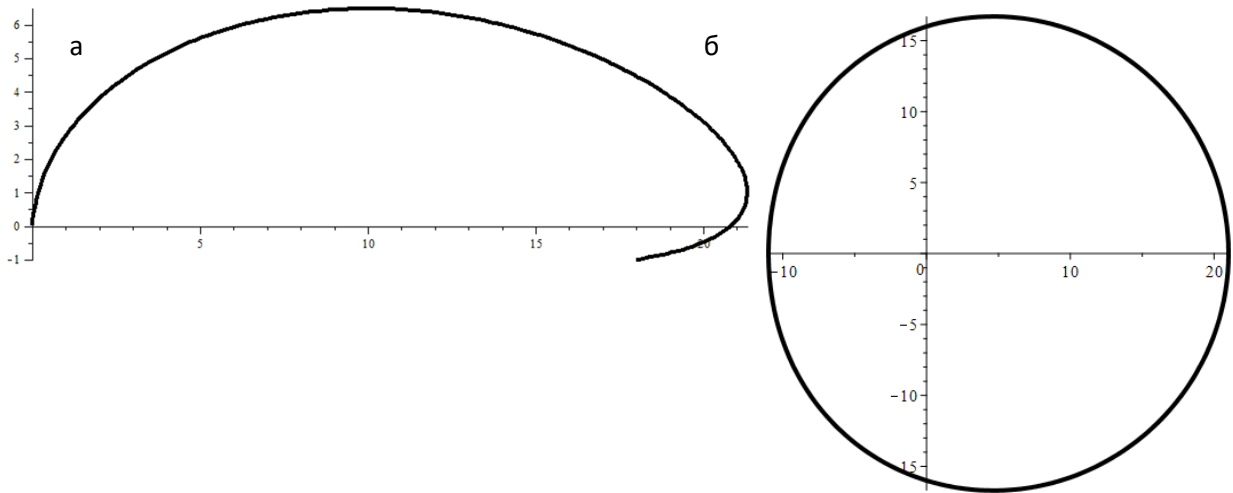


Рис. 2 Криві лінії обрані для відтворення ламаними:
а) крива Без'є, б) равлик Паскаля

В табл. 2 та 3 наведено результати відтворення кривих ламаними при лінійному методі інтерполяції інтегральної функції та сталому значенні похибки інтегрування $\delta^\Psi = 0.5 \cdot 10^{-10}$ методом Кленшоу - Кертиса при отриманні значень Ψ_i . Результати представлені у вигляді показників розподілів, що отримувались для послідовності відносних похибок ланок ламаних. Це: максимальне та мінімальні значення послідовності - γ_{\max} , γ_{\min} , середнє - $\bar{\gamma}$, медіана - Me , розмах значень відносних похибок для послідовності ланок ламаної - Δ , стандарт (середньо квадратичне відхилення) - σ та коефіцієнт варіації - c_v . Із

представлених результатів слідує, що при значеннях $\delta = \varepsilon/64, \varepsilon/256$ похибка апроксимації стабілізується знаходячись в межах діапазону допустимих значень досить близько до його верхньої границі – 0.9...0.98. Збільшення кількості на порядок не давало значного впливу на результати апроксимації. Більш наявно уявити результати можна безпосередньо порівнявши діаграми розподілів похибок при різних значеннях δ , зобразивши їх в одному діапазоні значень – рис. 3 та 4.

Результати моделювання кривих з наявними точками перегину.

Розглянемо результати інтерполяції ламаними за формулою розподілу – (5) для випадків кривих, що зображені на рис. 5. Перша крива є кривою Без'є п'ятого порядку (рис. 5а) та має одну точку перегину, а інша носить назву равлика Паскаля (рис. 5б) з двома точками перегину. При відтворенні цих кривих були враховані отримані результати щодо визначення кількості точок первісної дискретизації функції (5). Однак, оскільки для таких кривих друга похідна у виразах (24), (25) має розриви, бралось значення $N = 10 m$.

Щоб оцінити якість відтворення кривих, скористаємось запропонованою раніше методикою та побудуємо декілька стовпчикових діаграм розподілів похибок за ланками інтерполяційної ламаної. На рис. 6 зображено розподіли похибок кривих, де по горизонтальній осі відкладено номери ланок ламаної, що відраховуються в сторону збільшення параметру кривої, а по вертикальній осі – відносні похибки для відповідної ланки. На цих рисунках можливо побачити аномальні мінімуми та максимуми, які пояснюються наявністю у кривих точок перегину, де кривина приймає нульове значення та відбувається зміна її знаку. Формула (5) інтегральної функції не враховує можливу наявність таких ділянок кривої, оскільки від'ємні значення кривини нівелюються присутністю модулю. Отже, такі значення можна вважати за викиди. Так, для розподілу кривої з одною точкою перегину (рис. 6а) значенню максимуму похибки, що відбувається на 18 ланці, відповідає інтервал зміни параметру t кривої від, який містить значення, де підінтегральна функція (4) приймає нульове значення. Саме таке значення містить й інтервал 53 ланки, де знаходиться мінімум розподілу на рис. 6б. Аналогічні до попереднього випадку тенденції демонструють результати моделювання відтворення іншої кривої – рис. 7. Завдяки наявності вже двох симетричних відносно горизонтальної осі точок перегину равлика Паскаля, спостерігаються аномалії мінімальних та максимальних значень розподілів, які так само є симетричними.

Таблиця 2

Результати відтворення кривої – рис. 2а при $\varepsilon=0.001$ (кількість вузлів апроксимації $m = 122$)

Показник	Значення похибки інтерполяції інтегральної функції - δ						
	ε	$\varepsilon/4$	$\varepsilon/8$	$\varepsilon/32$	$\varepsilon/64$	$\varepsilon/256$	ε^2
γ_{\max}	0.98939	0.97467	0.97235	0.97148	0.97124	0.97122	0.97120
γ_{\min}	0.95361	0.96763	0.96939	0.97040	0.97060	0.97070	0.97072
Δ	0.03578	0.00704	0.00296	0.00108	0.00063	0,00052	0.00047
$\bar{\gamma}$	0.97093	0.97091	0.97092	0.97091	0.97091	0.97091	0.97091
Me	0.97045	0.97083	0.97091	0.97090	0.97090	0.97091	0.97091
σ	0.00731	0.00106	0.00063	0.00020	0.00014	0.00010	$9.72 \cdot 10^{-5}$
$c_v, \%$	0.752	0.109	0.065	0.020	0.015	0.010	0.010
N	73	144	203	405	572	1143	2257

Таблиця 3

Результати відтворення кривої – рис. 2б при $\varepsilon=0.01$ (кількість вузлів апроксимації $m = 91$)

Показник	Значення похибки інтерполяції інтегральної функції , δ						
	ε	$\varepsilon/4$	$\varepsilon/8$	$\varepsilon/32$	$\varepsilon/64$	$\varepsilon/256$	ε^2
γ_{\max}	1.03187	0.98851	0.97909	0.97709	0.97643	0.97597	0.97595
γ_{\min}	0.91859	0.96454	0.97131	0,97454	0.97506	0.97529	0.97536
Δ	0.11328	0.02397	0.00778	0.00256	0.00137	0.00068	0.00059
$\bar{\gamma}$	0.97588	0.97579	0.97578	0.97578	0.97578	0.97578	0.97578
Me	0.97549	0.97517	0.97604	0.97578	0.97579	0.97579	0.97578
σ	0.01965	0.00537	0.00177	0.00058	0.00026	0.00010	$9 \cdot 10^{-5}$
$c_v, \%$	2.014	0.551	0.181	0.059	0.027	0.010	0.009
N	28	55	78	154	217	433	764

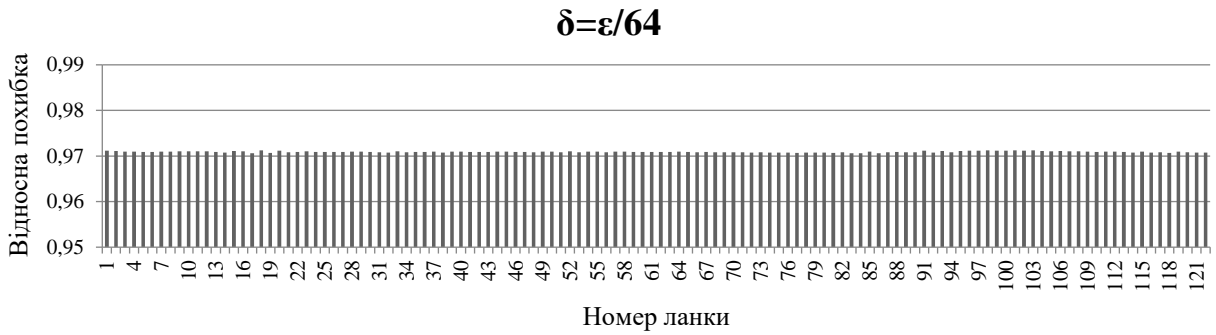
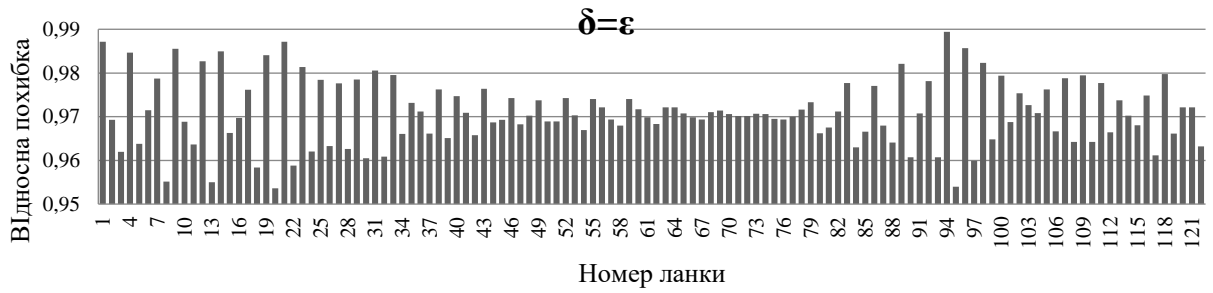


Рис. 3 Розподіли похибок при відтворенні кривої– рис. 2а із заданою точністю $\epsilon=0.001$

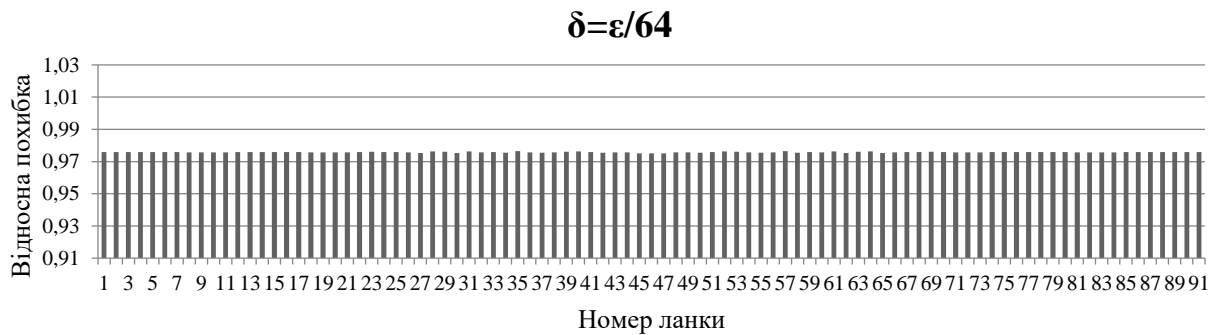
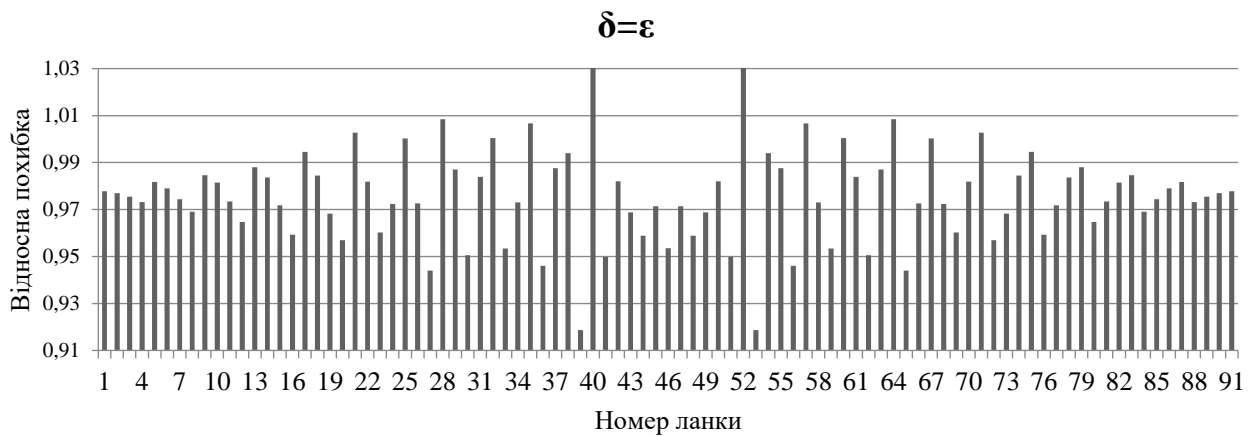


Рис. 4 Розподіли похибок при відтворенні кривої– рис. 2б із заданою точністю $\epsilon=0.01$

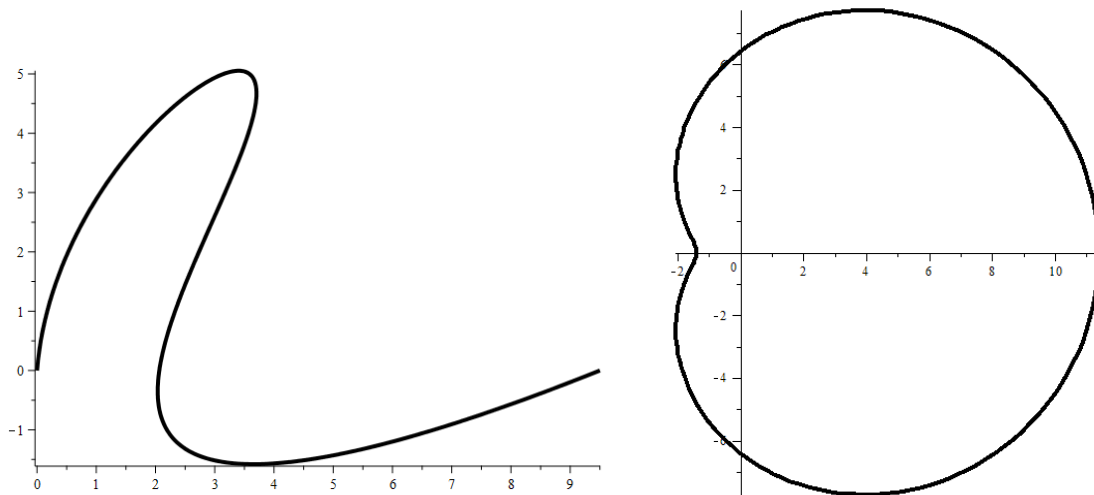


Рисунок 5 Криві лінії обрані для відтворення ламаними: а) крива Без'є п'ятого порядку з координатми точок опорного полігону (0.0; 0.0), (0.55; 6.9), (11.8; 9.2), (-2.5; -1.25), (-1.0; -4.2), (9.5; 0.0); б) равлик Паскаля: радіус вихідного кола – 2.5, зміщення – 6.4.

Отже, застосування функції – регулятора вузлів у формі (5) до випадку, коли крива, що відтворюється має точки перегину, дає результати апроксимації з порушенням монотонності на ланках ламаної, які відповідають ділянкам дуги, де відбувається зміна напрямку кривини. Це може призводити до появи аномальних значень похибки апроксимації, які перевищують допустимі значення. Запропоноване у [2] використання формули (6) до таких випадків, передбачає наявність доданка, за рахунок якого можливо дещо зменшити вузлові значення, щоб уникнути занадто великих значень похибки. Але в цій роботі було тільки визначенні межі зміни параметру α без вказівок, за якими критеріями може бути обране конкретне значення цього параметру при існуванні кривої та необхідній точності її наближення.

Таким чином, набуває актуальності питання можливості оптимізації значення параметру α при визначенні вузлів інтерполяції плоских кривих, що мають точки перегину.

Обговорення критеріїв, за якими один розподіл похибок апроксимації можна вважати кращім аніж інший, почнімо з визначення того, якими рисами може мати оптимальне рішення. Вочевидь, розподіл похибок інтерполяції можна вважати таким, якщо при той самій кількості ланок він забезпечує: - значення

послідовності похибок, які не перевищують допустиме значення; - значення окремих похибок близькі між собою, іншими словами забезпечується мінімальне середньо квадратичне відхилення послідовності похибок, або коефіцієнт варіації; - відсутність аномалій, тобто занадто малих або високих значень, що відрізняються від інших. Згідно з цим, за показники послідовностей похибок, що підлягають дослідженню можна прийняти максимальне значення відносної похибки послідовності, її середнє значення, середнє квадратичне відхилення та коефіцієнт варіації.

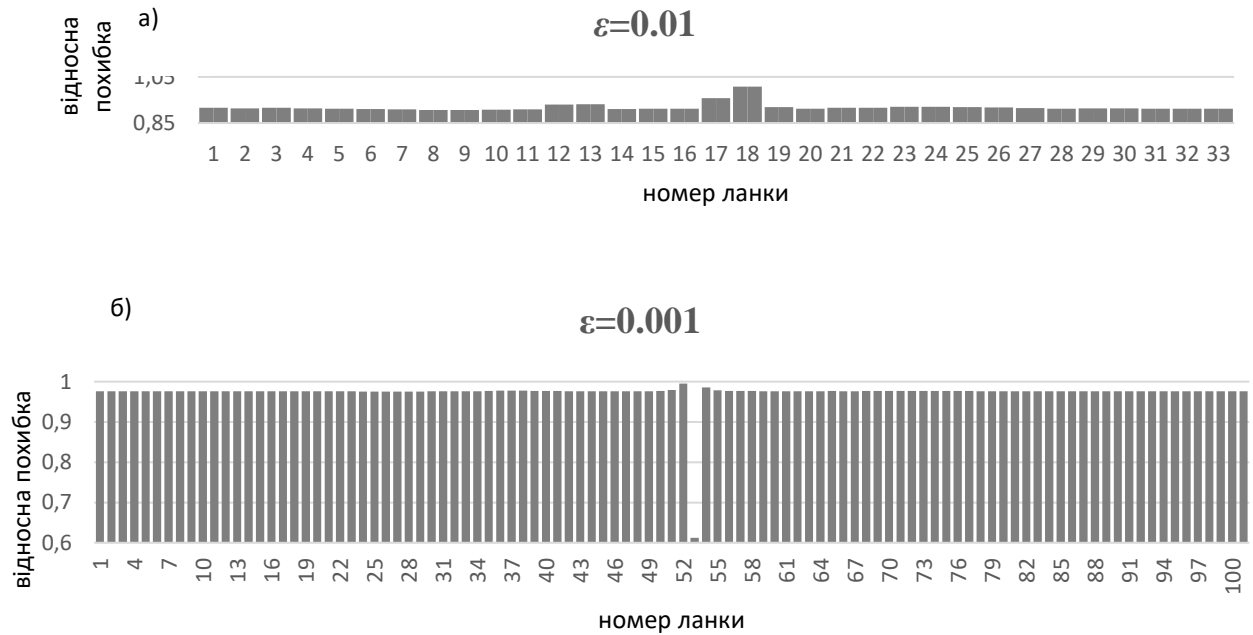
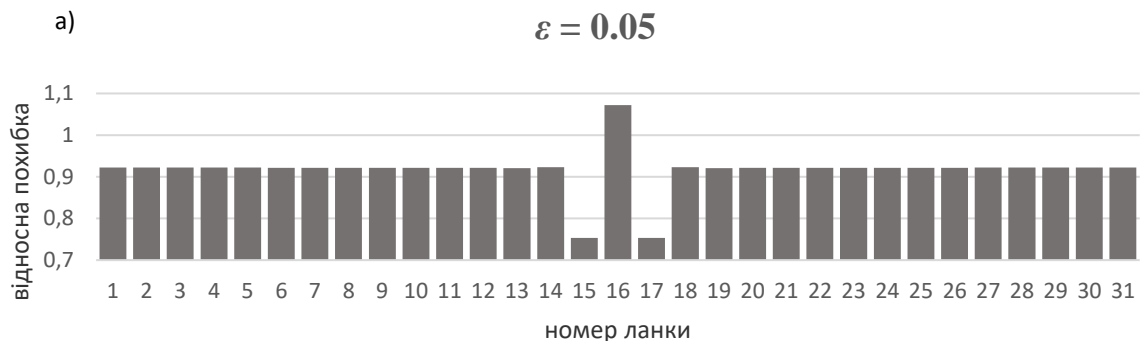


Рис. 6 Результати відтворення кривої Без'є (рис. 1а) при визначення вузлів інтерполяції за виразом (3) при допустимій похибці: а) $\varepsilon = 0.01$; б) $\varepsilon = 0.001$



б)

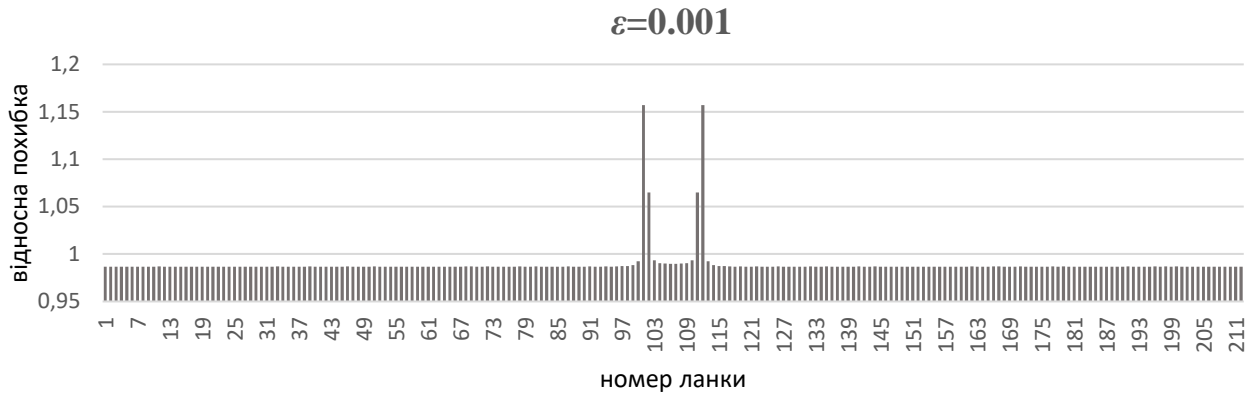


Рис. 7 Результати відтворення равлика Паскаля (рис. 1б) при визначення вузлів інтерполяції за виразом (3) при допустимій похибці:

а) $\varepsilon = 0.05$; б) $\varepsilon = 0.001$

Для вивчення характеру зміни зазначених показників, побудуємо графіки зміни цих величин в залежності від значень параметру α функції- регулятора вузлів. Побудову здійснимо для тих самих кривих рис. 5а та 5б. Щоб отримати значення кожного з показників робилась попередня дискретизація значень по t функції – регулятора, обчислювались значення інтегральної функції Ψ з урахуванням доданку (6), далі отримувались вузлові значення параметру для асимптотично - оптимального розподілу, координати вузлових точок ламаної інтерполяції i , нарешті, значення відстаней від кривої до кожної ланки ламаної, які брались за абсолютні значення похибок апроксимації. Зіставивши значення ряду абсолютних похибок до значення допустимої похибки ε були отримані послідовності відносних похибок $\{\gamma_j\}$, за значеннями яких отримувались статистичні характеристики послідовностей. Значення параметру α функції-регулятора приймались виходячи з меж, що були визначені в [2], зі збільшенням вправо до 1, тобто в інтервалі $[0;1]$. При цьому для обчислення значень рядів похибок брались 50 значень рівномірного розбиття цього інтервалу (побудовано 50 інтерполяційних ламаних заданої кривої).

На рис. 8 зображено графіки показників розподілів похибок при $\varepsilon = 0.001$ для кривої Без'є (рис. 5а.), аналогічні ним тенденції виявляють графіки для равлика Паскаля (рис. 5б).

Проаналізувавши побудовані графіки, можна відмітити наступне:

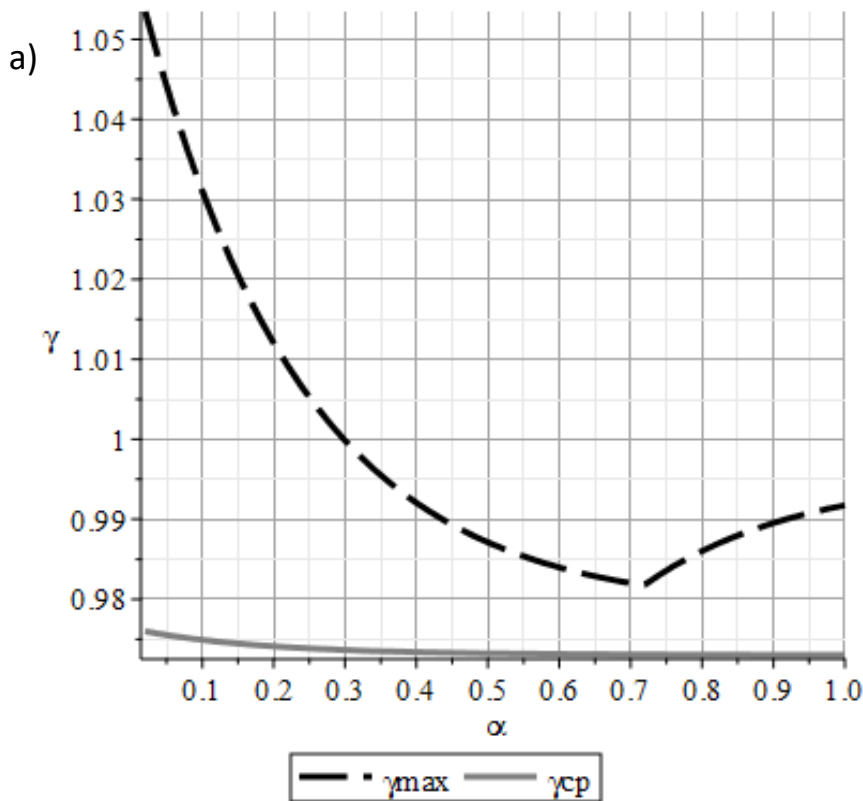
- для обох кривих присутні екстремуми (мінімуми) на графіках максимальної похибки;

- графіки значень середньої похибки рядів менш всього були схильні до змін (для першого випадка в межах зміни параметру графік монотонно спадав наближаючись до асимптоти, для другого випадку на графіки присутній слабкий

мінімум, який за значенням параметру дає максимальну похибку, що перевищує допустиме значення);

- порівнявши графіки середньо квадратичного відхилення та коефіцієнту варіації можна відмітити, що вони демонструють однакові тенденції, що можна просто пояснити, адже коефіцієнт варіації залежить від стандартного відхилення та середнього для послідовності похибок, при чому остання величина досить слабо змінюється. Для першої кривої ці графіки не виявляють екстремальних значень, спадаючи зі збільшенням α наближаючись до горизонтальної асимптоти. Для другої кривої графіки, що розглядаються, мають два мінімуми (локальний та глобальний), при цьому, якщо значення α для екстремумів порівняти зі значеннями екстремумів для графіків інших характеристик, то можна дослідити, що першому (меншому) мінімуму відповідають значення близькі до мінімуму для середньої похибки послідовності, а другому (глобальному) мінімуму – значення мінімуму для максимальної похибки;

- графіки розмаху значень похибок ламаних для обох кривих загалом відображають аналогічні до зміни стандартного відхилення та коефіцієнту варіації тенденції залежності з більш чітко означеними у випадку другої кривої максимумами та мінімумами.



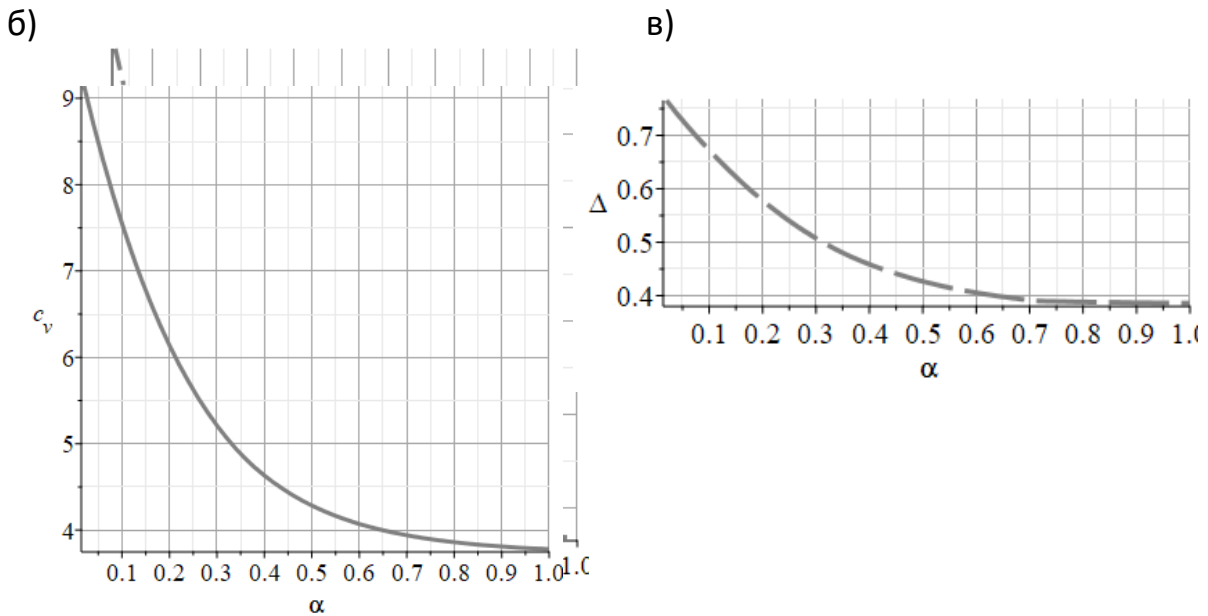


Рис. 8 - Графіки статистичних характеристик рядів похибок в залежності від значень параметру α для кривої – рис. а: а) максимальна та середня похибка; б) коефіцієнт варіації; в) розмах

Таким чином, зважаючи на пріоритет забезпечення похибки апроксимації в допустимих межах для всіх ланок ламаної, а також, враховуючи описані вище тенденції зміни показників розподілу з урахуванням припущення унімодальності для методів оптимізації [5], можна обрати за цільову функцію максимальну серед усіх похибок послідовності ланок ламаної. Тоді задачу мінімізації цільової функції можна записати у вигляді

$$f(\alpha) = \max \{ \delta_i (i = 1..m) \} \rightarrow \min . \quad (5)$$

Оскільки вираз цільової функції не носить аналітичний характер та не може бути знайдена її похідна, до пошуку оптимального рішення можуть бути застосовані методи нульового порядку [5], стратегія яких базується на використанні інформації тільки про цільову функцію. Серед цих методів було обрано методи золотого перетину та більш швидкий метод Брента [5] який використовує комбінацію методу золотого перетину та метод квадратичної інтерполяції.

Результати обчислення значення параметру α за вказаним алгоритмом для кривих – рис. 5 для різної точності наближення наведено в таблиці 4. Порівнявши результати наведені в цій таблиці для $\varepsilon = 0.001$ з графіками максимальної похибки, пересвідчимось у відповідності результатів обчислень.

Результати пошуку оптимального значення параметру α

Назва кривої	Значення параметру α при похибці апроксимації - ε			
	$\varepsilon = 0.05$	$\varepsilon = 0.01$	$\varepsilon = 0.005$	$\varepsilon = 0.001$
Крива Без'є (рис. а)	0.99609375	0.4765625	0.61328125	0.71875
Равлик Паскаля Рис. б	0.4296875	0.71875	0.734375	0.6171875

На рис. 9 у вигляді стовпчикових діаграм показані послідовності похибок ланок ламаних при отриманні їх за оптимізованими значеннями параметру α в рівнянні (4). Якщо порівняти ці діаграми з відповідними розподілами, що зображені на рис. 6 та 7, можна побачити, що отримані в результаті оптимізації послідовності вже не містять значень, що перевищують допустиму похибку.

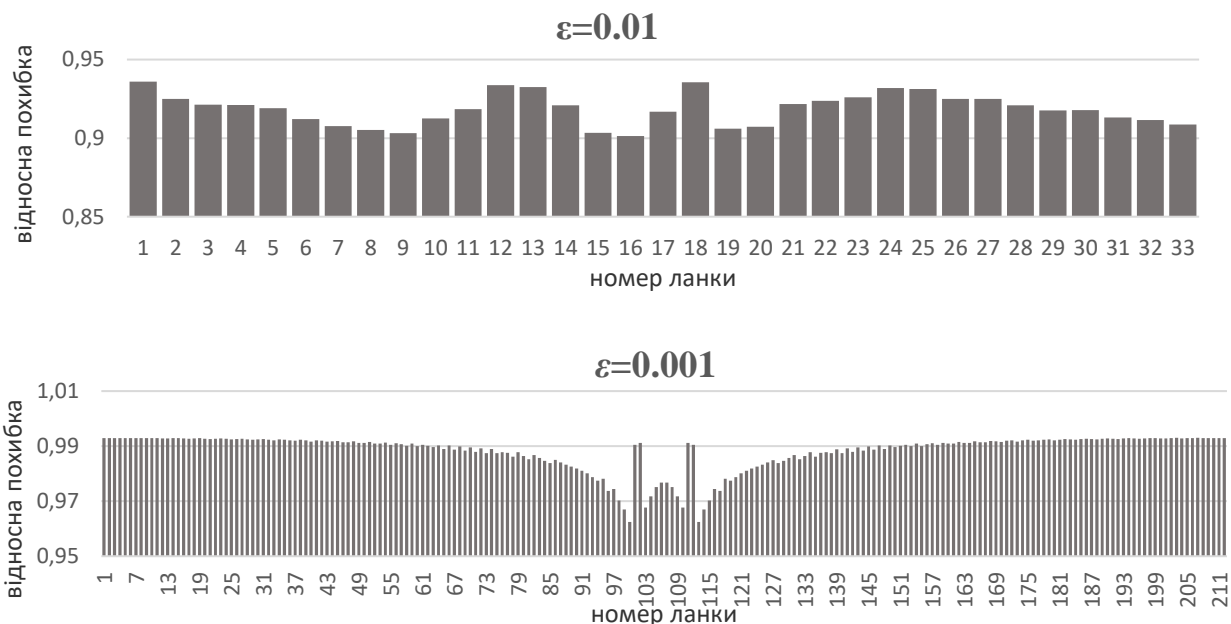


Рисунок 9 Результати відтворення кривих при визначення вузлів інтерполяції за виразом (6) з оптимізацією параметру α : а) крива рис. 5а; б) крива рис. 5б

Обговорення результатів і висновки.

Моделювання відтворення реальних плоских параметричних кривих опуклої форми ламаними за асимптотично - оптимальним алгоритмом з лінійним методом інтерполяції значень інтегральної функції показало цілком прийнятні

результати роботи алгоритму без перевищення допустимої похибки апроксимації у випадках достатнього ступеню дискретизації значень зазначеної функції.

Порівняння кількісних показників дискретизації за виразами (23) та (25) демонструє, що формула (23) може давати як підвищену, так і знижену кількість точок дискретизації по відношенню до значень стабілізації похибки апроксимації на рівні $\epsilon/64$ (див. табл. 2, 3).

Результати моделювання також дозволили встановити вплив підвищення якості апроксимації інтегральної функції за рахунок збільшення кількості вузлів дискретизації на результати відтворення кривої ламаною. Він полягає в покращенні показників розподілу похибок апроксимації за ланками апроксимуючої ламаної, а, саме зниження коефіцієнтів варіації та середньоквадратичних відхилень цих рядів, що відображає стабілізацію значень похибок навколо середнього значення.

Проведені дослідження з моделювання відтворення плоских кривих, що мають точки перегину, показали придатність застосування формули (4) з оптимізацією параметру α для регулювання розподілу вузлів лінійної інтерполяції таких кривих за асимптотично – оптимальним алгоритмом. При цьому обрання за цільову функцію максимуму серед похибок послідовності ланок ламаної дозволило забезпечити рівень похибок нижче граничного значення. З результатів моделювання слідує, що серед значень параметру α , які відповідали оптимальному розв'язку для різних умов, були значення, що виходили за вказані в роботі [2] межі. Отже, це питання має бути досліджено окремо.

Список літератури

1. A real-time interpolator for parametric curves / W. Zhong, X. Luo, W. Chang [etc] // International Journal of Machine Tools and Manufacture, 2018. - v. 125 - P. 133-145.
2. Лигун А.А. Асимптотические методы восстановления кривых/ А. А. Лигун. А. А. Шумейко - Киев: Институт математики НАН Украины, 1997. - 358 с.
3. Фельдман Л. П. Чисельні методи в інформатиці / Л. П. Фельдман, А. І. Петренко, О.А. Дмитрієва. – К.: ВНУ, 2006. – 480 с.
4. Волков Е. А. Численные методы. Учеб. пособие для вузов / Е. А. Волков. - 5-е изд. испр. - СПб.: Лань, 2008. - 248 с.
5. Arora Rajesh K. Optimization: algorithms and applications / Rajesh K. Arora. - Hoboken: CRC Press, 2015. – 454 p.

ГЛАВА 10

РОЗРОБКА ПЕДАГОГІЧНОЇ МОДЕЛІ ФОРМУВАННЯ КОМПЕТЕНЦІЙ ФАХІВЦІВ ПОЛІГРАФІЧНОГО ПРОФІЛЮ З ПОЗИЦІЇ СИСТЕМНОГО ПІДХОДУ

Вступ і постановка задачі

Сучасний стан економіки, виробництва і суспільних відносин в нашій країні в порівнянні з досягненнями науково-технічної революції в розвинених країнах світу призвело до необхідності переосмислення основних цілей і завдань вітчизняної педагогіки, основних вимог до наукової та практичної підготовки сучасної людини. З урахуванням цього на одне з перших місць в освіті виходить завдання підготовки молоді до творчої праці, який є каталізатором засвоєння нової наукової і технічної інформації, прискорює її творчу переробку і генерацію компетентностей сучасного фахівця.

У Радянському Союзі існувала централізована система керівництва і організації творчої діяльності учнів. Функціонувала мережа гуртків моделювання, шкіл юних раціоналізаторів та винахідників, конструкторських бюро учнів, станцій юних техніків різних рівнів. Були розроблені численні форми організації технічної творчості учнів шкіл і профтехучилищ, студентів вузів. Незважаючи на це вже в 70-х роках з'явилися дослідження, що відзначали недостатню ефективність існуючої системи технічної творчості учнів.

Здатність до самостійного, творчого мислення, до ініціативної творчої діяльності не є побічним ефектом процесу засвоєння знань, не розвивається сама по собі. Навпаки, як зазначається в численних дослідженнях творчої діяльності учнів, розвиток творчих здібностей вимагає особливої уваги і спеціального педагогічного впливу. У навчальних закладах необхідно свідомо створювати творчу атмосферу.

Нам вдалося виявити у вітчизняній педагогіці не так багато спеціальних робіт, предметом яких була б педагогічна методологія системного підходу в реалізації технічної творчості: В. П. Беспалько, Ф.Ф.Королев, Н. В. Кузьміна, А.М.Саранов, А.М.Сідоркін [1-5]. Як правило, педагоги-дослідники в роботі над конкретними проблемами або безпосередньо використовували алгоритм загальнонаукового системного підходу, або розробляли спеціальну методику системного підходу для вирішення локальних завдань власного дослідження.

Нечастими були випадки, коли вчений при розробці методики системного підходу для конкретного дослідження проробляв серйозну методологічну роботу з обґрунтуванням специфіки методології системного підходу в педагогіці.

З середини 70-х років, коли вже нікому не треба було доводити правомірність застосування системного підходу в педагогіці шляхом обґрунтування системної природи педагогічних об'єктів, дослідники звертаються до пошуку специфічних особливостей педагогічних об'єктів як систем.

Спочатку пошуки в цьому напрямку були пов'язані з спробами перетворення педагогічної системи до будь-якого іншого, більш простого або більш вивченого класу систем (інформаційних, управлінських, соціальних) і відповідної їх характеристикою на основі принципу ізоморфізму.

Так, І.П.Бім [6], характеризуючи систему технічної творчості як складну, зазначає, що "її найбільш істотне властивість - інформаційна природа, тобто функціональна націленість на передачу і прийом, зберігання і переробку інформації про навколишню дійсність з метою спілкування індивіда до трудового досвіду попередніх поколінь, з метою формування його особистого досвіду". Системо утворюючими властивостями системи дослідник вважає цілісність як єдність функцій всіх елементів і керованість. Як бачимо, спроба визначити специфіку педагогічної системи в даному випадку закінчилась зведенням її до іншого класу систем - інформаційних систем, а виділення системо утворюючих властивостей взагалі перекреслило зусилля дослідника виділити "особливе", оскільки дані властивості, по-перше, характерні для кожного другого системного об'єкта самої різної природи, а по-друге, не є системо утворюючими.

В. П. Беспалько [1] специфіку технічної творчості вбачав в тому, що в них протікають творчі процеси, визначаючи педагогічну систему як систему управління цими творчими процесами. Контекстуально можна визначити, що відмінність творчих процесів від інших зв'язується вченим із забезпеченням переходу абітурієнт-фахівець, але чи впливає ця характеристика педагогічного процесу на появу специфічних рис в системі управління, вчений не з'ясовує. Судячи з того, що управління В. П. Беспалько розглядає як "будь-який вплив на систему з метою підтримки або зміни її алгоритму функціонування", зведення педагогічної системи до системи управління не було продуктивним для визначення внутрішньої специфіки педагогічних систем.

Ряд педагогів вичерпували характеристику особливостей технічної творчості зазначенням на її приналежність до класу соціальних систем. Так, А.М.Саранов [5], метою дослідження, якого було виявлення методологічних регулятивів системного підходу для побудови цілісної теорії творчого процесу, організованого класним керівником, пов'язував можливості досягнення цієї мети з визначенням процедур дослідження на основі системного підходу за умови врахування специфіки педагогічних явищ. Однак єдиним зазначенням на специфіку є твердження про те, що "соціальні системи (в тому числі і педагогічні) характеризуються доцільністю, тобто прагненням до досягнення цілей. Це один з її системно утворюючих чинників". Тобто і в цьому випадку педагогічна система зводиться до системи певного класу - соціальної, і розглядається вже як соціальна система. Причому, з усіх характеристик соціальної системи, що відрізняють її від біологічних, механічних та інших систем, і вже відомих в філософії і загальнонауковому системному підході обрана тільки одна - доцільність. Таким чином, сутність педагогічної системи редукована двічі: спочатку зведенням її до соціальної системи, потім зведенням суті соціальної системи до доцільності.

Проведений аналіз літературних джерел дозволяє розглядати дослідження проблем технічної творчості у вигляді цілісної методології системного підходу, що включає в себе процедури різних епістемологічних рівнів - від виділення об'єкта пізнання до конструювання реальної системи в конкретних умовах.

Сучасні методи побудови творчих систем повинні бути пов'язані не стільки з оцінкою і відбором найбільш правдоподібних версій, скільки з синтезом результатів досліджень в загальну концепцію педагогічних систем. Визначення специфіки педагогічних систем неможливо без послідовного застосування процедур системного підходу.

У зв'язку з цим перспективним є шлях вирішення проблеми специфіки педагогічних систем: від загальносистемних характеристик педагогічних об'єктів (склад, структура, функції, системно утворюючі зв'язки, властивості і т.п.) до їх власне педагогічному змісту.

Метою роботи є розробка системної моделі формування компетентностей у фахівців поліграфічного профілю.

Основна частина

Творчий потенціал як соціально значуща якість людини є однією з найважливіших його характеристик як члена того чи іншого суспільства людей, творчої особистості. Для розвитку в особистості людини такої якості необхідна гнучка методологія навчального процесу, що базується на законах психології освіти і творчості, історичному підході до розвитку науки, техніки і технології. Це вимагає внесення змін і корекції в концепцію вищої освіти, в її змістовні компоненти.

У представленому дослідженні значна частина присвячена розробці основ викладання спеціальних поліграфічних дисциплін як фактору, якій розвиває змістовний аспект навчання, підйому творчого потенціалу особистості молодого фахівця.

У коло розроблюваних проблем включені також завдання аналізу природничо наукових основ і компонентів творчої діяльності людини, соціально-психологічного підходу до формування мотивів творчої активності особистості на стадії навчання, дослідження технологічних аспектів викладання спеціальних дисциплін, впливу міжособистісних відносин між викладачами і студентами на розвиток компетентностей молодого фахівця.

У процесі дослідження творчого потенціалу майбутніх фахівців нами було виділено найважливіші якості особистості фахівця за наступними напрямками:

Якості, які виражають ставлення до роботи: працьовитість; уважне ставлення до роботи; творче ставлення до роботи.

Якості, що характеризують загальний стиль поведінки і діяльності: старанність; самостійність; вірність слову; авторитетність; енергійність.

Знання: технічні знання по своїй спеціальності; математичні знання; фізичні знання; загальна культура; інформованість про діяльність підприємства, його завдання та плани.

Якість розуму: гнучкість, прозорливість, інтуїція.

Інженерно-організаційні вміння: вміння вирішувати технічні завдання; вміння працювати з технічною літературою і довідниками; вміння бачити завдання; вміння навчати роботи; вміння орієнтуватися в роботі; досвідченість в роботі; вміння проводити технічну політику; вміння здійснювати взаємодію з іншими підрозділами; вміння пояснити технічне завдання; вміння планувати роботу.

Адміністративно-організаторські вміння: вміння створювати трудову атмосферу; вміння керувати людьми; вміння постояти за колектив; вміння відстояти ідею; вміння розбиратися в людях; вміння переконувати людей.

Якості, що характеризують ставлення до людей: чесність; неупередженість; вихованість.

Якості, що характеризують відношення до себе: вимогливість; скромність; впевненість; самовдосконалення.

Наведений перелік найважливіших якостей фахівця, послужив основою для складання моделі формування компетентностей у фахівців поліграфічного профілю в процесі вивчення спеціальних дисциплін (табл. 1).

У структурі розвитку навичок технічної творчості виділяють ряд видів навчання, що формуються на основі різноманіття методів діяльності педагогів і учнів. Найбільшого поширення набув репродуктивно-проблемний вид навчання, основу якого складає пояснення матеріалу вчителем і репродуктивне засвоєння знань учнями. Репродуктивно-проблемний вид навчання це своєрідне поєднання прийомів навчання, орієнтованих на репродуктивне засвоєння навчального матеріалу, з прийомами, що залучають в самостійний пошук знань в процесі вирішення проблемних ситуацій, а також виконання практичних завдань з внесенням елементів раціоналізації в порівнянні з типовими приписами.

При репродуктивно-пошуковому підході необхідно розвивати якості особистості в складі дисциплін представлених в табл. 1. При цьому важливо вибрати найкраще поєднання репродуктивних і пошукових елементів, а також виявляти максимальні можливості для пошукового підходу до навчання.

В процесі навчання проявляється дія цілого ряду педагогічних законів. Наприклад: соціальної обумовленості процесу навчання, єдності викладання і навчання, єдності навчання і розвитку особистості. Прояв кожного з законів висуває певні вимоги до побудови процесу навчання. Багато з цих вимог на основі багаторічного досвіду синтезовані в деякі найбільш істотні вимоги, які отримали назву принципів навчання.

Багато дослідників, висуваючи цю проблему на перший план, підкреслюють, що основними джерелами перетворення змісту професійної освіти повинні стати наукові дослідження і дослідно-конструкторські розробки, високі технології і актуальна соціальна практика.

Висуваючи на перший план технічну складову, розглянемо вплив інженерно-організаційних умінь на спеціальні компетентності фахівців .

Тоді табл.1. приймає такий вигляд (табл.2).

Матриця має п'ять рядків (А, Б, С, Д, Е) і п'ять стовпців. Завдання наукового дослідження є розподілити дисципліни (або теми які вивчаються) по позиціях даної матриці. При цьому необхідно буде надати дисциплінам вагові категорії.

Введення кількісних оціночних балів (за важливістю) для кожного з якостей дозволить здійснювати оцінку масиву навчальних методик з метою їх ранжування (наприклад, на основі методу експертних оцінок і статистичної обробки даних експертів).

У кожній дисципліні (загальнонаукової, загально технічної, спеціальної) студент отримує фундаментальні знання відповідного рівня. Проблема підйому рівня компетентностей пов'язана з якістю системології навчальних дисциплін, з наявністю в них таких якостей знання, як узагальненість, конкретність, повнота і ефективність їх застосування.

Інтелектуально-творча спрямованість розвитку особистості молодого фахівця, формування та закріплення стійких інтелектуальних якостей може бути досягнуто за рахунок підвищення рівня інтелектуалізації змісту вищої освіти як основи розвитку творчого мислення. Воно пов'язано з насиченням навчальних дисциплін сучасними науковими концепціями і гіпотезами. З формуванням в них сучасних систем знань, що характеризуються такими якісними показниками, як стрункість, динамічність, узагальненість, високий рівень пізнавальних орієнтирів. Визначається виробленням механізмів мислення, що стимулюють інтелектуальний розвиток, закріпленням в них основ методології наукового пізнання, раціональних методів вибору і постановки мети, концепції її досягнення, здатності до оціночних дій.

Наукові знання, що вводяться в навчальні дисципліни, повинні відповідати не тільки інформаційним, а й цілям що розвивають. Для цього повинні передбачатися встановлення широких зв'язків і узагальнень в досліджуваному матеріалі, перенесення засвоєних знань і способів оперування ними на новий матеріал.

Модель формування компетентностей у фахівців поліграфічного профілю в процесі вивчення спеціальних дисциплін

Компетентності		Дисципліни по курсам				
		1-ий	2-ий	3-ий	4-ий	5-ий
1	2	3	4	5	6	7
Знання	технічні знання по своїй спеціальності і математичні знання; фізичні знання; загальна культура	Вступ до спеціальності Історія української культури Історія України	Вища математика Фізика Хімія Взаємозамінність, стандартизація та технічні вимірювання	Філософія Електротехніка та електроніка Технічна механіка	Екологія Політологія	Інтелектуальна власність Системний аналіз
Інженерно-організаційні вміння	вміння вирішувати технічні завдання; вміння працювати з технічною літературою і довідниками; вміння бачити завдання;	Інженерна та комп'ютерна графіка Інформатика та обчислювальна техніка	Основи технології поліграфічних виробництв Теоретичні основи технологій друку	Поліграфічні матеріали Теоретичні основи формних процесів Технологія обробки графічної інформації	Поліграфічне обладнання Технології після друкарських процесів Технологія друку Пакувальне виробництво Технології оперативних засобів друку	Макетування поліграфічної продукції Основи наукових досліджень Проектування поліграфічних підприємств

1	2	3	4	5	6	7
	вміння пояснити технічне завдання; вміння планувати роботу.			Оцінка та контроль якості поліграфічної продукції Комп'ютерні системи розробки поліграфічних видань	Технологія обробки текстової інформації	
Адміністративно-організаторські вміння	вміння створювати трудову атмосферу; вміння керувати людьми; вміння постояти за колектив; вміння відстояти ідею; вміння розбиратися в людях;	Психологія Соціологія	Вікова та педагогічна психологія Психологія праці	Дидактичні основи професійної освіти Основи інженерно-педагогічної творчості	Комунікативні процеси у педагогічній діяльності Економіка підприємств та маркетинг	Менеджмент освіти Маркетинг і менеджмент поліграфічних підприємств

Модель формування інженерно-організаційних навичок

Компетентності	Дисципліни по курсам				
	1-ий	2-ий	3-ий	4-ий	5-ий
<p>А) вміння вирішувати технічні завдання;</p> <p>Б) вміння працювати з технічною літературою і довідниками;</p> <p>С) вміння бачити завдання;</p> <p>Д) вміння пояснити технічне завдання;</p> <p>Е) вміння планувати роботу.</p>	<p>Інженерна та комп'ютерна графіка</p> <p>Інформатика та обчислювальна техніка</p>	<p>Основи технології поліграфічних виробництв</p> <p>Теоретичні основи технологій друку</p>	<p>Поліграфічні матеріали</p> <p>Теоретичні основи формних процесів</p> <p>Технологія обробки графічної інформації</p> <p>Оцінка та контроль якості поліграфічної продукції</p> <p>Комп'ютерні системи розробки поліграфічних видань</p>	<p>Поліграфічне обладнання</p> <p>Технології після друкарських процесів</p> <p>Технологія друку</p> <p>Пакувальне виробництво</p> <p>Технології оперативних засобів друку</p> <p>Технологія обробки текстової інформації</p>	<p>Макетування поліграфічної продукції</p> <p>Основи наукових досліджень</p> <p>Проектування поліграфічних підприємств</p>

Ефективним напрямком формування змісту вищої професійної освіти є включення в спеціальні навчальні дисципліни матеріалів, що відображають характер і динаміку науково-технічного прогресу і розвиток соціально-економічних процесів, принципи реалізації зв'язку "наука-технологія", поєднання професійно спрямованих фундаментальних знань з новими інтенсивними технологіями досліджень. Перспективним і найбільш ефективним напрямком інтелектуалізації змісту вищої професійної освіти має стати використання сучасної методології, що відображає єдність системного стилю мислення з моделюючим пізнанням, тобто насиченням усіх боків освіти доступними формами і методами матеріального і математичного моделювання, в тому числі геометричного, аналогового і інших.

В сучасних умовах в основу підготовки фахівців повинна бути покладена технологія реального моделювання всього циклу навчання від задуму до дослідження, від інженерного проектування, конструювання до реалізації розробки у споживача.

Успішність формування компетентностей складається з декількох складових. По-перше, якісний рівень професорсько-викладацького складу (наявність наукових ступенів, стаж викладацької роботи, кількість публікацій, участь у науковій роботі і т.п.). По-друге, готовність студентів сприймати нові знання (їх тезаурус). Третя складова - це ефективність передачі знань від професорсько-викладацького складу студентам, тобто визначення функції трансформації знань в уміння.

Встановлення цілей і кордонів систем пов'язано з відмінностями в критеріях ефективності системи. Система освіти може служити яскравим прикладом цього, так як різні люди оцінюють її по-різному. Викладачі, допоміжний штат, адміністрація, батьки і самі учні мають різні погляди на те, як фінансувати освіту, та й на якість освіти. Завдання полягає в тому, щоб узгодити і врахувати інтереси всіх сторін і зробити систему формування компетентностей майбутніх фахівців максимально відповідає запитам суспільства.

Дане завдання є різноманітним. При цьому на наш погляд необхідно реалізувати ряд кроків.

Крок 1. Вибір і призначення чинників.

У будь-якій багатовимірної задачі насамперед має бути визначення факторів, за якими буде проводитися порівняння різних варіантів. Такі

фактори, властивості або критерії повинні бути представлені у вигляді морфологічної матриці, яка б показала взаємозалежності між факторами.

Оцінку рівня технічної творчості студентів розглянемо на прикладі третього курсу.

З табл. 2 маємо п'ять рядків умінь: А) вміння вирішувати технічні завдання; Б) вміння працювати з технічною літературою і довідниками; С) вміння бачити завдання; Д) вміння пояснити технічне завдання; Е) вміння планувати роботу.

І п'ять стовпців містять дисципліни, що вивчаються: 1. Поліграфічні матеріали; 2. Теоретичні основи формних процесів; 3. Технологія обробки графічної інформації; 4. Оцінка і контроль якості поліграфічної продукції; 5. Комп'ютерні системи розробки поліграфічних видань.

Отримана морфологічна матриця представлена в табл. 3.

Таблиця 3

Морфологічна матриця формування компетентностей для 3-го курсу

		1	2	3	4	5
		Поліграфічні матеріали	Теоретичні основи формних процесів	Технологія обробки графічної інформації	Оцінка і контроль якості поліграфічної продукції	Комп'ютерні системи розробки поліграфічних видань
А	вміння вирішувати технічні завдання	P_{A1}	P_{A2}	P_{A3}	P_{A4}	P_{A5}
Б	вміння працювати з технічною літературою і довідниками	P_{B1}	P_{B2}	P_{B3}	P_{B4}	P_{B5}
С	вміння бачити завдання	P_{C1}	P_{C2}	P_{C3}	P_{C4}	P_{C5}
Д	вміння пояснити технічне завдання	P_{D1}	P_{D2}	P_{D3}	P_{D4}	P_{D5}
Е	вміння планувати роботу	P_{E1}	P_{E2}	P_{E3}	P_{E4}	P_{E5}

Тут коефіцієнти P_{ij} висловлюють вагові коефіцієнти впливу досліджуваної дисципліни на формування тих чи інших компетентностей. Розрахунок даних коефіцієнтів буде розглянуто нижче.

Кожен із зазначених коефіцієнтів може бути розбитий на під фактори, або компоненти, завдяки чому його можна оцінити і в підсумку виміряти. Подальша декомпозиція системи передбачає розбиття системи на досліджувані теми. Розглянемо декомпозицію на прикладі дисципліни «Оцінка і контроль якості поліграфічної продукції» (табл. 4). У таблиці 4 в шпальтах відображаються досліджувані теми

Таблиця 4

Морфологічна матриця формування навичок технічної творчості для дисципліни «Оцінка і контроль якості поліграфічної продукції»

		1	2	3	4	...	13
		Параметри якості друкованої продукції	Вимірювання кольору в поліграфії .	Денситометричні та колориметричні методи контролю	Контроль формних процесів		Організація статистичного контролю якості в друкарнях
А	вміння вирішувати технічні завдання	P_{A41}	P_{A42}	P_{A43}	P_{A44}	...	P_{A413}
Б	вміння працювати з технічною літературою і довідниками	P_{B41}	P_{B42}	P_{B43}	P_{B44}	...	P_{B413}
С	вміння бачити завдання	P_{C41}	P_{C42}	P_{C43}	P_{C44}	...	P_{C413}
Д	вміння пояснити технічне завдання	P_{D41}	P_{D42}	P_{D43}	P_{D44}	...	P_{D413}
Е	вміння планувати роботу	P_{E41}	P_{E42}	P_{E43}	P_{E44}	...	P_{E413}

Крок 2. Призначення ваг факторів і під факторів

Далі необхідно призначити відносні ваги кожного з п'яти умінь. Тобто визначити їх частку в формуванні спеціальних компетентностей. При цьому сума відносних ваг повинна дорівнювати одиниці.

Наприклад для теми «Параметри якості друкованої продукції» ваги можуть мати значення представлені в табл. 5.

Таблиця 5

Призначення відносних ваг розглядаються вмінням

	Уміння	Приклад призначення ваги
А	вміння вирішувати технічні завдання	$P_{A41} = 0,25$
Б	вміння працювати з технічною літературою і довідниками	$P_{B41} = 0,20$
С	вміння бачити завдання	$P_{C41} = 0,30$
Д	вміння пояснити технічне завдання	$P_{D41} = 0,15$
Е	вміння планувати роботу	$P_{E41} = 0,10$

Крок 3. Нормалізація ваг

Ваги факторів і під факторів необхідно нормалізувати. Цей крок полягає в перемноженні ваг, отриманих для кожного окремого фактора, і ваг під факторів що відносяться до нього. Результатом множення є підсумкова вага для кожного даного під фактору.

$$P_{A4} = P_{A41} \cdot P_{A42} \cdot P_{A43} \cdot P_{A44} \cdot P_{A45} \cdot \dots \cdot P_{A413} \quad (1)$$

Крок 4. Підрахунок балів за варіантами

В аналізованому прикладі ми оцінюємо ту чи іншу методику формування компетентностей. Кращою буде та методика, яка набере більшу оцінку.

Порівняння зважених балів, приписаних різних варіантів, робить можливим ранжування останніх. У технічних і соціальних системах, особливо при побудові оптимальних діагностичних процесів, широко використовується теорія інформації. Математична теорія зв'язку виникла в працях Вінера і Шеннона, та отримала застосування і в інших областях науки як загальна теорія зв'язку статистичних систем.

Відповідно до цієї теорії інформація - це різниця ентропії системи до і після отримання інформації

$$\Delta I = H_0(A) - H_1(A) \quad (2),$$

де $H_0(A)$ - початкова ентропія системи; $H_1(A)$ - ентропія системи після отримання інформації.

Інформаційна ентропія, як відомо, оцінюється за формулою

$$H(A) = -\sum P_i \ln P_i \quad (3),$$

де P - ймовірність знаходження системи в i -му стані.

Як приклад розрахуємо можливий рівень ентропії для першої теми «Параметри якості друкованої продукції» (табл. 6). У разі успішного оволодіння студентом всіма вміннями значення ентропії складе $H_0(A) = 7,39$. При успішному засвоєнні інших тем, рівень ентропії буде знижуватися і в ідеалі прагнути до нуля.

Таблиця 6

Розрахунок рівня ентропії

		Параметри якості друкованої продукції		
		P_i	$\ln P_i$	Σ
А	вміння вирішувати технічні завдання	0,25	-1,39	-1,14
Б	вміння працювати з технічною літературою і довідниками	0,20	-1,61	-1,41
С	вміння бачити завдання	0,30	-1,20	-0,90
Д	вміння пояснити технічне завдання	0,15	-1,89	-1,74
Е	вміння планувати роботу	0,10	-2,30	-2,20
				-7,39

Таким чином, з ростом компетентностей ентропія буде зменшуватися. Дійсно, для сумнівної події ($P > 0$) ентропія буде прагнути до нескінченності ($H(A) \rightarrow \infty$). Для достовірної події ($P \rightarrow 1$) ентропія буде прагнути до нуля ($H(A) \rightarrow 0$).

Знаючи стан системи в початковому стані і в поточному стані можна визначити прогрес або регрес системи. Даний підхід дозволяє оцінювати методику яка застосовується для формування компетентностей студентів та оперативно коригувати її.

Висновки

В результаті проведеної роботи розроблені рекомендації по діагностиці рівнів підготовленості студентів до науково-технічної діяльності, відбору змісту курсової підготовки, розкрито методику проведення занять, що сприяють стимулюванню у учнів позитивних мотивів до науково-технічної діяльності, формуванню у них системного підходу до технічної творчості.

Література

1. Беспалько В.П. Основы теории педагогических систем. - Воронеж: ВГУ, 2007 - 304 с
2. Королев Ф.Ф. Системный подход и возможности его применения в педагогике // Проблемы теории воспитания./Ред. Л.П.Будева, Л.И.Новикова, Г.Н.Филонов. - М.: Педагогика, 2004. - 260 с. - С.209-222.
3. Кузьмина Н.В. Системный подход в педагогических исследованиях //Методология педагогических исследований /ред. А.И.Пискунов, Г.В.Воробьев. - М.: НИИ ОП АПН СССР, 1980. - 165 с. - С.82-117
4. Методология системного подхода в педагогике /А.М.Сидоркин //Обзор.инф.НИИ ОП АПН СССР. Сер. "Обзоры по информационному обеспечению педагогических программ и основным направлениям развития педагогической науки и практики" Вып.3(33) - М.: НИИ ОП АПН СССР, 1989. - 56 с
5. Саранов А.М. Системный подход в исследованиях учебно-воспитательного процесса средней школы (на примере системы воспитательной работы классного руководителя). Автореферат дисс. ... к.п.н. - Казань, 1985. - 24 с.
6. Бим И.П. Система обучения иностранным языкам в средней школе и учебник как модель ее реализации. Автореферат дисс. ... д.п.н. - М., 1976. - 48 с.

ДОДАТОК

ВІДОМОСТІ ПРО АВТОРІВ ТА АНОТАЦІЇ

ГЛАВА 1

ІНТЕГРУВАННЯ ЧАТ-БОТА @ES_ECONOMY_KARKAS_BOT З ЕКСПЕРТНОЮ СИСТЕМОЮ ДЛЯ ВИЗНАЧЕННЯ КЛАСУ КРЕДИТОСПРОМОЖНОСТІ ПОЗИЧАЛЬНИКА

В. П. Бурдаєв

Анотація. В роботі представлені результати інтегрування чат-бота @es_economy_karkas_bot з прототипами експертних систем для прийняття рішення в режимі онлайн. Аналізується структура бази знань прототипу експертної системи для визначення класу кредитоспроможності позичальника. Наведено приклад онлайн консультації чат-бота в фінансовій предметної області на прикладі визначення оцінки кредитоспроможності позичальника.

Ключові слова: чат-бот, повідомлення, база знань, експертна система, клас кредитоспроможності позичальник.

Abstract. The paper presents the results of integrating the @es_economy_karkas_bot chatbot with prototypes of expert systems for making decisions online. The structure of the knowledge base of the prototype of the expert system is analyzed to determine the creditworthiness class of the borrower. An example of an online chatbot consultation in the financial subject area is given on the example of determining a borrower's creditworthiness assessment.

Keywords: chatbot, message, knowledge base, expert system, creditworthiness class borrower.

ГЛАВА 2

МУЛЬТИАГЕНТНА СИСТЕМА РОЗПОДІЛЕННІ ТОРГОВОЇ ФІРМИ

Н.Г. Аксак, М.В. Кушнар'єв

Анотація. Розроблено багатоагентну систему для планування та координації групових дій агентів, проведення переговорів та вибору оптимального рішення. Система реалізована на основі чотирьох рівневої архітектури, що дозволяє ефективно обробляти запити і керувати продажем товарів в різних торгових точках компанії. Реалізовані основні функції агентів продавців, які полягають в продажу свого товару з каталогу, і агентів покупців - придбати товари за дорученням своїх власників за

найменшою ціною. Імітаційне моделювання підтверджує доцільність застосування розробленої мультиагентної системи розподіленої фірми.

Ключові слова: мультиагентна система, аукціони, агенти, переговорні механізми, телемаркетинг, JADE

Abstract. A multi-agent system has been developed for planning and coordinating group actions of agents, conducting negotiations and choosing the optimal solution. The system is implemented on the basis of a four-tier architecture, which allows you to efficiently process requests and manage the sale of goods in various outlets of the company. The main functions of agents-sellers, who sell their goods according to the catalog, and the functions of agents-buyers - to buy goods on instruct their owners at the lowest price were realized. Simulation modeling confirms the feasibility of using the developed multi-agent system of a distributed firm.

Keywords: multi-agent systems, auctions, agent, negotiation mechanism, telemarketing, JADE.

ГЛАВА 3

СТВОРЕННЯ ПРИСТРОЇВ КОНТРОЛЮ І ЛОКАЛІЗАЦІЇ ПОМИЛОК В РОЗПОДІЛЕНИХ СИСТЕМАХ

М. Ю. Лосєв

Анотація. В роботі виконана формалізація процесу обробки інформації на основі методики синтезу багатоканальних пристроїв тестового контролю з локалізацією помилок у вхідних інформаційних послідовностях. Запропонована архітектура сигнатурного аналізатора, що дозволяє контролювати і діагностувати дані по групах. Наведені приклади створення пристроїв для різних умов використання в розподілених системах.

Ключові слова: процес обробки інформації, інформаційна послідовність, розподілена система, сигнатурний аналіз, матриця станів, характеристичний поліном.

Abstract. The paper formalizes the process of information processing based on the method of synthesis of multichannel test control devices with localization of errors in the input information sequences. The architecture of the signature analyzer which allows to control and diagnose the data on groups is offered. Examples of creating devices for different conditions of use in distributed systems are given.

Keywords: information processing process, information sequence, distributed system, signature analysis, state matrix, characteristic polynomial.

ГЛАВА 4

РЕГУЛЯРИЗОВАНИЙ РЕКУРЕНТНИЙ МЕТОД ПОТОЧНОГО РЕГРЕСІЙНОГО АНАЛІЗУ ДЛЯ ПОБУДОВИ ЛІНІЙНИХ МОДЕЛЕЙ

О.Г. Руденко, О.О. Безсонов

***Анотація.** Розглянуто задачу побудови рекурентної форми алгоритму поточного регресійного аналізу, що дозволяє здійснювати оцінювання невідомих параметрів при наявності стаціонарних завад. Отримано основні співвідношення, які описують процеси накопичення нової і скидання застарілої інформації. Показано, що алгоритми, які розглядаються, використовують допоміжну оцінку. Наведено результати імітаційного моделювання, які свідчать про ефективність підходу, що розвивається.*

***Ключові слова:** оцінювання, рекурентний алгоритм, регуляризація, завада, включення інформації, скидання інформації, імітаційне моделювання*

***Abstract.** The problem of constructing a recurrent form of the current regression analysis algorithm is considered, which allows estimating unknown parameters in the presence of stationary interferences. The basic relations describing the processes of accumulation of new and dumping of obsolete information are obtained. It is shown that the algorithms under consideration use auxiliary estimation. The results of simulation modeling are presented, which testify to the effectiveness of the developing approach.*

***Keywords:** estimation, recurrent algorithm, regularization, noise, information inclusion, information reset, simulation modeling*

ГЛАВА 5

КЛАСИФІКАЦІЯ ОБ'ЄКТІВ НА ОСНОВІ МОДЕЛІ ДЕРЕВОВИДНОЇ ШТУЧНОЇ ІМУННОЇ МЕРЕЖІ

М. М. Корабльов, О. О. Фомічов

***Анотація.** У цій роботі описано рішення проблеми класифікації об'єктів у багатовимірному просторі атрибутів на основі модифікованої моделі штучної імунної мережі з використанням принципів мінімального покриваючого дерева (MST – minimum spanning tree). При класифікації з неконтрольованим навчанням (кластеризації) на різних етапах реалізації штучної імунної мережі об'єкти використовуються як антигени та антитіла для формування навчальної вибірки (класифікатора). У випадку*

класифікації з контрольованим навчанням об'єкти з навчального набору використовуються як набір антигенів, а об'єкти класифікації – як набір антитіл. Визначення класу для кожного об'єкта базується на значенні авідності, яке описує силу кооперативної афінної взаємодії антитіл з антигеном. Використання запропонованої моделі дозволяє прискорити процес класифікації порівняно з моделями на основі методів MST та C-means, а також автоматизувати процес визначення кількості класів за відсутності навчальної вибірки.

Ключові слова: класифікація, кластеризація, імунне навчання, авідність, валентність, деревоподібна імунна мережа.

Abstract. This paper describes the solution of the object classification problem in a multidimensional attribute space based on a modified artificial immune network model using the principles of a minimum spanning tree (MST). During the classification with unsupervised learning (clustering) at different execution stages of the immune network, objects are used as antigens and antibodies to form a training sample (classifier). In the case of classification with supervised learning, objects from a training set are used as a set of antigens, and objects to classification are used as a set of antibodies. The class definition for each object is based on the avidity value, which describes the strength of cooperative affinity interactions of antibodies with antigen. Using the proposed model allows to speed up the classification process in comparison with models based on the MST and C-means methods, as well as automating the process of determining the number of classes in the absence of a training sample.

Keywords: classification, clustering, immune learning, avidity, valence, tree-like immune network.

ГЛАВА 6

ДОСЛІДЖЕННЯ ПОСИЛАЛЬНИХ МЕТОДІВ РАНЖУВАННЯ ДОКУМЕНТІВ ЗА РЕЗУЛЬТАТАМИ ІНФОРМАЦІЙНОГО ПОШУКУ

С. Г. Удовенко, Л. Е. Чала

Анотація. Досліджено посилальні методи ранжування документів за запитами користувачів PageRank, HITS, а також запропоновано багатоетапний метод ранжування. Для ітераційного обчислення показників авторитетності і посередницької оцінки використовувалися додатково розроблені функції. Багатоетапний метод ранжування передбачає проведення декількох послідовних етапів ранжування, використовуючи при цьому на кожному етапі певний запит. Даний метод дозволяє послідовно уточнювати список отриманих результатів.

Ключові слова: методи ранжування, інформаційний пошук, оцінювання релевантності, запит користувача, імітаційне моделювання

Abstract. Reference methods of ranking documents for users request PageRank and HITS are studied, and a multi-stage ranking method is proposed. Additionally developed functions were used for iterative calculation of authority indicators and mediation assessment. The multi-stage ranking method involves several consecutive stages of ranking, using at each stage a specific request. This method allows you to consistently refine the list of results.

Keywords: ranking methods, information search, relevance evaluation, user request, simulation

ГЛАВА 7

МЕТОДОЛОГІЯ SAFE ТА ЇЇ ВИКОРИСТАННЯ ДЛЯ ВПРОВАДЖЕННЯ ВИКОНАННЯ ЦІЛЕЙ КІБЕРБЕЗПЕКИ У ГНУЧКОМУ ПІДХОДІ ДО РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Федорченко В.М., Гапон А.О., Поляков А.О.

Анотація. Процес вибору методології розробки програмного забезпечення є дуже важливим на початковій стадії проектування продукту. Розробка ПО, як і будь-яка інша технічна дисципліна, має справу з такими основними проблемами: якість, вартість і надійність. У зв'язку з цим правильна організація процесу розробки програмного забезпечення є основою досягнення кращого результату в очікувані терміни, з хорошим рівнем якості, безпеки і з запланованим бюджетом.

Ключові слова: SAFe, модель загроз, Agile Release Train, функції, портфоліо, епік, GDPR, PIA, DPIA, PbDD, масштабування, спринт, Scrum.

Abstract. The process of choosing a methodology for the development of software security is even more important at the cob stage of product design. Software development, as all other technical disciplines can help with the following main problems: quality, efficiency, reliability, safety and to do it with a planned budget.

Keywords: SAFe, Threat Model, Agile Release Train, Functions, Portfolio, EPIK, GDPR, PIA, DPIA, PbDD, scaling, sprint, scrum.

ГЛАВА 8

ПІДХОДИ ДО ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

І. О. Ушакова

Анотація Метою роботи є аналіз проблем забезпечення якості та тестування програмного забезпечення в процесі його створення в ІТ-компаніях. Об'єктом дослідження є процеси тестування, контролю якості та забезпечення якості. Предметом дослідження є характеристики та моделі оцінювання якості та форми організації роботи команд QA і QC. В процесів дослідження був використаний системний підхід, порівняльний аналіз методів оцінювання якості та форм організації роботи з контролю та забезпечення якості програмних продуктів. Для оцінювання якості програмного забезпечення була запропонована класифікація цих методів, розглянуті методи інтегрального оцінювання якості, заснованого на витратах, і метод оцінки якості на основі ієрархічної моделі. Для оцінювання відмінностей між контролем якості та забезпеченням якості був проведений аналіз їх обов'язків, планування роботи та документації, що дало можливість порівняти виконувані функції і умови роботи. Так QC підтверджує, що конкретні результати відповідають стандартам і специфікаціям, а QA є ширшою функцією, яка охоплює планування і контроль на протязі всього життєвого циклу розроблення. Тестування ж є невід'ємною частиною контролю якості. Для того, щоб в ІТ-компанії процеси управління якістю були ефективними команди QA і QC повинні співпрацювати. Наукова новизна полягає у розробленні методичної основи оцінювання якості програмного забезпечення, розробленні рекомендацій щодо вдосконалення організації роботи команд QA і QC в ІТ-компанії.

Ключові слова: програмне забезпечення, тестування, контроль якості, забезпечення якості, модель, метрики, організація роботи.

Abstract. The aim of the work is to analyze the problems of quality assurance and software testing in the process of its creation in IT companies. The object of research is the processes of testing, quality control and quality assurance. The subject of the research is the approaches, characteristics and models of quality assessment and forms of organization of QA and QC teams. A classification of these methods has been proposed to assess software quality. The methods of integrated quality assessment based on costs and the method of quality assessment based on a hierarchical model were considered. A systematic approach, comparative analysis of quality assessment methods and forms of organization of work on quality control and quality assurance of software products was used in the research process. A

classification of these methods has been proposed to assess software quality. Methods of integrated quality assessment based on costs, and the method of quality assessment based on a hierarchical model were considered. The scientific novelty is to develop a methodological basis for assessing the quality of software, developing recommendations for improving the organization of QA and QC teams in an IT company.

Keywords: *software, testing, quality control, quality assurance, model, metric, organization of work.*

ГЛАВА 9

МОДЕЛЮВАННЯ ВІДТВОРЕННЯ ПЛОСКИХ КРИВИХ ЛАМАНИМИ ЗА АСИМПТОТИЧНО - ОПТИМАЛЬНИМ АЛГОРИТМОМ ІНТЕРПОЛЯЦІЇ

О. В. Фролов

Анотація. *Розглянуто моделювання апроксимації плоских параметричних кривих ламаними за асимптотично оптимальним алгоритмом інтерполяції в хаусдорфовій метриці на основі обчислення похибок апроксимації, досліджено вплив кількісної характеристики ступеню дискретизації інтегральної функції - регулятора вузлів інтерполяції на показники розподілу похибок, а також оптимізація параметру функції – регулятора вузлів для випадку кривих з наявними точками перегину.*

Ключові слова: *інтерполяція, ламана, еквідистанта, інтегрування, плоска параметрична крива, похибка, точка перегину.*

Abstract. *The modeling of piecewise linear approximation of a flat parametric curve by asymptotically optimal interpolation algorithm in Hausdorff's metric based on calculation of approximation errors is considered. The algorithm is based on the numerical integration of the distribution function and linear interpolation of its values. The influence of the primary discretization of the regulator function on the distribution characteristics of the polyline errors was considered.*

Keywords: *interpolation, polyline, optimization, flat parametric curve, error, inflection point, lost function.*

ГЛАВА 10

РОЗРОБКА ПЕДАГОГІЧНОЇ МОДЕЛІ ФОРМУВАННЯ КОМПЕТЕНЦІЙ ФАХІВЦІВ ПОЛІГРАФІЧНОГО ПРОФІЛЮ З ПОЗИЦІЇ СИСТЕМНОГО ПІДХОДУ

А. С. Гордєєв

Анотація. В розділі запропоновано методуку по діагностиці рівнів підготовленості студентів до науково-технічної діяльності, відбору змісту курсової підготовки, розкрито методуку проведення занять, що сприяють стимулюванню у учнів позитивних мотивів до науково-технічної діяльності, формуванню у них системного підходу до технічної творчості. Метою роботи є розробка системної моделі формування компетентностей у фахівців поліграфічного профілю. Визначено, з ростом компетентностей ентропія буде зменшуватися. Дійсно, для сумнівної події ентропія буде прагнути до нескінченності. Для достовірної події ентропія буде прагнути до нуля. Знаючи стан системи в початковому стані і в поточному стані можна визначити прогрес або регрес системи. Даний підхід дозволяє оцінювати методуку яка застосовується для формування компетентностей студентів та оперативно коригувати її.

Ключові слова: компетентності, системний підхід, поліграфія, технічна інформація, епістемологічні рівні.

Abstract. The section offers methods for diagnosing the levels of preparedness of students for scientific and technical activities, selection of the content of course preparation, reveals the methods of conducting classes that stimulate students' positive motives for scientific and technical activities, forming a systematic approach to technical creativity. The aim of the work is to develop a system model of competence formation in printing specialists. It is determined that the entropy will decrease with increasing competencies. Indeed, for a dubious event, entropy will tend to infinity. For a reliable event, the entropy will tend to zero. Knowing the state of the system in the initial state and in the current state, you can determine the progress or regression of the system. This approach allows you to assess the methodology used to form the competencies of students and quickly adjust it.

Keywords: competencies, system approach, polygraphy, technical information, epistemological levels.

НАУКОВЕ ВИДАННЯ

СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ І СИСТЕМИ

Монографія

під ред. д.е.н., проф. Пономаренка В. С.

Відповідальний за випуск **Бурдаєв В. П.**

Відповідальний редактор **Грабовський Є. М.**

Підп. до друку 1.06.2021 Формат 60 × 90 1/16. Папір. Друк офсетний.

Ум.-друк. арк. 13,25 Обл.-вид. арк. Тираж 150 екз. Зам. №

Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи

Видавець і виготівник – ПП «Стиль-іздат»