

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерних наук \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_ Програмної інженерії \_\_\_\_\_  
(повна назва)

**АТЕСТАЦІЙНА РОБОТА**  
**Пояснювальна записка**

\_\_\_\_\_ другий (магістерський) \_\_\_\_\_  
(рівень вищої освіти)

**ДОСЛІДЖЕННЯ МЕТОДІВ АДАПТИВНОГО ТЕСТУВАННЯ**  
**ЗНАНЬ В ДИСТАНЦІЙНІЙ ОСВІТІ**

(тема)

Виконав: студент 2 курсу, групи ІПЗм-18-4  
спеціальності 121 – Інженерія програмного  
забезпечення

(код і повна назва спеціальності)

освітньо-наукової програми Інженерія  
програмного забезпечення

(повна назва освітньої програми)

\_\_\_\_\_ Ляшик В.А. \_\_\_\_\_  
(прізвище, ініціали)

Керівник \_\_\_\_\_ проф. Шостак І.В. \_\_\_\_\_  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. \_\_\_\_\_

З.В.Дудар

2020 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення  
(код і повна назва)

Освітньо-наукова програма Інженерія програмного забезпечення  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

## ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

Студентові Ляшику Володимир Андрійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів адаптивного тестування знань в  
дистанційній освіті

затверджена наказом по університету від « \_\_\_\_\_ » \_\_\_\_\_ 2020 р № \_\_\_\_\_

2. Термін подання студентом роботи до екзаменаційної комісії «11» травня 2020 р.

3. Вихідні дані до роботи Алгоритми обробки великих обсягів даних,  
алгоритми моделювання Марківських мереж. Використовувати ОС Windows,  
середовище об'єктно-орієнтованого проектування.

4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз  
проблемної галузі і постановка задачі, методи пошуку корисних даних, опис  
об'єктних моделей, використовувані методи та алгоритми, архітектура  
програмної системи, опис розробленої програмної системи, результати  
тестування програмної системи

## 5. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	проф. Шостак І.В.		

**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз предметної галузі	25 березня 2020 р.	
2.	Огляд існуючих методів	31 березня 2020 р.	
3.	Методи створення та аналізу марківських мереж	15 квітня 2020 р.	
4.	Підготовка пояснювальної записки	20 квітня 2020 р.	
5.	Спецчастина	28 квітня 2020 р.	
6.	Підготовка презентації та доповіді	03 травня 2020 р.	
7.	Попередній захист	05 травня 2020 р.	
8.	Нормоконтроль, рецензування	07 травня 2020 р.	
9.	Занесення роботи в електронний архів	08 травня 2020 р.	
10.	Допуск до захисту в зав. кафедрі	10 травня 2020 р.	

Дата видачі завдання \_ « \_\_\_\_\_ » \_ \_\_\_\_\_ 2020 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ проф. Шостак І.В.  
(підпис) (посада, прізвище, ініціали)

**РЕФЕРАТ ABSTRACT**

Пояснювальна записка містить: 110 с., 23 табл., 45 рис., 3 дод., 32 джерела.

АДАПТИВНЕ ТЕСТУВАННЯ, МЕРЕЖІ МАРКОВА, МОДЕЛЬ РАША,  
ПАРАМЕТРИЧНА СТАТИСТИКА, АПРОКСИМАЦІЯ ФУНКЦІЙ,  
КРИТЕРІАЛЬНІ ОЦІНКИ

Об'єкт дослідження – процес контролю знань у дистанційній формі навчання.

Мета роботи – підвищення ефективності контролю знань навчаємих у дистанційній формі навчання за рахунок застосування адаптивних методів комп'ютерного тестування.

Методи дослідження: методи булевої логіки; методи теорії графів; мережі Маркова та Петрі.

В результаті удосконалено алгоритми оптимального багатоступінчастого адаптивного тестування та розроблений прототип програмного забезпечення.

ADAPTIVE TESTING, MARKOV NETS, RASH MODEL, PARAMETRIC  
STATISTICS, STOCHASTIC METHODS, APPROXIMATION FUNCTIONS,  
CRITERIAL ESTIMATES

The object of research is the process of knowledge control in the form of distance learning.

The subject of the research is adaptive methods for controlling knowledge of educable.

Aim: to increase the effectiveness of knowledge of educable control in the distance form of education through the use of adaptive computer-testing methods.

Research methods: Boolean logic methods; graph theory methods; Markov and Petri nets; methods of mathematical statistics; functional and system testing methods for verifying the automated adaptive testing model.

## ЗМІСТ

Вступ .....	7
1 Аналіз стану вирішення проблеми і постановка задач дослідження.....	10
1.1 Стан дистанційної освіти в Україні .....	10
1.2 Форми представлення комп'ютерних тестових завдань .....	13
1.3 Аналіз методів реалізації тестових систем .....	14
1.4 Поняття адаптивного тестування й принципи його реалізації .....	16
1.5 Методи об'єктивної оцінки підготовленості випробуваного .....	19
1.6 Постановка задач дослідження .....	23
2 Опис проведених теоретичних досліджень .....	26
2.1 Аналіз моделей навчального курсу .....	26
2.2 Модель системи адаптивного керування .....	28
2.3 Модель суб'єкта навчання .....	32
2.4 Огляд існуючих методів адаптивного тестування .....	35
3 Опис проектування алгоритмів системи.....	40
3.1 Моделювання адаптивного тестування .....	40
3.2 Опис методу роботи системи адаптивного тестування .....	45
3.3 Модифікований алгоритм адаптивного тестування .....	48
3.4 Алгоритм керування станами мережі питань .....	52
4 Опис розробленого програмного забезпечення.....	55
4.1 Основні методи реалізації алгоритмів .....	55
4.2 Моделювання рівномірного розподілу питань .....	56
4.3 Моделювання переходів у Марківських мережах .....	58
4.4 Реалізація модифікованого алгоритму вибору тестових завдань .....	60
4.5 Алгоритм опитування .....	62
4.6 Проектування прототипу ПЗ .....	64
4.7 Архітектура розробленого ПЗ .....	68
5 Опис можливості використання отриманих результатів.....	75

	6
Висновки .....	79
Перелік джерел посилання .....	81
Додаток А Програмний код .....	84
Додаток Б Слайди презентації .....	94
Додаток В Апробація результатів роботи.....	109

## ВСТУП

Наразі серед проблем дистанційної освіти [1] особливе місце займає проблема автоматизації контролю знань. Питання комп'ютерного контролю становлять великий інтерес для викладачів вузів і творців засобів реалізації такого контролю. Тим часом, питання комп'ютерного контролю недостатньо широко освітлені в теоретичному плані, і інтерес до них зазвичай реалізується в більшості випадків шляхом створення чергової програми комп'ютерного контролю із задалегідь складеним набором контрольних завдань. Засоби адаптивного тестування недостатньо освітлені й пророблені недостатньо прозоро.

Контроль знань або тестування – це процес, проведений з метою визначення рівня знань [1]. Це найбільш стандартизований і об'єктивний метод контролю й оцінювання знань, умінь і навичок випробуваного, який позбавлений таких традиційних недоліків інших методів контролю знань, як неоднорідність вимог, суб'єктивність екзаменаторів, невизначеність системи оцінок і т.п. Рівні знань зазвичай дискретизуються. При такому підході, тестування може розглядатися як деякий діагностичний процес, а стани, що характеризують оцінки знань випробуваного, — як діагностичні стани. Тести є ефективним засобом перевірки якості знань, одержуваних студентами, і оперативного контролю ходу навчання [2]. Інформаційні освітні ресурси (ІОР) тестові матеріали, що містять, можна розбити на дві категорії:

- орієнтовані на проходження студентами тестів у писемній формі з подальшою перевіркою вручну викладачем ( як варіант – сканування результатів тестування з метою їх подальшої автоматизованої перевірки);

- системи комп'ютерного тестування (СКС) з відповідним наповненням тестовими матеріалами.

Переваги другої категорії тестових ІОР очевидні.

- вони дозволяють звільнити викладача від рутинної роботи при проведенні іспитів і проміжній оцінці знань у традиційному навчальному процесі, а при

навчанні з використанням дистанційних технологій стають основним засобом контролю;

- надають можливість автоматизації обробки результатів, об'єктивність контролю й швидкість перевірки якості підготовки великого числа тестуємих по широкому колу питань. Це дозволяє визначити розділи, які представляють найбільшу складність у вивченні, і, можливо, коректувати процес навчання залежно від результатів тестування;

- надають можливість реалізації навчальної функції;

- дозволяють зробити індивідуалізацію процесу засвоєння знань учнями.

Функції контролю знань носять не тільки контролюючий характер, але також навчально-виховний і розвиваючий.

Форми тестових завдань можна представити в наступних формах:

- закрита форма;

- відкрита форма;

- на відповідність;

- на встановлення правильної послідовності.

Проведений аналіз існуючих КСТ показав, що вони в більшості випадків орієнтовані на проведення тестів, а не на їхню розробку. При реалізації тестування жодна з розглянутих КСТ не підтримує адаптивні методи проведення тестів, слабо розвинена політонічна оцінка [3] виконання тестових завдань.

Тестування є важливим елементом не тільки контролю знань, але й навчання. При навчальній тестуванні користувачеві після проходження тесту надаються посилання на ті розділи навчального матеріалу, на запитання по яких він відповів невірно.

Розглядаючи комп'ютерні системи контролю знань із погляду їх практичної реалізації, необхідно відзначити, що всі вони містять у собі наступні компоненти (іноді не виділені явно в структурі системи) [4].

- підсистему підготовки тестів;

- підсистему проведення тестування;

- підсистему аналізу результатів тестування.

Включення сучасних інформаційних технологій в освітній процес створює реальні можливості підвищення якості утвору. Однак слід визнати, що рівень інформатизації навчальної й наукової діяльності залишається ще досить низьким, недостатньо розроблена нормативно-правова база для дистанційного навчання.

Таким чином, мета атестаційної роботи магістра – підвищення ефективності контролю знань тих, яких навчають, у дистанційній формі навчання за рахунок застосування адаптивних методів комп'ютерного тестування.

Для досягнення мети необхідно розв'язати наступні завдання.

- провести аналіз сучасного стану проблеми контролю знань у дистанційному утворі;

- провести огляд і аналіз існуючих систем комп'ютерного тестування;

- розробити адаптивний метод контролю знань, тих, кого навчають, у дистанційній формі навчання;

- розробити прототип програмного забезпечення для адаптивного тестування в дистанційному навчанні.

# 1 АНАЛІЗ СТАНУ ВИРІШЕННЯ ПРОБЛЕМИ І ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

## 1.1 Стан дистанційної освіти в Україні

На сьогоднішній день не відповідає вимогам до інформаційного суспільства, яке прагне інтегруватися в європейське й світове співтовариство [3].

По-перше, Україна відстає від розвинених країн у застосуванні технологій дистанційного навчання при підготовці, перепідготовці й підвищенні кваліфікації фахівців різних областей і рівнів.

По-друге, має місце істотне відставання телекомунікаційних мереж передачі даних, які відзначаються недостатньою пропускнуою здатністю, надійністю зв'язки і її низькою якістю.

Світовий процес переходу від індустріального до інформаційного суспільства, а також соціально-економічні зміни, які відбуваються в Україні, вимагають істотних змін у багатьох сферах діяльності держави. У першу чергу це стосується реформування освіти. Національною програмою «Освіта. Україна XXI вік» передбачений розвиток освіти на основі нових прогресивних концепцій, впровадження в навчально-виховний процес новітніх педагогічних технологій і науково-методичних досягнень, створення нової системи інформаційного забезпечення утвору, входження України в трансконтинентальну систему комп'ютерної інформації [4].

Розвиток освітньої системи в Україні повинен привести до:

- появи нових можливостей для відновлення змісту навчання й методів викладання дисциплін і поширення знань;
- розширенню доступу до всіх рівнів утвору, реалізації можливості його одержання для великої кількості молодих людей, включаючи тих, хто не може вчитися у вищих навчальних закладах із традиційними формами через відсутність фінансових або фізичних можливостей, професійної зайнятості, віддаленості від великих міст, престижних навчальних закладів і т.п.; реалізації системи

безперервної освіти " через усе життя", включаючи середню, довузівську, вищу й післядипломну; індивідуалізації навчання при масовості утвору.

Для досягнення зазначених результатів необхідно швидкими темпами розбудовувати дистанційну форму навчання, впровадження якої в Україні передбачене Національною програмою інформатизації.

Контролююча функція [5] полягає у виявленні стану знань і вмінь учнів, рівня їх розвитку, у вивченні ступеня засвоєння приймань пізнавальної діяльності, навичок раціональної навчальної праці.

За допомогою контролю визначається вихідний рівень для подальшого оволодіння знаннями, уміннями й навичками, вивчається глибина й обсяг їх засвоєння. Рівняється плановане з дійсними результатами, установлюється ефективність використовуваних викладачем методів, форм і засобів навчання.

Навчальна функція [6] контролю полягає в удосконалюванні знань і вмінь, їх систематизації. У процесі перевірки учні повторюють і закріплюють вивчений матеріал. Вони не тільки відтворюють раніше вивчене, але й застосовують знання й уміння в новій ситуації. Перевірка допомагає виділити головне, основне в досліджуваному матеріалі, зробити, що перевіряються знання й уміння більш ясними й точними. Контроль сприяє також узагальненню й систематизації знань.

Сутність діагностичної функції контролю [7] – в одержанні інформації про помилки, недоліки й пробілах у знаннях і вміннях учнів причинах, що їх породжують, утруднень учнів в оволодінні навчальним матеріалом, про число помилок. Результати діагностичних перевірок допомагають вибрати найбільш інтенсивну методику навчання, а також уточнити напрямок подальшого вдосконалювання змісту методів і засобів навчання.

Прогностична функція [8] перевірки служить одержанню випереджальної інформації про навчально-виховний процес. У результаті перевірки знань одержують підстави для прогнозу про хід певного відрізка навчального процесу: чи досить сформовані конкретні знання, уміння й навички для засвоєння наступної порції навчального матеріалу (роздягнула, теми).

Результати прогнозу використовують для створення моделі подальшої поведінки, що вчиться, допускає сьогодні помилки даного типу або, що має певні пробіли в системі приймань пізнавальної діяльності.

Прогноз допомагає одержати вірні висновки для подальшого планування й здійснення навчального процесу.

Розвиваюча функція [6] контролю знань полягає в стимулюванні пізнавальної активності учнів, у розвитку їх творчих здібностей. Контроль має виняткові можливості в розвитку учнів. У процесі контролю розвиваються мова, пам'ять, увага, уява, воля й мислення школярів. Контроль дуже впливає на розвиток і прояв таких якостей особистості, як здатності, схильності, інтереси, потреби.

Сутність функції контролю, що орієнтує, [6] – в одержанні інформації про ступінь досягнення мети навчання окремим учнем і групою в цілому – наскільки засвоєний і як глибоко вивчений навчальний матеріал. Контроль орієнтує учнів у їхніх утрудненнях і досягненнях.

Розкриваючи пробіли, помилки й недоліки учнів, він указує їм напрямки додавання сил по вдосконалюванню знань і вмінь. Контроль допомагає учневі краще довідатися самого себе, оцінити свої знання й можливості.

Функція контролю, що виховує, полягає у вихованні в учнів відповідального відношення до навчання, дисципліни, акуратності, чесності. Перевірка спонукує більш серйозно й регулярно контролювати себе при виконанні завдань. Вона є умовою виховання твердої волі, наполегливості, звички до регулярної праці .

Виділення функції контролю підкреслює його роль і значення в процесі навчання. У навчальному процесі самі функції проявляються в різному ступені й різних комбінаціях. Реалізація виділених функцій на практиці робить контроль більш ефективним, а також ефективніше стає й сам навчальний процес [7].

## 1.2 Форми представлення комп'ютерних тестових завдань

Існують наступні форми тестових завдань: закрита форма, відкрита форма, на відповідність та на встановлення правильної послідовності [8]:

Тестове завдання закритої форми складається із проблемної частини, у якій формулюється завдання (проблема), відповідей, що містять правильну відповідь (еталон) і дистрактори – тобто невірна, неправильна відповідь у закритому тестовому завданні з вибірковими відповідями.

Проблемна частина тестового завдання може бути написана як у питальній, так і стверджувальній формах.

До закритих тестових завдань ставляться:

- завдання з альтернативним вибором (дихотомічні);
- завдання із множинним вибором: з одним негативним або позитивною правильною відповіддю; із множинними відповідями (вірно/невірно), розширеного вибору.

Тестові завдання відкритої форми являють собою сукупність завдань із вільно конструйованим відповіддю: завдання доповнення (з обмеженням на відповідь) і завдання вільного викладу(на відповідь не накладаються ніякі обмеження).

У завданнях з обмеженням на відповідь тестуємому необхідно вписати пропущене слово, символ, знак і т.ін. Прочерк ставиться на місце ключового терміна, значення якого перевіряється. Місце розташування прочерку може бути на початку, у середині й наприкінці пропозиції.

У відкритих тестових завданнях вільного викладу на відповідь не накладаються ніякі обмеження. У випадку комп'ютерного тестування складання таких завдань дуже трудомістко у зв'язку зі складністю формалізації відповідей.

Тестові завдання на відповідність – це сукупність завдань, у яких тестуємому пропонується відновити відповідність між елементами двох стовпців. Може Стовпець, з якого вибираються відповіді, повинен бути довше, чим перший

список, а якщо ні, то може мати місце випадкове вгадування. Завдання на відповідність застосовуються для виявлення актуальних знань і контролю вмінь знаходити спільність або відмінність у досліджуваних об'єктах, явищах, процесах. Основні труднощі в розробці такого виду завдань пов'язана з добором правдоподібних надлишкових елементів у правому стовпці.

Тестові завдання на встановлення послідовності – це сукупність тестових завдань, у проблемній частині яких у довільному порядку описані дії або процеси, пов'язані з певним завданням, для рішення якої тестуємому необхідно встановити необхідний порядок наведених дій і вказати його за допомогою цифр. За допомогою завдань на встановлення правильної послідовності з'ясовують знання й уміння по виконанню певних приписань, правил, законів. У якості елементів приводяться слова, які в певній послідовності дають правильне формулювання правила, закону й т.п. Схожими є завдання на ранжирування – розташування елементів по зростанню або убутанню їх значимості, властивостей, кількісних характеристик і т.п.

### 1.3 Аналіз методів реалізації тестових систем

Різні розроблювачі по-різному підходять уводити, увести до ладу вибору програм і технологій створення цих систем. Розглянуто особливості різних підходів у реалізації конкретних тестових систем.

Одним із частих підходів є використання систем клієнт-сервер, у яких як клієнта виступає браузер, а в якості сервера – зв'язка з Web – сервера й сервера додатків [5].

Характеристики:

- відсутня залежність від операційної системи;
- відсутня необхідність в установці й налаштування клієнтської частини;
- мова програмування HTML має більші можливості за поданням даних;

- відсутність проблем з передачею даних у мережі Internet (конфлікти із системами захисту типу Firewall виключені, тому що використовуються стандартні протоколи й порти доступу);

- укладач тестів під час уведення контрольних-тестових матеріалів повинен перебувати в мережі;

- мова HTML орієнтована на представлення даних, а не на їхнє створення, що сильно утрудняє редагування вже створених даних. Це є проблемою багатьох систем, що засновані на Web-інтерфейсі;

- відсутня можливість використання при підготовці тестових завдань технології OLE, що утрудняє вставку складних об'єктів (формул та ін.).

Альтернативою є створення спеціалізованого ПЗ для проведення тестування й підготовки тестових матеріалів. Даний підхід при підготовці тестів має наступні характеристики:

- автономність (не потрібна наявність мережного з'єднання);
- зручність створення й редагування тестових матеріалів.

Використання спеціалізованого ПЗ при проведенні тестування дозволяє:

- мінімізувати обсяг переданих даних;
- реалізувати складні інтерактивних тестові завдання.

Його недоліками є необхідність попередньої установки й налаштування додаткового ПЗ, а іноді й прив'язка до конкретної операційної системи.

- можливі обмеження по форматах представлення інформації.
- необхідність установки спеціального клієнтського додатка тестування.
- проблема відновлення версій клієнтів при вдосконалюванні ПЗ.

Існують наступні критерії якості тестів.

Традиційні вимоги до тестових матеріалів незалежно від форми їх представлення:

- валідність (змістовна й функціональна);
- надійність;

- однозначність – тобто однаковість оцінки якості виконання тесту різними експертами, коректна постановка питань, що не допускає їхнє різне тлумачення;
- здатність диференціювання;
- наявність засобів автоматизованого тестування.
- відповідність тестових матеріалів міжнародним стандартам (IMS QTI [7]), їх інтероперабельність.
- різноманітність типів тестових завдань;
- різноманітність способів побудови траєкторій тестування (тверда послідовність, випадкова вибірка завдань, адаптивне тестування);
- різноманітність режимів використання в тестуванні (навчальне тестування, тренувальне тестування, контрольне тестування);
- диференціація оцінки тестових завдань (можливість призначення різних балів різним питанням, облік частково правильних відповідей та ін.);
- можливість використання в тестах графічних ілюстрацій, анімації, аудіо й відео інформації;
- контроль часу (на тестове завдання або тест у цілому), відображення поточного стану тестування (кількість часу, що залишився, і тестових завдань та ін.);
- наявність інструментів аналізу результатів тестування й оцінки якості тестових матеріалів.

#### 1.4 Поняття адаптивного тестування й принципи його реалізації

При наявності великої кількості систем комп'ютерного тестування, постійнім створенні нових систем необхідно визначитися з основними критеріями, яким повинні відповідати такі системи. Попередній аналіз показує, що весь набір параметрів може бути зведений у наступні чотири основні групи.

- типи підтримуваних тестових завдань;
- сервіс по створенню тестових завдань;
- побудова траєкторії тестування;
- обробка результатів тестування;
- адміністративні функції.

Для порівняння були обрані відомі системи Прометей 4.0 [9], АСТ [10], Ellekta 5.01 [11], Test Commander 3.0 [12], Webquiz XP 2.0.34 [13]. Як показує аналіз, основні параметри провідних систем багато в чому схожі. Загальним недоліком усіх систем (у напрямку його подолання багато хто з них, імовірно, будуть розвиватися) є відсутність або обмеженість можливостей по обміну інформацією з міжнародним стандартом IMS QTI, а також відсутність засобів адаптивного тестування [14].

Під адаптивним тестовим контролем розуміють комп'ютеризовану систему науково обґрунтованої перевірки й оцінки результатів навчання, що має високу ефективність за рахунок оптимізації процедур генерації, пред'явлення й оцінки результатів виконання адаптивних тестів.

Ефективність контрольних-оцінних процедур підвищується при використанні багатокрокової стратегії відбору й пред'явлення завдань, заснованої на алгоритмах з повною контекстною залежністю, у яких черговий крок відбувається тільки після оцінки результатів виконання попереднього кроку (або кроків). Після виконання випробуваним чергового завдання щораз виникає потреба в ухваленні рішення про вибір рівня складності наступного завдання залежно від того, вірною або невірною була попередня відповідь.

Алгоритм підбору й пред'явлення завдань будується за принципом зворотного зв'язку, коли при правильній відповіді випробуваного чергове завдання вибирається більш важким, а невірна відповідь спричиняє пред'явлення наступного більш легкого завдання, ніж те, на яке випробуваним була дана невірна відповідь. Також є можливість завдання додаткових питань по темах, які той, якого навчають, знає не дуже добре для більш тонкого з'ясування рівня знань

у даних областях. Таким чином, можна сказати, що адаптивна модель нагадує викладача на іспиті – якщо той, якого навчають, відповідає на, що задаються питання впевнено й правильно, викладач досить швидко ставить йому позитивну оцінку. Якщо той, якого навчають, починає «плавати», то викладач задає йому додаткові або навідні запитання того ж рівня складності або по тій же темі. І, нарешті, якщо той, якого навчають, із самого початку відповідає погано, оцінку викладач теж ставить досить швидко, але негативну.

До неодмінних умов реалізації таких алгоритмів слід віднести:

- наявність банку каліброваних завдань зі стійкими оцінками їх параметрів, що дозволяють прогнозувати успіх або неуспіх випробуваного при доборі чергового завдання адаптивного тесту;

- використання інструментальних-програмно-інструментальних засобів і комп'ютерних програм для індивідуалізації алгоритмів добору завдань, заснованих, як правило, на оцінці ймовірності правильного виконання навчальних завдань;

- використання параметричних моделей Item Response Theory [15] (психометрична теорія завдання і відповідь).

Багатокрокові стратегії адаптивного тестування можна підрозділити на фіксоване, фіксовано-ветвящиеся й варьїрующе-ветвящиеся [15] варіюють від того, як конструюються багатокрокові адаптивні тести. Співставлення різних видів адаптивного тестування призводить до вибору комп'ютерного адаптивного тестування, заснованого на багатокрокових стратегіях, що варіюють, у якості основного підходу. Найбільш важлива перевага стратегій, що варіюють, пов'язане з можливістю оперативного реагування на результати виконання навчальних завдань шляхом переоцінки рівня підготовки того, якого навчають, після виконання кожного чергового завдання адаптивного тесту.

Тестування зазвичай починається із завдань середньої складності, але можна починати й з легких завдань, тобто йти за принципом підвищення складності.

Тестування закінчується, коли параметри функції успіху (по моделі Раша [15]) є статистично певними й дозволяють оцінити по тій або іншій шкалі рівень підготовки випробуваного.

Особливості адаптивного тестування:

- дозволяє більш гнучко й точно вимірювати знання тих, яких навчають;
- дозволяє вимірювати знання меншою кількістю завдань, чому в класичній моделі;
- виявляє теми, які той, якого навчають, знає погано й дозволяє задати по них низка додаткових питань;
- заздалегідь невідомо, скільки питань необхідно задати тому, якого навчають, щоб визначити його рівень знань. Якщо питань, закладених у систему тестування, виявляється недостатньо, можна перервати тестування й оцінювати результат по тій кількості питань, на яку відповів той, якого навчають;
- можливе застосування тільки на ПК.

Надійність результатів тестування в цьому випадку досить висока, тому що здійснюється пристосування під рівень знань конкретного того, якого навчають,, що забезпечує більш високу точність вимірів.

### 1.5 Методи об'єктивної оцінки підготовленості випробуваного

Побудувати адекватну модель тестування складно, тому що рівень підготовки ( як вимірюваний параметр) і труднощі тестових завдань ( як параметр завдання, встановлюваний при його атестації) мають різні метрики. Первинною інформацією при тестуванні знань є набраний бал випробуваних або так званий первинний бал, обумовлений у відсотках (або частках) як відношення кількості правильних до загальної кількості питань заданого рівня складності. Перевагою цієї оцінки є її простота й наочність.

Однак проблема полягає в тому, що первинний бал є не абсолютної, а відносною оцінкою. Він суттєво залежить від труднощів завдань тесту й на іншому тесті він може виявитися іншим, причому самі труднощі тесту у свою чергу визначається всім контингентом випробуваних. Бажане мати попередню оцінку рівня підготовленості випробуваних, підтверджувану на різних тестах, що мають заздалегідь певний рівень труднощів завдань.

Другим істотним недоліком первинних балів є їхня нелінійність стосовно тем параметрам, які вони повинні характеризувати (рівень підготовленості). Зокрема, якщо тест складається з 100 завдань, то різниця в первинних балах  $b_1 - b_2 = 86 - 82 = 4$  відповідає більшій відмінності в рівні підготовленості учасників, чому та ж різниця для учасників, що мають, наприклад 23 і 19 балів. Порівнюючи первинні бали необхідно розуміти, що первинні бали є лише індикатором підготовленості випробуваних, а не її заходом. Теорія тестових завдань (IRT) [15] дозволяє встановити зв'язок між рівнем знань випробуваних і результатами виконання тестів, що дозволяє визначити рівень знань незалежно від складності завдань. У якості заходу труднощі завдань і заходу рівня знань застосовується поняття «логіт» [15].

Г.Раш увів два поняття: «логіт рівня знань» і «логіт рівня труднощів завдання». Першу він визначив як натуральний логарифм відносини частки правильних відповідей випробуваного, на всі завдання тесту, до частки неправильних відповідей, а другу – як натуральний логарифм іншого відношення – частки неправильних відповідей на завдання тесту до частки правильних відповідей на те ж завдання, по безлічі випробуваних. Єдина логарифмічна шкала дозволяє встановити необхідна відповідність між рівнем підготовки й труднощами завдання й, більше того, зробити корекцію результатів тестування при тестах різної складності.

Технологія комп'ютеризованого тестування повинна мати наступні характеристики:

- наявність інтерактивного інструментального середовища;
- мультипредметное застосування;

- адекватне відбиття моделі, що конструюється, предметної області в процесі тестування;
- можливість вибору алгоритму тестування;
- можливість інтегруватися у різні утворювальні технології;
- можливість створення профілів і рівнів;
- масштабованість;
- доступність;
- дружність користувацького інтерфейсу;
- ведення бази тестових багаторівневих завдань;
- планування й керування;
- націленість на досягнення більш високих результатів і підвищення мотивації.

Адаптивне тестування визначається як «сукупність процесів генерації, пред'явлення й оцінки результатів виконання адаптивних тестів, що забезпечує приріст ефективності вимірів у порівнянні із традиційним тестуванням завдяки оптимізації добору характеристик завдань, їх кількості, послідовності й швидкості пред'явлення стосовно до особливостей підготовки тестуємих» [16].

При адаптивнім тестуванні в процесі проходження тесту (або набору тестів) будується модель того, якого навчають, яка використовується для генерації або вибору наступних завдань тестування залежно від рівня підготовки того, якого навчають,. У комплексних системах отримана модель також може використовуватися в процесі навчання. У цей час адаптивне тестування реалізується в основному у вигляді різних алгоритмів комп'ютерного тестування.

Адаптивне тестування повинне задовольняти наступним вимогам:

- можливість регулювати пропорції пропонованих легких, середніх і важких завдань залежно від числа правильних відповідей тестуємого;
- можливість регулювати пропорцій пропонованих різних тематичних розділів навчальної програми в тесті;
- можливість регулювати рівня складності пропонованих тестів з урахуванням семантичної компетенції тестуємого;

- включення адаптивного механізму перекладу на більш високий рівень завдань на тому самому рівні пропонованих завдань;
- кожне завдання більш високого рівня оцінюється більш високими балами.

Такі характеристики технології комп'ютеризованого тестування як наявність інтерактивного інструментального середовища й застосування в довільній галузі знань є вирішеними завданнями й не можуть бути предметом дослідження. Те ж саме можна сказати про п. п. 9, 10 характеристик.

Вибір алгоритмів тестування у цей час фактично обмежений формами представлення тестових завдань і алгоритмами оцінки результатів тестування. Досягнення більш високих результатів і підвищення мотивації навчання в остаточному підсумку є основною метою будь-якого тестування. Сам процес тестування обумовлений алгоритмом тестування повинен бути максимально формалізований і, у той же час, гнучкий для того щоб одержувати адекватні оцінки знань тестуємих. Крім того, можливість регулювати пропорції пропонованих легених, середніх і важких завдань залежно від числа правильних відповідей тестуємого є нетривіальною вимогою. Це пов'язане з тим, що, в остаточному підсумку, при оцінці здатностей суб'єкта навчання так чи інакше використовуються статистичні методи для валідної апроксимації функції успіху в розумінні Раша.

Для екзаменатора природною є вимога мінімального обсягу тестових завдань, на які повинен відповісти суб'єкт навчання, однак для статистично повної оцінки здатностей кількість відповідей суб'єкта тестування повинне бути як можна більшим. Таким чином, останнє є суперечливим і, зважаючи на те, що самі тести, що розробляються, повинні бути адекватним відбиттям предметної галузі, сама процедура адаптивного тестування (включаючи розроблювальні тести, алгоритм тестування, алгоритм оцінки знань) повинна надійно контролюватися в процесі розробки й тестуватися як продукт.

Таким чином, вибір такого алгоритму тестування, структури тестів, які б задовольняли викладеним вимогам і характеристикам не є тривіальним завданням і надалі визначається як завдання дослідження.

## 1.6 Постановка задач дослідження

Для розробки адаптивних комп'ютеризованих систем навчання й тестування найбільш зручним представляється використання змішаної структури моделі того, якого навчають, яка дозволяє враховувати не тільки кількісні, але і якісні характеристики тих, яких навчають, [20]-[22].

При комп'ютерному навчанні можна виділити три методи по відповідно трьом ієрархічним рівням адаптації до тих, кого навчають [23]:

- адаптація до тим, яких навчають, як категорії користувачів;
- адаптація до групи тим, яких навчають,;
- адаптація до окремого тому, якого навчають,.

Перший рівень адаптації передбачає адаптацію до кожної категорії користувачів комп'ютерної системи навчання залежно від їхніх потреб і звичайно реалізується створенням спеціального інтерфейсу для кожного класу тих, яких навчають,. Такий підхід характерний для універсальних комп'ютерних систем.

Адаптація до групи тих, яких навчають, забезпечує адаптацію залежно від обраної спеціальності, утворювальної програми й психологічної спрямованості особистості. При цьому визначаються мети навчання, тобто обсяг необхідних знань, умінь і навичок і ступінь їх освоєння.

На третьому рівні досягається максимальний ступінь адаптації до тим, яких навчають, тому що він заснований на обліку особистісних характеристик того, якого навчають, його попередніх і поточних знань, умінь і навичок, досвіду роботи на ЕОМ, здатностей. Для організації адаптації до суб'єкта навчання, використовують різні методи:

- побудова адаптивної послідовності навчання;
- адаптивна вистава інформації;
- інтелектуальний аналіз і діалогова підтримка рішення завдань;
- адаптивна підтримка в навігації по досліджуваних дисциплінах;
- адаптивна підтримка співробітництва між тими, кого навчають, і викладачем.

Інформація витягає з відповідей того, якого навчають, у процесі рішення завдання або з його передісторії. Комплексна модель адаптивної комп'ютерної системи навчання й тестування (КСНТ) будується на основі моделей тих, яких навчають, представлення знань предметної області, системи тестування й керування процесом навчання.

Комплексна модель адаптивної КСНТ повинна включати:

- семантичну модель навчального курсу;
- модель тих, кого навчають;
- модель організації процесу навчання й тестування знань тих, кого навчають.

Предметну область навчальної дисципліни, представлену моделлю знань навчальних матеріалів. У якості моделі знань про структуру навчального матеріалу КСНТ обрана семантична мережа, яка містить відомості про поняття предметної області (Про) навчальної дисципліни, про їхні взаємозв'язки, складність окремих понять. Під методичним матеріалом будемо розуміти сукупність інформації, що містить знання про мети, форми, методи й засобах навчання й рішення навчально-виховних завдань.

Модель суб'єкта навчання, вирішує завдання опису попередніх знань тих, яких навчають, про навчальний курс і їх індивідуальних особистісних особливостях. Вона містить досить повну інформацію про тім, якого навчають, : рівень його знань, умінь і навичок, здатність до навчання, здатність виконання завдань (чи вміє він використовувати отриману інформацію), особистісні характеристики (тип, орієнтація) і інші параметри. Ці параметри можуть бути як

кількісними, так і якісними. Модель адаптивної КСНТ показано на рисунку 1.1. Вона містить декілька рівнів.

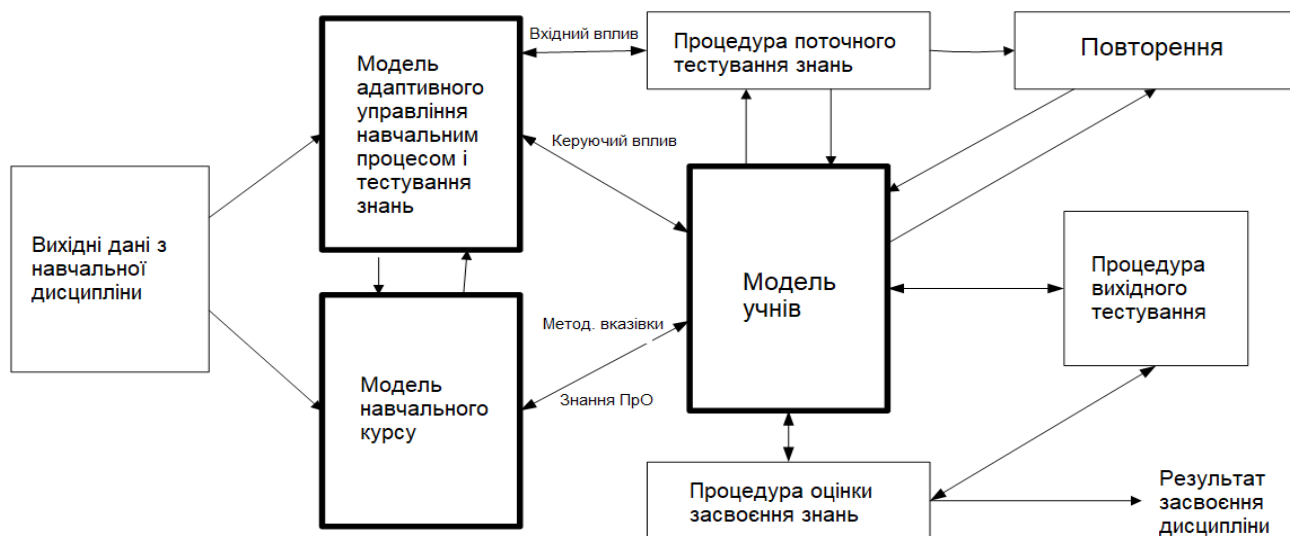


Рисунок 1.1 – Взаємодія моделей і процесів адаптивної комп'ютеризованої системи навчання й тестування

Модель системи адаптивного керування навчальним процесом і тестування знань суб'єкта навчання, призначено для автоматичного визначення порядку представлення їм матеріалів навчальної дисципліни залежно від результатів тестування на початку вхідного рівня знань по навчальній дисципліні й потім по вже вивчених темах і розділам.

Модель будується на основі експертних оцінок заданих рівнів засвоєння навчального матеріалу. Система повинна дозволити організувати перевірку знань із гнучким порядком видачі тестів, що залежать від рівня складності, що перевіряється матеріалу й попередніх відповідей тих, яких навчають. Якщо рівень підготовки низок і тестуємий часто дає невірні відповіді, то система почне перевіряти його знання по даній темі, пропонуючи завдання більш низького рівня складності. Чим більше помилок допускає суб'єкт тестування, тим простіше стають питання й тем нижче буде оцінка знань.

Таким чином, у процесі тестування КСНТ адаптується до рівня знань того, якого навчають.

## 2 ОПИС ПРОВЕДЕНИХ ТЕОРЕТИЧНИХ ДОСЛІДЖЕНЬ

### 2.1 Аналіз моделей навчального курсу

Модель навчального курсу можна використовувати як для керування процесом навчання, так і для обґрунтованого рішення питання про включення тих або інших фрагментів знань у програму навчального курсу. Вона спрощує розробку й розуміння навчальних програм, особливо типових і базових, а також проведення експертизи їх якості [16].

Навчальний курс характеризується сутностями й зв'язками між ними. У якості сутностей у навчальних системах можна розглядати поняття або теми, кожної з яких відповідає одиниця навчального матеріалу, що не вимагає (з погляду викладача) розподілу на підтеми. Кожна тема описується набором параметрів (атрибутів), істотних для керування навчанням.

У якості предметної області була обрана дисципліна «Надійність ПО», тому що вона є базовою при підготовці фахівців в області тестування програмного забезпечення, а дана спеціалізація є однією із самих затребуваних на ринку праці.

Для моделі навчального курсу, як найбільш зручної й адекватної класичній побудові навчального матеріалу, обрана семантична мережа. Вихідними даними для такої мережі є дані за структурою джерел учбово-методичної літератури по дисциплінах навчання, а саме: кількість розділів і тем у розділах навчальної дисципліни, їх тексти, графіки й малюнки, гіперпосилання й зв'язку між ними.

Семантична модель базується на семантичній мережі [17]. Це система знань, що має певний зміст у вигляді цілісного образу мережі. Механізм висновку полягає в поширенні по мережі збудження залежно від топології мережі й вхідних даних. Визначено семантичну мережу  $\Omega$  як двійку виду

$$\Omega = \{V, D\},$$

де  $V = \{v_i\}$  – множина вершин (вузлів мережі),

$D = \{d_j\}$  – множина дуг.

Вершина  $v_i$  семантичної мережі може бути визначена як

$$v_i = \{S, c\},$$

де  $S = \{s_k\}$  – безліч крапок входу у вершину,

$c$  – функція, що визначає стан вершини як диз'юнкцію станів множини синапсів  $S$ , що співставляються до вершини  $v_i$ .

Із програмної точки зору кожна вершина являє собою некоторий процес. Дуги між елементами визначають взаємозв'язки між вершинами й задають послідовність вивчення курсу.

У даній моделі виділені відносини двох типів: ієрархічні (предок – нащадок, частина – ціле і т.д.), відносини предшествовання (вперед – назад). Відповідно до ієрархічних відносин структурним компонентам приписуються індекси, кожний з яких відбиває шлях до відповідної до вершини від кореневої (навчальний матеріал у цілому). На рисунку 2.1 виділено п'ять рівнів, уважаючи кореневу вершину.

У конкретних реалізаціях кількість і найменування рівнів можуть бути різними. Індекси використовуються для ідентифікації й адресації структурних компонентів. Структурні компоненти, що не мають підлеглих (листя на дереві) звичайно називаються кадрами, сторінками, екранами.

На множині супідрядних вершин одного ієрархічного рівня, тобто, що мають на попереднім рівні загальну підпорядковуючу вершину (предок), визначаються відносини, що служать для опису порядку пред'явлення тому, якого навчають, відповідають структурних компонентів. Часто ці відносини називають переглядовими послідовностями.

Крім того, у структурі, як правило, можна виділити набір загальних інформаційних компонентів, асоційованих з навчальними матеріалом, але не охоплюваних його ієрархічною декомпозицією. До них належать: блок змісту, словник (глосарій), покажчики, тезаурус.

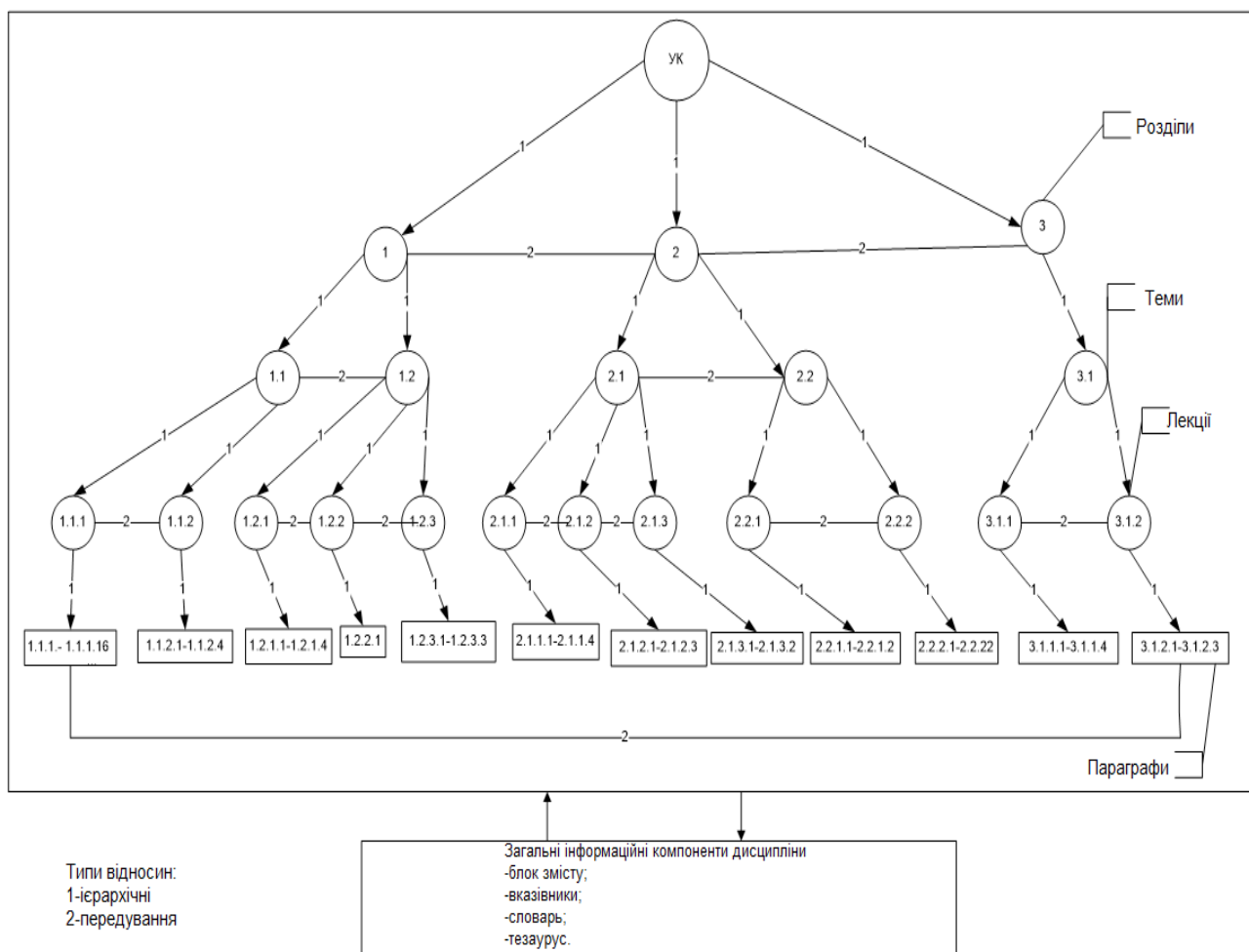


Рисунок 2.1 – Модель дисципліни

Модель навчального курсу на базі семантичної мережі реалізує адаптивне представлення інформації в базі навчального курсу. Модель використовує строгий формальний апарат семантичних мереж, що дозволяє автоматизувати рішення (тобто розробити алгоритм) низки фундаментальних завдань навчального процесу.

## 2.2 Модель системи адаптивного керування

Наступною моделлю комплексної моделі КСНТ є модель системи адаптивного керування навчальним процесом і тестування знань тих, яких навчають, призначена для автоматичного визначення порядку представлення їм

матеріалів навчальної дисципліни залежно від результатів тестування по вже вивчених темах і розділам [3]-[4], [6].

На основі завдань навчання, передбачуваних індивідуальних особливостей тих, яких навчають, і відведеного часу на вивчення тем і розділів навчальної дисципліни з можливих педагогічних технологій, описаних методичними знаннями, системою керування КСНТ формується методика навчання, тобто набір методів і форм навчання для кожного  $j$ -го того, якого навчають. Потім з обліком певних системою керування КСНТ початкового рівня знань того, якого навчають, і обраної для нього методики навчання, з вихідних знань ( $L_{01}, \dots, L_{0k}$ ) формуються порції навчального матеріалу (рівень занять), контрольних питань тестів тем і розділів ( $L_{k1}, \dots, L_{kk}$ ). Таким чином, модель організації навчального процесу, дозволяє представити навчальний матеріал розділів і підрозділів дисципліни у вигляді тем і занять певним чином згрупованих блоків знань, пропонованих тому, якого навчають, у процесі заняття.

Щоб побудувати й використовувати семантичну модель навчального матеріалу в КСНТ необхідно структурувати матеріал навчальної дисципліни на рівні розділів і підрозділів.

Звичайно модель знань навчальної дисципліни орієнтована на потокову логіку зв'язку окремих розділів і підрозділів. Але іноді виявляється важливим при побудові навчального матеріалу враховувати й логічну незалежність знань. Незалежність блоків знань дозволяє будувати різні варіанти керованої послідовностей викладу навчального матеріалу й вибирати з них найкращі з погляду викладача. У цьому випадку для адаптивного керування використовується так звана ярусно-паралельна форма графа знань (ЯПГ). ЯПГ це змішана модель знань – одночасно потокова і ярусна, причому в кожному ярусі графа використовується потокова структура знань [20 ], [23].

Ярусно-паралельний граф (ЯПГ) будується шляхом часткового впорядкування вершин по рівнях, на яких розташовані незалежні по логічних зв'язках блоки знань так, що на 0-м рівні розташовані вхідні знання, а на останньому рівні – цільове знання.

Граф кінцевий, має множину вхідних вершин (типу  $L_0$ ), проміжних ( $L_i$ ) і єдину вихідну вершину (типу  $L_k$ ). Висновок цільових знань реалізується системою формул висновку для кожної з його вершин, крім вхідних [23]. Граф показує, з яких складових і як складається цільове знання.

На рисунку 2.2 представлений ярусно-паралельний граф знань навчальної дисципліни.

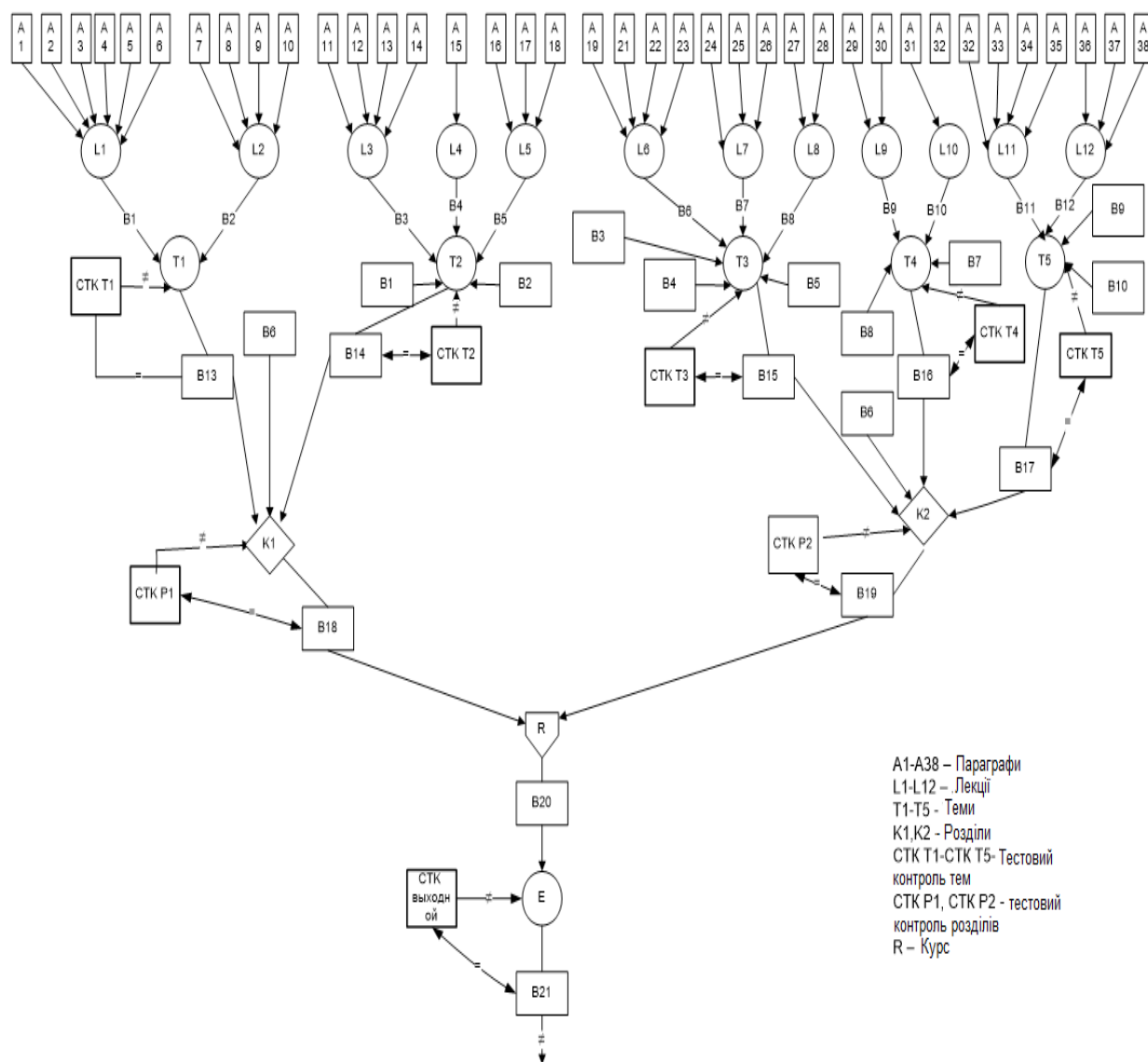


Рисунок 2.2 – Модель системи адаптивного керування навчальним процесом

Модель організації навчального процесу побудована у вигляді ярусно-паралельного графа знань навчальної дисципліни й призначена для рішення наступних інформаційних завдань:

- опису предметної області;

- опису бази знань організації навчального процесу (методичних знань);
- формування порцій навчального матеріалу для його пред'явлення тим, яких навчають,;
- організації поточного, рубіжного й підсумкового контролю засвоєння знань тими, яких навчають.

Попередній тестовий контроль забезпечує класифікацію тих, яких навчають, і формування черги на обслуговування КСНТ. Результати вивчення тем контролюються тестовим контролем СТК (система тестового контролю) Т1 – СТК Т5, розділів СТК Р1 – СТК Р2 і аналізуються спеціальним методом для визначення траєкторії подальшого руху по керуванню роботою КСНТ. Залік здійснюється системою СТК<sub>вих</sub> вихідного тестового контролю по навчальній дисципліні в цілому.

У цьому випадку навчання з використанням КСНТ проводиться шляхом пред'явлення тим, яких навчають, послідовності сформованих порцій навчального матеріалу, контрольних питань і оперативної корекції знань того, якого навчають,. При цьому шляхом обходу семантичної мережі навчального матеріалу відповідно до поточної відповіді того, якого навчають,, СТК йому видаються чергові питання, одержує відповіді на них, забезпечуючи виставу результатів тестування й збереження їх в електронному журналі. Так, якщо за результатами поточного тестування теми (роздягнула) навчальної дисципліни, той, якого навчають, показав відмінні, гарні й задовільні знання, то видається команда на продовження вивчення наступної теми або роздягнула, а якщо ні, то — команда на повторення навчального матеріалу теми або роздягнула.

Контроль здійснюється за допомогою системи тестування вивчених тем, де встановлюється поточний рівень знань того, якого навчають, і на підставі цього проводиться оперативна корекція ходу обчислювального процесу. По завершенню певного етапу навчання, проводиться рубіжний по розділу або підсумковий контроль знань того, якого навчають, по навчальній дисципліні. На основі отриманих оперативних даних ухвалюється рішення про завершення навчання або коректуванню знань того, якого навчають,.

Таким чином, при реалізації такої адаптивної комплексної моделі КСНТ при організації навчального процесу здійснюється керування процесом вивчення навчальної дисципліни тими, яких навчають, практично без втручання викладача в цей процес, що дозволяє використовувати КСНТ не тільки при аудиторних заняттях, але й у системі дистанційного навчання.

### 2.3 Модель суб'єкта навчання

Модель того, якого навчають, є однією з базових компонентів інтелектуальних комп'ютерних систем навчання. Вона містить досить повну інформацію про тім, якого навчають,: рівень його знань, умінь і навичок, здатність до навчання, здатність виконання завдань ( чи вміє він використовувати отриману інформацію), особистісні характеристики (тип, орієнтація) і інші параметри. Модель студента динамічна, тобто змінюється в процесі проходження курсу, у ході роботи із системою [24].

Модель суб'єкта навчання – сукупність набору характеристик того, якого навчають,, вимірюваних під час роботи системи з тим, яких навчають, і визначає ступінь засвоєння їм знань по досліджуваному предмету й методів (правил) обробки цього набору. У першу чергу ці правила повинні проводити зміни самої моделі того, якого навчають, за результатами його роботи із системою.

Модель суб'єкта навчання повинна містити в собі інформацію:

- про мету навчання;
- про знання того, якого навчають, у рамках досліджуваного курсу (поточний стан процесу навчання);
- про особливості подачі навчальних матеріалів і вибору контрольних завдань і питань;
- про правила зміни моделі того, якого навчають, за результатами роботи з тим, яких навчають,.

Моделі того, якого навчають, можуть бути розділені на дві основні групи: фіксуючі й імітаційні. Фіксуючі, у свою чергу, включають скалярні, оверлейні (векторні й мережні) і генетичні графи. До імітаційних можна віднести моделі обмежень, помилок і фальш-правил [28].

Для кожного того, кого навчають, може бути задана своя мета роботи із системою й своя підмножина досліджуваного матеріалу, яке визначає початкове настроювання системи і є базою для подальшої роботи з тим, яких навчають,.

Можна виділити два основні підходи до побудови таких моделей.

- в експертно-навчальних системах (ЕНС) під моделлю того, якого навчають, розуміють набір характеристик (параметрів) і сукупність правил, які на підставі значень цих характеристик управляють процесом спілкування системи з тим, кого навчають;

- в інших класах навчальних систем під моделлю того, якого навчають, звичайно розуміють набір параметрів, вимірюваних під час роботи системи з тим, яких навчають, і визначає ступінь засвоєння їм знань по досліджуваному предмету.

Розуміння моделі того, якого навчають, як набору параметрів – це занадто вузьке розуміння. Без обліку методів, які працюють із даним набором характеристик, цей набір втрачає зміст. Отже, модель об'єкта, якого навчають, як сукупність набору характеристик того, якого навчають, і методів (правил) обробки цього набору. В ЕНС ці правила прив'язані до конкретного ПЗ, що дозволяє, природно, більш якісно імітувати спілкування того, якого навчають, з фахівцем у даній предметній області. Але АНС не може бути орієнтована на конкретну ПО, тому й призначення правил, що входять у модель суб'єкта навчання, повинне бути іншим. У першу чергу ці правила повинні проводити зміни самої моделі того, якого навчають, за результатами його роботи із системою. Це дозволить викладачеві управляти формуванням цієї моделі, тобто здійснювати алгоритмічне настроювання без програмування.

Оверлейна модель того, якого навчають, – найбільш проста для реалізації моделі того, якого навчають,. Вона будується в припущенні, що знання того,

якого навчають, і знання системи мають аналогічну структуру, при цьому знання того, якого навчають, є підмножиною знань системи. Кожній темі додається числовий атрибут, що показує ступінь розуміння тим, яких навчають, матеріалу по цій темі. Значення цього атрибута визначаються в ході опитування суб'єкта навчання.

Різницева модель того, якого навчають, – модель того, якого навчають,, при побудові якої система аналізує відповіді того, якого навчають, і порівнює їх з тими знаннями, які закладені в системі і якими користується експерт при рішенні подібних завдань. Відмінності між цими знаннями й лягають в основу моделі того, якого навчають,. Ця модель дозволяє враховувати не тільки відсутність знань у того, якого навчають,, але й неправильне їхнє використання.

Пертурбаційна модель того, якого навчають, – модель того, якого навчають,, яка будується в припущенні, що знання того, якого навчають, і знання системи можуть частково не збігатися. У цьому випадку важливою передумовою побудови такої моделі є ідентифікація причин розбіжності, тому що без визначення розбіжностей модель того, якого навчають, буде занадто невизначеною.

Розрізняють наступні причини розбіжностей:

- недолік знань;
  - наявність помилкових знань;
  - неправильне застосування знань або невміння їх застосувати;
- помилка, породжена неухважністю;
- навмисне допущена помилка (дається перша відповідь, що попалася).

Остання причина легко виявиться, якщо задати кілька простих питань. Виявлення інших причин можна здійснити шляхом повторного опитування того, якого навчають, (так зване уточнення).

У цілому, аналіз наукових досліджень показує, що незважаючи на значне просування, що намітилося останнім часом у реалізації різних комп'ютерних технологій навчання, потенціал інформаційних технологій реалізується в процесі навчання ще недостатньо через відсутність надійних підходів до створення

адаптивних тестів, алгоритмічної реалізації адаптивного тестування як процесу, адекватного контролю при конструюванні систем адаптивного тестування й розробки тестів і тестування зазначених систем.

Ключовим моментом у цьому переліку є метод тестування, який має часто суперечні один одному характеристики й також повинен задовольняти суперечливим вимогам. Тому основним завданням даного дослідження є, — вибір і створення методів тестування, що відповідають принципам побудови адаптивних систем тестування, що і є як можна більш універсальним, простим у реалізації, що й дозволяють формально, з мінімальною кількістю правил визначати структури тестових завдань.

#### 2.4 Аналіз існуючих методів адаптивного тестування

Аналіз методів створення тестових систем дозволив виділити наступні основні методи адаптивного тестування.

- Item-Pool-Partitioning (IPP);
- Weighted-deviation method (WDM);
- Testlet-based adaptive testing (TBAT);
- Multi-stage testing (MST).

IPP метод ґрунтується на критерії максимуму інформації. Суть методу полягає в наступному: для кожного тестового завдання з безлічі тестів визначається його зв'язок з іншими темами лекційного курсу ( тобто одне питання може ставитися до декільком темам). Принцип максимуму інформації реалізується, у першу чергу, через визначення завдання з найбільшою кількістю зв'язків з іншими темами лекційного курсу. Якщо була дана правильна відповідь, то здійснюється перехід на наступне питання з таким же більшим кількість зв'язків, а якщо ні, то ставиться питання з меншим ступенем зв'язаності ( у порівнянні з першим) і т.д. У випадку рівності у двох або декількох питань



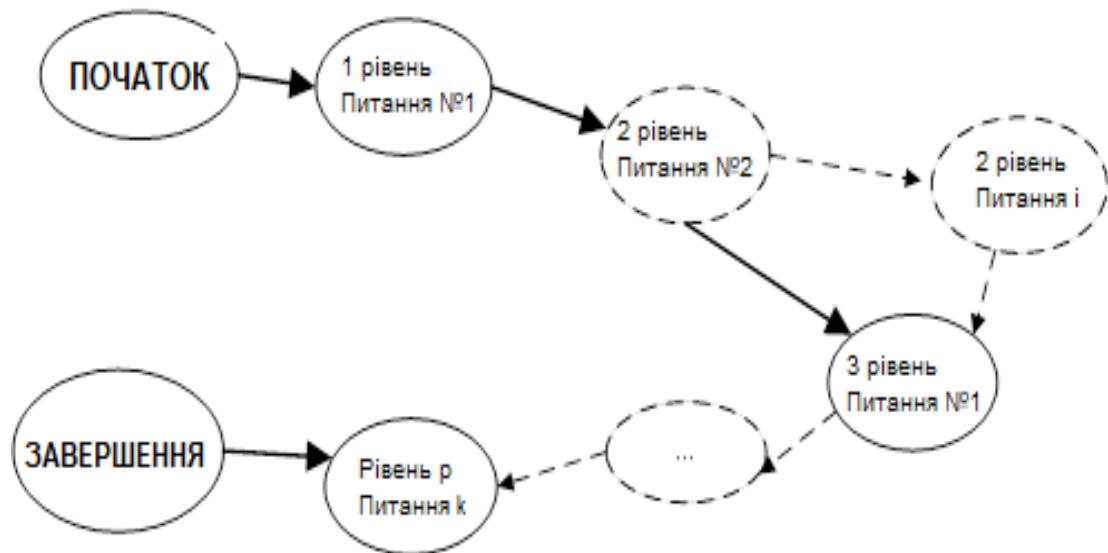


Рисунок 2.4 – Процедура тестування при IPP підході

Більш загальним підходом є метод WDM – метод зважених відхилень. У цьому підході всі істотні ознаки формуються як ряд верхніх і нижніх границь для цілого ряду завдань. На відміну від методу IPP зв'язок з лекційними темами конкретного завдання має додатковий ваговий коефіцієнт, що є заходом потужності цьому зв'язку. Зазвичай він нормується від 0 до 1. Початок тестування починається з питань тих тем, які входять у групи тем з найбільше сумою вагових коефіцієнтів (вибирається випадковим образом) з послідовним переходом до тем, що мають меншу суму вагових коефіцієнтів. Графічна вистава організації даного підходу презентовано на рис. 2.5.

Для кожної  $i$ -ї зв'язки ( $1 \leq i \leq m$ , де  $m$  – кількість тем) є свій ваговий коефіцієнт  $l_{ij}$  ( $0 \leq l_{ij} \leq 1$ ). Усього  $k$  питань ( $1 \leq j \leq k$ ). Також, як і в першому підході є  $P$  рівнів:

- рівень 1:  $(\sum l_{min})^1 < \sum l_{ij}^1 < (\sum l_{max})^1$  ;
- рівень 2:  $(\sum l_{min})^2 < \sum l_{ij}^2 < (\sum l_{max})^2$  ;
- ... ..
- рівень  $P$ :  $(\sum l_{min})^P < \sum l_{ij}^P < (\sum l_{max})^P$  .

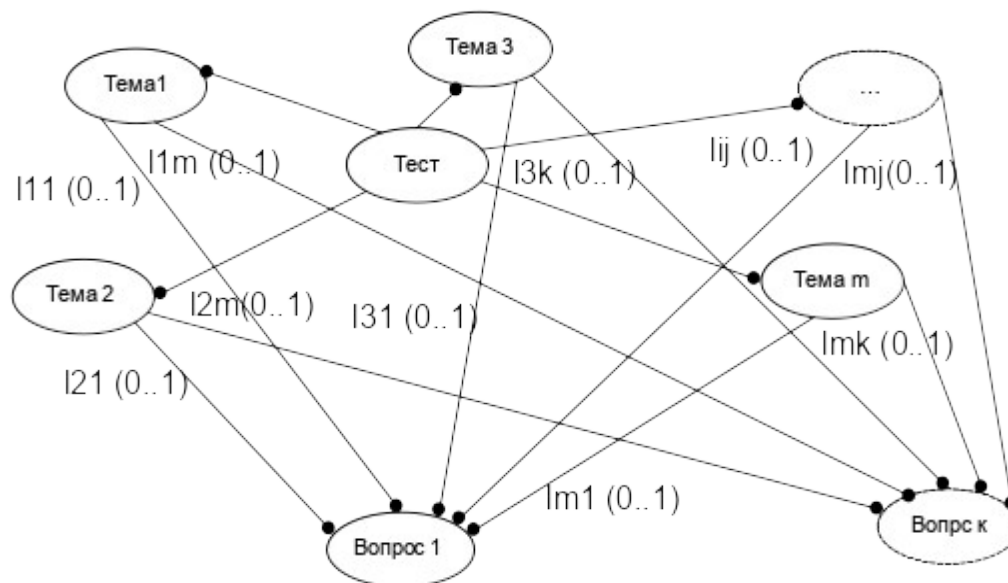


Рисунок 2.5 – Графічне подання організації WDM підходу

У даному найпростішому випадку передбачається перехід від більш загального ( відповідно до торби потужностей зв'язків) питання високого рівня до питання більш низького рівня незалежно від правильності відповіді. Тест закінчується при проходженні всіх рівнів.

Виникла необхідність у сполученні істотних специфікацій і статистичного критерію у виборі завдання для комп'ютерного адаптивного тестування, Сутність рішення полягає в тому, щоб змінити розмір одиниць (частин) у безлічі завдань. На противагу дискретним завданням, безлічі завдань устанавлюють так звані блоки, які являють собою зв'язані завдання, об'єднані загальною тематикою (або за іншою ознакою), причому ці зв'язки розглядаються як неподільні одиниці. Блоки питань об'єднані таким чином, що припускають фіксований порядок видачі тестових завдань. Приміром, вони можуть бути організовані відповідно до ієрархічної структурою, при якій тестуємі й одержує завдання найвищого (низького) рівня труднощів. Якщо тестуємі відповідає правильно, то йде нагору (униз). Суб'єкт тестування переходить від одного блоку до іншого.

Головною відмінністю даного методу від попереднього полягає в тому, що неподільними одиницями є не питання, а блоки питань, задані по певному сценарію. Ілюстрація блоку питань представлено на рис. 2.6.



Рисунок 2.6 – Ілюстрація блоку питань при ТВАТ-підході

Ідея адаптивного тестування, заснована на блоках питань, має безпосереднє відношення до одного із самих старих форматів багатоступінчастого тестування. При багатоступінчастому методі тестуємий проходить через послідовність тестів, рухаючись убік більш складних питань при успішних відповідях або до більш простих, якщо його відповіді невірні.

При цьому перехід від питання до питання відбувається за певними правилами. У більш ранній літературі обговорювалася паперова версія цього методу без статистичної обробки знань суб'єкта тестування після кожного тестового завдання, однак автоматизація у практиці тестування надає можливість статистичної оцінки знань на кожному кроці тестування.

### 3 ОПИС ПРОЕКТУВАННЯ АЛГОРИТМІВ СИСТЕМИ

#### 3.1 Моделювання адаптивного тестування

Запропонована модель лінійного програмування для конструювання систем багатоступінчастого тестування, засновану на критерії максимуму інформації, що дозволяє брати до уваги різноманітні обмеження для різних тестових структур. Можливість включення таких обмежень у системи східчастого адаптивного тестування привернула увагу до цього методу.

Одним з видів багатоступінчастого підходу є пірамідальне тестування, що представлено на рисунку 3.1.



Рисунок 3.1 – Пірамідальна модель адаптивного тестування

На рисунку 3.1 жирною лінією показаний маршрут одного суб'єкта навчання (траєкторія відповідей на запитання).

Якщо зміна порядку пред'явлення тестового завдання відбувається на кожному кроці тестування (постійна адаптація), то траєкторія відповідей на запитання прийме вид представлений на рисунку 3.2.



Рисунок 3.2 – Схема проходження завдань при постійній адаптації  
(П – помилкова відповідь, В – вірна відповідь)

Якщо ухвалення рішення про зміну порядку проходження завдань здійснюється після аналізу результатів звітів випробуваного на спеціальному блоку завдань (блокова адаптація), то схема проходження завдань має такий вигляд (рисунок 3.3).



Рисунок 3.3 – Схема проходження завдань при блоковій адаптації:  
(О – помилкова відповідь, В – вірна відповідь)

Вище наведені підходи різні по своїй суті. Методи IPP і WDM вводять обмеження через алгоритм вибору завдання. Останні два підходи вбудовують усі обмеження безпосередньо в групу з множині тестів з якої потім іде керування самого процесу тестування. Відмінності між наведеними методами приводять до наступних аспектів:

- до рівня адаптації, який можливий під час тестування;
- до розширення опису самого завдання;
- до можливості експертній оцінці дійсного змісту тестів;
- до природи реалізації обмежень, керованих процесом тестування;
- до можливості порушення цих обмежень.

Перші два методи передбачають відновлення оцінки знань після кожного тестового завдання, – у такий спосіб вони припускають можливість максимального рівня адаптації. Однак, для того щоб процедура тестування була успішною, обоє методу враховують усі стосовні до завдання закодовані атрибути. Якщо потенційно важливий атрибут упущений, зміст тесту може стати розбалансованим.

Метод WDM мінімізує тільки зважені суми відхилень від обмежень, тому деякі із цих обмежень можуть бути втрачені навіть тоді, коли повністю виконане кодування атрибутів тестового завдання.

В IPP-підході вибір завдань і реалізація обмежень є зв'язаним. Хоча послідовний вибір тестових завдань припускає оптимальну адаптацію, послідовна реалізація обмежень є більш ніж не ідеальною.

Алгоритми з такими особливостями мають схильність до вибору завдань із найбільшою кількістю зв'язків з іншими темами на початку тестування. Однак вибір деяких із цих питань може виявитися не оптимальним при подальшій проходженні тесту. У цьому випадку це приведе до того, що результат оцінки знань буде міні адекватним, чому при оптимальнім адаптивнім тестуванні й/або до неможливості завершення тесту без порушення обмежень.

В TBAТ і MST методах є опція експертного оцінювання всього тестового матеріалу перед процедурою тестування. Явне кодування істотних для завдань

атрибутів не завжди є необхідним. Однак ці підходи не мають повної адаптації при виборі завдання. Ці методи дозволяють часткову адаптацію з ієрархічними переходами. Також завдання угруповання блоків тестових завдань (testlets) і створення багатоступінчастої тестової системи, таким чином, щоб будь-яка частина системи задовольняла всім обмеженням, приводить до великої комбінаторної проблеми, яка у свою чергу може привести до неоптимальної системи з ієрархічними переходами й/або до порушення обмежень. Однак, як було відзначено, формальні методи для створення багатоступінчастих систем тестування все-таки існують.

Використання цих методів дійсно дозволяє одержати явне кодування всіх важливих для завдання атрибутів. В останніх двох методах групи питань вибираються послідовно, а це значить, що адаптація не є оптимальною. З іншої сторони обмеження реалізуються одночасно, коли групуються блоки тестових завдань або тестові завдання перед початком тестування.

Аналіз здійснювався за наступними критеріями:

- інформативність – застосування принципу максимуму інформації для кінцевої кількості заданих тестових завдань (теорія Раша);
- статистична обробка – у процесі тестування здійснюється оцінка здатностей суб'єкта тестування методами математичної статистики (наприклад  $i$ -я ймовірність правильної відповіді на завдання  $i$ -го рівня труднощі), крім цього статистична обробка дозволяє визначити момент закінчення процесу тестування, що відбувається, коли функція розподілу  $i$ -х ймовірностей правильних відповідей на завдання  $i$ -го рівня складності змінюється несуттєво.
- ймовірність помилок – є ймовірність того, що тестування буде проводитися неадекватно знанням суб'єкта тестування або відбудеться порушення обмежень САТ як системи тестування;
- складність проектування;
- адекватність оцінки знань суб'єкта тестування;
- рівень адаптації, пов'язаний із критерієм інформативності – чим вище рівень адаптації, тем вище інформативність системи;

- експертна оцінка – результативний висновок, що дозволяє судити про ті або інших властивостях тестових завдань;
- формалізація процедур створення тестових завдань і обмежень, що припускає застосування рутинного алгоритму для досягнення заданої мети при певному наборі початкових умов.

Результати аналізу наведено в таблиці 3.1.

Ця оцінка існуючих методів комп'ютерного адаптивного тестування приводить до дуже важливої дилеми. Алгоритм із оптимальними властивостями повинен був би вибирати завдання в такій послідовності, яка дозволяла досягати оптимальної адаптації й одночасно враховувати всі обмеження для того, щоб запобігти порушенням деяких з них або не одержати неоптимальну адаптацію при подальшій тестуванні.

Таблиця 3.1 – Результати аналізу методів адаптивного тестування

№ п/п	Методи	IPP	WDM	ТВАТ	MST
	Критерії				
1	Інформативність	-	-	-	+
2	Статистична обробка	-	-	+	+
3	Імовірність помилок	+++	+++	++	+
4	Складність проектування	++	+++	++	+
5	Адекватність оцінки знань тестуємого	---	--	--	-
6	Рівень адаптації	+	+	++	+++
7	Експертна оцінка	-	-	-	+
8	Формалізація процедур створення тестових завдань і обмежень	-	-	-	+

Можливі наступні рішення цієї дилеми.

- реалізація алгоритму з можливістю повернення назад, для того щоб поліпшити наступні рішення;

- реалізація маршового алгоритму, який би враховував наслідки прийнятих у майбутньому рішень.

В адаптивнім тестуванні відстеження у зворотному порядку неможливо; алгоритм застосовується в масштабі реального часу й попередній вибір не може бути відмінний. Таким чином, залишається тільки одна можливість: використовувати алгоритм, який щораз вибирає нове завдання. Це винятково новий клас алгоритмів.

### 3.2 Опис методу роботи системи адаптивного тестування

Для визначення базового алгоритму, необхідно привести сценарій роботи системи. У його основі лежить модель приймання іспиту викладачем у студента, як модель адаптивного тестування. Такий вибір сценарію роботи системи обумовлений тим, що, по-перше, дана процедура історично зложилася дуже давно й добре формалізована, по-друге, — при проектуванні тестів, їх розроблювачеві необхідно опиратися на загальноприйняті, відомі й використовувані їм методи з мінімальною модифікацією. Далі наведений сценарій приймання іспиту.

Сам алгоритм адаптивного тестування повинен мати на кожному кроці (при переході від одного завдання до іншого) максимальною інформативністю (тобто надавати максимум інформації про здатності студента на одне задане питання). У той же час, не можна повністю відмовлятися від обліку суб'єктивних властивостей студента, які можуть виражатися в нерозумінні очевидно поставленого питання або завдання.

Додаткові питання є невід'ємною частиною тестів за певних умов, однак потрібно врахувати:

- не всі додаткові питання є тотожні як з погляду складності, так і з погляду повноти відповідності основному питанню;
- ланцюжка додаткових питань є логічно зв'язаними послідовностями;

- питання ( як основні, так і додаткові) задаються послідовно, тобто неможливо задати два або більш питання одночасно;
- частота завдання додаткових питань різна.

За базову, але не єдину модель системи адаптивного тестування, обрана вистава у вигляді Марківських мереж [22], яке має наступні властивості:

- основною одиницею є питання певної складності, у якого можуть бути ланцюга додаткових питань;
- вибір додаткового питання визначається виходячи з імовірності появи кожного додаткового питання як потоку найпростіших подій з експонентним розподілом або шляхом логічного висновку на вибір невірної відповіді;
- система тестів є замкненою в розумінні Марковских мереж, тобто якщо питання (основний або додатковий) являє собою стан  $s_i$ , що має ймовірність перебування в ній системи  $P(s_i)$ , те сумарна ймовірність  $P_{\Sigma}$  перебування системи в  $S$  тотожно визначається як:

$$P_{\Sigma} = \sum P(s_i) = 1;$$

- зазначений стан – тестове завдання як процес оцінки знань і вмінь випробуваного, повинне виконуватися у випадку вдалої відповіді на один з питань і/або додаткового питання (або їх ланцюги), тобто «вирахування» одного з питань не повинне «руйнувати» зазначений стан.

Останнє дозволяє обходити питання нескінченно, тобто завершення тесту можливо в наступних випадках:

- вичерпані всі питання в банку тестових завдань;
- досягнуто кінець тесту;
- рівень знань оцінений з достатньою точністю;
- рівень знань розташований на досить далекій відстані від значення критерію проходження тесту;
- випробуваний показує свою неспроможність при відповідях на запитання тесту.

Подання алгоритму тестування у вигляді Марківських мереж не є вичерпним. Як відзначалося раніше, як при вірному, так і при невірній відповіді на запитання повинне ухвалюватися одне з наступних рішень:

- перехід до наступного основного питання з вибором рівня його складності;
- перехід до додаткових питань (точніше до їхніх дерев), при цьому потрібно відкинути вже задані додаткові питання (як окремі, так і їх дерева);
- повернення до основного питання, якщо отримана відповідь на додатковий (або додаткові) питання;
- закінчення тестування.

«Питання» (тестове завдання) як суб'єкт у розглянутому процесі тестування припускає виконання правил, певних екзаменатором. Це відповідає й традиційному процесу приймання іспиту, тобто залежно від відповіді на поставлені питання екзаменатор ухвалює зазначені вище рішення. При цьому він бере до уваги як фрагментарно розділений відповідь (оцінка за кожне питання й висновок середнього), так і за певними їм правилами весь ланцюжок відповідей суб'єкта тестування.

Загальним підходом у прийнятих методах, крім загальних правил для всіх завдань, є компонування завдань із різних частин (малюнків, таблиць, мультимедійного змісту), обумовлених як стимули. Це дозволяє, заощаджувати ресурси по розміщенню стимулів і виділяти обробну програму, загальні правила в різні частини ПЗ.

Даний підхід не дозволяє в достатньому ступені індивідуалізувати «питання». Насамперед, це пов'язане з тим, що кожне питання поєднує як безпосереднє завдання, так і рішення, пов'язане з виконанням цього завдання з відповідями на додаткові питання.

### 3.3 Модифікований алгоритм адаптивного тестування

Враховуючи викладене, для забезпечення гнучкості в прийнятті рішень, простоти створення питань і правил, що визначають прийняття рішень для конкретного питання, доцільним є об'єднання питання й процедури ухвалення рішення, пов'язаної з ним. Такий підхід спрощує як процедуру тестування, так і саму систему тестування з погляду задоволення вимогам мінімальної складності застосовуваних алгоритмів.

Прийняття загальних для всього процесу тестування рішень, вимагає загальних підходів в одному сеансі тестування. Ці підходи визначаються:

- застосуванням загального методу (або методів), що визначає той крок у процесі тестування, коли додаткова інформація про знання суб'єкта тестування буде надлишковою;
- процедурою початку роботи системи тестування (вибір першого питання) і стратегією переходу від одного питання до наступного;
- наданням докладних результатів тестування, як у природній, так і обробленій тими або іншими методами формі.

Це вимагає використання протоколу тестування й застосування:

- алгоритмів операційного й статистичного аналізу результатів тестування з погляду надмірності або недостатності інформації;
- алгоритмів, що визначають, рівень підготовки суб'єкта тестування;
- алгоритмів, що забезпечують стохастичні переходи по мережах тестових завдань.

Фактично, модель студента в окремій сесії визначається:

- протоколом опитування;
- результатами оцінки знань.

Таким чином, даний підхід формує те, що є природнім для викладача, має аналог у класичнім розумінні іспиту й визначається як модель випробуваного.

Як відзначалося вище існують кілька підходів до алгоритмів оптимального адаптивного тестування, серед яких можна виділити два протилежні підходи.

Перший підхід, заснований на максимумі інформації, важко назвати адаптивним і в асимптотике він припускає відповіді суб'єкта тестування на всі питання, що саме по не ефективно.

Другий підхід заснований на принципі максимуму інформативності (похідній від інформації із траєкторії питань) і в асимптотике припускає існування одного питання, відповідь на який надає інформацію про знання й уміння суб'єкта тестування, що є також нереальним.

Модифікація кращих, відповідно до таблиці 3.1 алгоритмів класу MST, полягає в тому, що попередньо потрібно згрупувати завдання по рівнях складності, і випадковим образом вибрати деякий середній або певний за апіорною інформацією екзаменатора рівень питання. Впродовж процесу тестування у випадку успішної відповіді на поставлене тестове завдання, додати питання з найближчої групи питань більш високого рівня складності, а якщо ні, то, – з найближчої групи менш складного рівня.

Процедура додаткових питань пропонується виконуватися двома способами. Перший спосіб полягає в тому, що до закінчення множини додаткових питань одного основного (або єдиного питання) не здійснюється ніяких дій, крім фіксування відповідей на запитання з наступної передачі протоколу основному питанню, де ухвалюється одне з наступних рішень:

- зарахувати відповідь на основне питання як правильний з можливим коректуванням складності;
- вважати питання незадовільним і перейти до групи питань більш низького рівня;
- повторно поставити додаткове запитання із залученням, що залишилися додаткових питань;
- повторно поставити основне запитання без залучення додаткових питань.

Другий спосіб: «Псевдо-ітерактивний», – коли функції прийняття рішень про правильності відповіді й (або) перехід до інших основних, додатковим питанням або закінченні тестування передається додатковому питанню.

Останнє є загальною формою, тобто псевдо-ітерактивна процедура може бути зведена до маршової шляхом видалення логічного аналізу й прийняття рішень (тільки перехід у класичному розумінні Марківських мереж).

Наявність логічного аналізу й ухвалення рішення в тестовім завданні моделює правило поведінки екзаменатора. Тобто прийнятий ситуативний підхід на підставі правил для конкретного питання про поведінку (передача керування іншому питанню) у рамках окремо взятого завдання. Це дозволяє не тільки враховувати елементарну безліч результатів відповіді (ложно або істинно), але й те, яка відповідь із безлічі неправильних відповідей обраний з урахуванням стану протоколу тестування до звертання до конкретного питання.

Додаткові питання можуть бути обрані з основних з будь-яким рівнем складності й/або створені окремо (для конкретного тесту). Як і основним питанням, їм повинен бути привласнений рівень складності. Таким чином, можна узагальнити «питання» у тесті як структуру (клас) даних і визначити його поведінка. Питання повинні мати:

- ознаку, що визначає є він основним і/або додатковим;
- ознаку, що визначає рівень складності;
- ознаку, що визначає приналежність і відносини до іншими темами й питаннями;
- ознаку, що визначає питання, з якими в процесі тестування припинилися зв'язки;
- ознаку, що визначає чи може невірна відповідь перервати тестування;
- ознаку, що визначає зв'язки (ті питання, у які з нього можна перейти), обумовлені логічними висновками;
- ознаку, що визначає зв'язки (ті питання, у які з нього можна перейти звичайним для Марківських мереж способом);

- ознака «смерті» питання в класичних мережах Маркова, тобто випадок, коли питання вже було задано;

- ознака, що визначає чи є питання унікальним чи ні для даної мережі;

- ознака, що визначає який або які наступні питання можуть бути викликані цим питанням, тобто яким питанням може бути передане керування.

Питання повинні забезпечувати наступні положення:

- забезпечити приймання керування від попереднього питання з одержанням поточного протоколу опитування;

- забезпечити видачу тестового завдання й варіантів відповідей до них;

- забезпечити початок відліку власного часу в момент видачі завдання;

- підтримувати логічну функцію, що визначає зниження рівня складності, у випадку виходу за межі часу, що допускається, певного для відповіді на дане питання й/або прийняти за певних умов для цього питання рішення про правильний/неправильній відповіді;

- ухвалити рішення щодо продовження тестування або його припиненні;

- прийняти рішення й вибрати наступне питання по заданому стохастичному алгоритму або правилам;

- визначити чи є дійсним ознака «смерті» для цього питання;

- передати протокол опитування наступному питанню й контролерові мережі питань;

- передати ознаки, що визначають стан питання контролерові мережі питань;

- перевірити роботу питання (передбачається передача керування контролерові мережі й/або наступному питанню).

На рисунку 3.4 представлений фрагмент мережі питань (основне питання з додатковими).

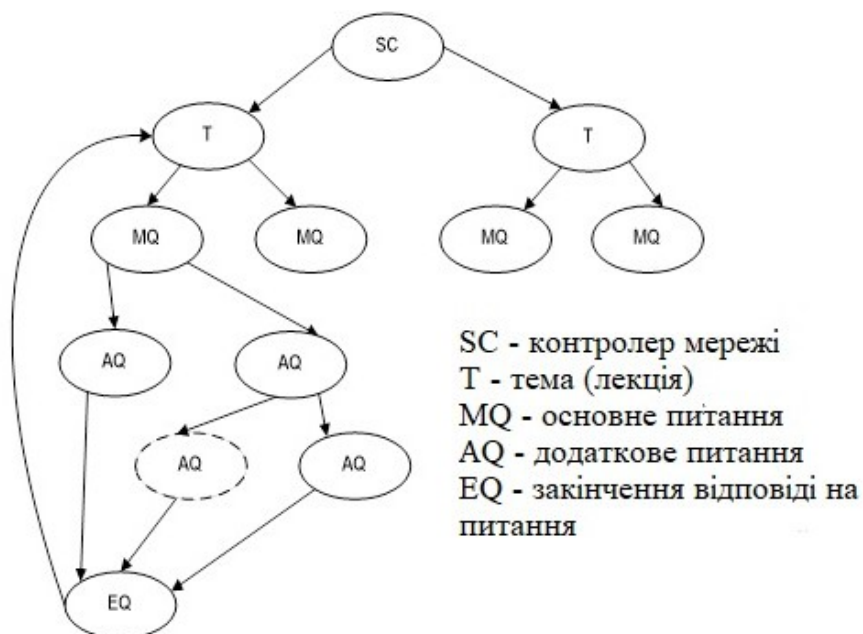


Рисунок 3.4 – Фрагмент мережі питань

### 3.4 Алгоритм керування станами мережі питань

Необхідність керування мережею питань обумовлена наступними причинами.

Як відзначалося вище, у класичній розумній мережі Маркова перехід з одного стану мережі в інше являє собою нескінченний процес. Існує також деякий початковий стан ( у цьому випадку це випадково обраний питання заданого рівня складності), що є стартовою крапкою цих переходів.

Вибираючи початковий рівень складності на початку тестування, і переходячи до іншого рівня складності, здійснюється перехід у підсистему, певну для обраного рівня складності для якої слушні правила описані вище. Слід зазначити, що принцип залучення додаткових питань, що мають інший рівень складності, ніж обраний трохи ускладнює поведінку мережі (рисунку 3.5).

Питання, як основне, так і додаткове не може бути двічі завдане. Таким чином, є модель «загибелі» й «розмноження» в термінах мереж Маркова.

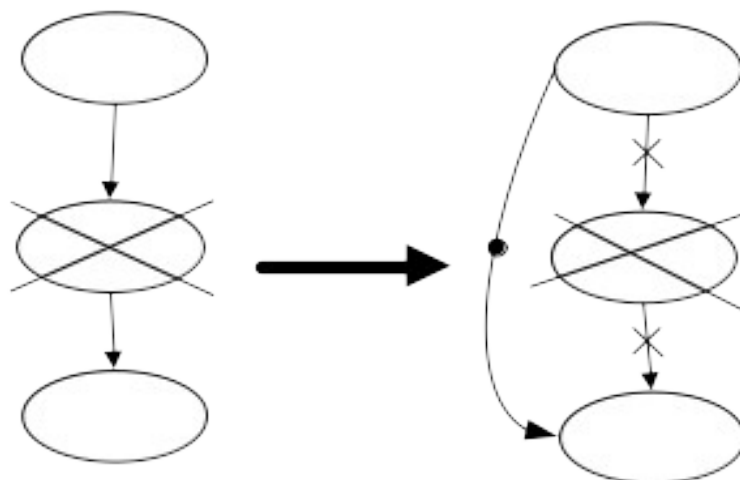


Рисунок 3.5 – Ілюстрація знищення (загибелі) найпростішої гілки додаткових питань

До основних питань (навіть із різних тем) можуть ставитися однакові додаткові питання або навіть їх ланцюжка. При завданні додаткового питання в контексті основного, він повинен бути знищений і в інших питаннях.

Коли віддаляється питання з послідовного ланцюжка, тобто коли є один вхід і один вихід, питання віддаляється й утворюється зв'язок, відповідна до виходу із зухвалою його питання.

У випадку, коли додаткові питання, що належать основному питанню, утворюють фрагмент деревоподібної мережі, то при використанні якого-небудь додаткового питання із усього безлічі цих питань певного для тесту взагалі, гілка деревоподібної мережі (графа) знищується повністю. Такий же результат має місце тоді, коли це додаткове питання задане безпосередньо в контексті конкретного основного питання тесту, але в цьому випадку цей результат має місце після закінчення відповіді на додаткові питання, що належать його гілці.

У результаті проведеного аналізу моделей і методів адаптивного тестування можна зробити висновки, що кращими є підходи, засновані на класі

багатоступінчастих методів адаптивного тестування, які ефективно підтримують комплексну модель адаптивного тестування.

Дана модель у свою чергу є оптимальною для реалізації контролю знань у дистанційному навчанні. Однак, виявлене протиріччя між адекватною статистичною оцінкою знань і вмінь випробуваного й обраного методу, заснованого на принципі максимуму інформативності.

Обрана модель представлення тестових завдань із використанням додаткових питань на основі мереж Маркова, яка може забезпечити вирішення зазначеного вище протиріччя.

Для рішення завдань тестування обраний метод адаптивного тестування Г. Раша. Визначені методи статистичної обробки результатів тестування й побудови функції розподілу ймовірностей правильних відповідей.

## 4 ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Основні методи побудови алгоритмів

Вибір алгоритму рішення завдання є частиною рішення проблеми побудови й реалізації моделі. З наведеного аналізу випливає, що моделі і їх алгоритмічна реалізація залежно від підходу до побудови адаптаційного тестування як методу суттєво відрізняються.

Існують універсальні технології побудови моделей систем і їх програмної реалізації, такі як Rational Rose, методи екстремального програмування, які дозволяють:

- провести аналіз системи (є через ту, яка стане прообразом програмного продукту);
- виділити учасників процесу, визначити їхні ролі (скласти суб'єктивні представлення про частини системи);
- визначити взаємодія між зазначеними суб'єктами і їх відповідальності, що дозволить вибрати необхідні алгоритми їх поведінки;
- визначити зв'язки й відносини між суб'єктами;
- на підставі певних властивостей поведінки, відносин і зв'язків суб'єктів розробити програмну реалізацію системи в цілому, застосувавши там, де це необхідно адекватні алгоритми, що моделюють як поведінка окремих об'єктів, їх зв'язків і відносин так і системи в цілому.

Крім того, при виборі алгоритмів доцільним є застосування принципу розумної достатності, тобто адекватності алгоритмів рішення поставленого завдання з мінімумом використання ресурсів. Це не виключає застосування складних алгоритмів, а ставиться тільки до обмеження необґрунтованого ускладнення завдання.

Зазначені технологи, як правило, мають справа зі сценарними описами моделі системи, тобто фактично ґрунтуються на суб'єктивних виставах експерта предметної області про те, як це повинне функціонувати. Формалізовані методи

аналізу дозволяють розділити як декомпозицію системи, так і зазначені відносини й зв'язки, робити класифікацію.

#### 4.2 Моделювання рівномірного розподілу питань

Для того, щоб змоделювати випадковий вибір основних питань при тестуванні використовується аналогія їх із класичними квитками на іспиті. Очевидно, що в цьому випадку вибір будь-якого основного питання заданого рівня складності є рівно-ймовірним, тобто підкоряється рівномірному розподілу ймовірностей. Для моделювання рівномірного розподілу обраний метод лінійного конгруентного генератора псевдовипадкових чисел (ГПВЧ) 1-го порядку. Вибір ГПВЧ 1-го порядку обумовлений тим, що в рамках даного дослідження послідовність неповторювальних чисел довжиною 999 562 елемента є достатньою. У випадку, коли це не так (розвинена система тестування) застосування ГПВЧ більш високого порядку не вносить принципових змін і принципово не відрізняється від обраного.

Алгоритм складається з таких кроків.

Крок1. На першому кроці визначається випадкове ціле число, що відповідає в загальному випадку кроку  $n-1$  (позначимо змінну, відповідну до цього числа як  $W_{n-1}$ ). Порядок рандомізованого вибору  $W_{n-1}$  буде визначений у наступному розділі. Значення  $W_{n-1}$  множиться на константу  $k_1$  і ділиться на іншу цілу константу  $p$ . Залишок від розподілу є цілим числом, яке вибирається в якості випадкового цілого на кроці  $n$ , тобто  $W_n$ :

$$W_n = k_1 * W_{n-1} \pmod{p}.$$

Випадкова змінна  $X_n$  рівномірно розподілена на одиничному інтервалі визначається як:

$$X_n = \frac{W_n}{p};$$

Відповідно номер основного питання  $n_q \in (0, \dots, N_q)$  можна визначити як

$$n_q = X_n \cdot N_q,$$

при цьому вибирається найменше найближче ціле число як номер питання.

Прийнята анотація безлічі питань  $n_q \in (0, \dots, N_q)$  обумовлена відповідністю представлення нумерації масивів в обраному (3++) мові програмування.

Крок 2. Перевірка  $W_n \geq 2$ , якщо  $W_n < 2$ , то  $W_n = 2$ .

Крок 3. Здійснюється перевизначення  $W_n = W_{n-1}$  і перехід до кроку 1, у випадку, якщо це за якимись причинами необхідно, інакше, — закінчення роботи.

Рекомендоване значення  $P = 999563$ ,  $k = 470\ 001$ .

Інструментальні засоби розробки ПЗ як правило мають у тому або іншому виді генератор псевдовипадкових чисел, у цей час, заснований на усіканні представлення поточного часу системним таймером. Неточність у виборі коефіцієнтів ГПВЧ (конгруентного) може призвести до далеко не рівномірного розподілу випадкової величини.

Визначення деякого початкового значення  $W_{n-1}$  для системи тестування як деякої заданої константи з наступним збереженням останнього значення після сеансу тестування як початкового для наступного сеансу проблематично, тому що після перезавантаження ПЗ буде отримана щораз повторювана послідовність питань тесту (якщо ця множина і її структура не змінювалася), що неприпустимо.

У даному розгляді має місце компроміс, що полягає в тому, що початкове значення  $W_{n-1}$  для сеансу тестування береться з використанням пропонованого в інструментальних середовищах розробки ПЗ генератора випадкових чисел (рівномірність розподілу некритична), а у випадку вибору основних питань і генерації псевдовипадкових чисел, відповідних до експонентного розподілу.

Прийнятний сценарій проведення іспиту передбачає:

- вибір додаткового питання або безлічі питань у випадку невірної відповіді на основне тестове завдання;

- вибір тем з наступним вибором основних питань по цих темах.

На цьому етапі не розглядаються які-небудь умовні переходи, а визначається модель стохастичних переходів і взаємодій. Для ясності в розумінні моделі, розглядається модель на момент початку тестування, тобто тоді, коли ще немає заборони для переходу до деяких питань, які були вже задані.

### 4.3 Моделювання переходів у Марківських мережах

Стохастичний вибір тем лекційного курсу обумовлений тим, що теми, як правило, не є рівнозначними. Інакше кажучи, теми можуть відрізнятися по кількості розглянутих питань, за рівнем складності теми в цілому й, як наслідок, питань теми, по важливості теми й по інших специфічних ознаках, суб'єктивно обумовленим викладачем.

Вибір питання (у рамках наведеного вище сценарію) здійснюється через вибір теми. Вибір теми здійснюється стохастично з обліком наведених вище обставин. Доцільним у цьому випадку є введення вагового коефіцієнта  $K_{wt}$  теми таким чином, щоб:

$$\sum_{i=1}^t K_{w_i}(i) = 1,$$

де  $T = (1 \dots t)$  визначає  $i$ -ю тему із загального числа  $t$  тем.

У цьому випадку вибір дозволеної теми питання здійснюється в такий спосіб.

Крок1. Для множини дозволених тем з використанням конгруентного генератора псевдовипадкових чисел генерується випадкове  $At^{(i)} = \{0..1\}$  число відповідне до рівномірного розподілу функції ймовірності/

Крок2. Визначається для кожної теми добуток  $K_{wt}^{(i)} \cdot At^{(i)}$ .

Крок3. Вибирається тема (з якої потім буде обрано питання) за ознакою найбільшого значення добутку  $K_{wt}^{(i)} \cdot At^{(i)}$ .

Аналогічним образом можуть вибиратися питання для кожної теми.

Вибір додаткового питання або їх ланцюжка вимагає іншого підходу. Точно також, як і для тем і основних питань існують переваги до їхнього вибору (відсутність переваг може розглядатися як їхня рівність). Однак, якщо при здачі реального іспиту вибір питань здійснюється практично миттєво й визначається рівномірною функцією розподілу, то для додаткових питань існують наступні правила:

- вони задаються по одному, тобто ситуація, коли задане більш одного питання одночасно неможлива;
- вибір додаткового питання здійснюється по розсуду викладача, який має по тем або іншим причинам певні переваги в цьому виборі.

Зазначені переваги виражаються в частоті питань, що задаються екзаменатором, за одиницю часу. Це призводить до визначення подібного роду вибору як найпростішого пуасонівського потоку подій, а процедури вибору – як моделі Марківських мереж.

Таким чином, для вибору додаткового питання метод представляється в такий спосіб.

Крок1. Для кожного додаткового питання, яке може бути як один, так і кореневим у ланцюзі (дереві) додаткових питань за допомогою конгруентного генератора псевдовипадкових чисел визначається число  $A_t^{(i)} \in \{0..1\}$ .

Крок2. По заданій для кожного додаткового питання  $\lambda_{(t)}^{(i)}$  визначається число  $b_{(t)}^{(i)}$  [19]:

$$b_{(t)}^{(i)} = - \frac{1}{\lambda_{(t)}^{(i)}} \ln(A_t^{(i)}).$$

Крок 3. Вибирається той додаткове питання, для якого значення  $b_{(t)}^{(i)}$  є найменшим.

Значення  $b_{(t)}^{(i)}$  при обґрунтованому  $\lambda_{(t)}^{(i)}$  відповідає часу, через який буде задано додаткове питання.

Наведений алгоритм із ваговими коефіцієнтами для тем і основних питань є доречним, а результат у двох випадках практично однаковий незважаючи на

різницю в шкалах оцінки (нормальну й логарифмічну). Однак у першому випадку ( для тем і основних питань) фактично маємо справа з об'єднаною моделлю формування квитків традиційного іспиту. У другому випадку маємо справа з моделлю додаткового опитування.

При тестуванні й верифікації тесту як продукту зазначений роздільний підхід дозволить оцінити різницю витрат часу при автоматизованім тестуванні й з іншої сторони може бути використаний для прогнозування традиційного іспиту як частина моделі, що визначає час, який затрачається на вибір додаткових питань.

#### 4.4 Реалізація модифікованого алгоритму вибору тестових завдань

Використання мережі Маркова, припускає наявність стохастичної моделі системи як абстракції (у загальному випадку ця модель може бути як кінцевої, так і нескінченної), яка має сукупність станів  $S = \{s_1, \dots, s_n\}$ . Ці стани є взаємовиключними й перехід з одного стану в інше здійснюється різними для системи шляхами, через появу події, що є в загальному випадку елементом найпростішого потоку подій. Кожний стан системи  $s_i$  повинне мати як мінімум один вхід і один вихід, тобто так звані «глухі» або кінцеві стани відсутні. Фактично будь-який стан системи може бути обране як початкове.

Однак на практиці вибір початкового стану визначається апріорною інформацією. Поведінка таких систем описується як за допомогою стохастичного моделювання так і рішенням рівнянь Колмогорова [23].

Після ініціалізації деякого початкового стану робота моделі, побудованої в зазначених рамках класичної системи мережі Маркова полягає в нескінченному обході мережі станів системи, у яких система перебуває в плинні деякого певного часу.

У атестаційній роботі безпосереднє використання класичного вистава мереж Маркова не забезпечує рішення поставленого завдання, тому що є цілий ряд обмежень логічного характеру, що визначають як перехід зі стану в стан, загибель станів [24], так і обмеження кількості переходів (тестування не може проводитися нескінченно й питання не повинні повторюватися).

Останній метод є кращим, тому що виконується наступна перевірка: чи сформувалася безліч такої ознаки як складність питань для всієї групи випробуваних і одночасно чи сформувалася функція розподілу ймовірності відповідей залежно від їхньої складності для конкретного суб'єкта тестування.

Алгоритм даного методу перевірки полягає в наступному:

Крок1. Проводиться опитування всієї групи випробуваних хоча б по 6-10 основних питаннях ( при цьому може враховується інформації про відповіді на додаткові питання, якщо вони були задані).

Крок2. Після чого будуються три матриці по вибірках відповідей на  $N_1$ ,  $N_2$  і  $N_3$  питань.

Крок3. Наступною дією є угруповання ймовірностей правильних відповідей на запитання за ознакою приналежності до обраних інтервалів складності питань. У кожному інтервалі обчислюється середнє значення ймовірності, яке визначається як відповідне до середнього значення оцінки складності питань у цьому інтервалі.

Крок4. Далі розподіляємо отримані для всіх груп спостереження (ймовірності відповідей на запитання) по зростанню ( тобто привласнюємо ранг). Якщо значення збігаються, то їм привласнюється середнє значення з рангів, які вони одержали б при відсутності збігів.

Крок5. Визначається критерій Краскала-Уоліса [24]:

$$H = \frac{12}{n(n+1)} \cdot \left( \frac{R_1^2}{N_1} + \frac{R_2^2}{N_2} + \frac{R_3^2}{N_3} \right) - 3 \cdot (n+1),$$

де  $H$  – критерій Краскала-Уоліса;

$n$  – сума питань у трьох вибірках;

$R$  – сумарний ранг питань для однієї вибірки;

$N$  – кількість питань в одній вибірці.

Крок6. По відповіднім до розподілу Краскала-Уоліса визначаємо верхню 5% межу критичного довірчого інтервалу [25] у випадку малих вибірок ( $N_1 \leq 4, N_2 \leq 4, N_3 \leq 5$ ), а у випадку більших вибірок застосовано критерій  $\chi^2$ .

Крок7. У випадку влучення обраного критерію в межі 5% критичного довірчого інтервалу (підтвердження гіпотези  $H_0$ ) – тестування припиняється, а якщо ні, то (підтвердження гіпотези  $H_1$ ) тестування триває (задаються наступні питання) і здійснюється перехід на Крок2.

Схему даного алгоритму наведено на рис 4.4.

#### 4.5 Алгоритм опитування

Описаний пірамідальний алгоритм опитування студентів, реалізація якого відповідає моделі Г. Раша.

Крок 1. Будується сітка питань для кожного студента.

Крок 2. Визначається початковий рівень складності кожного завдання.

Крок 3. Стохастично вибирається основне питання заданого рівня складності.

Крок 4. Обране опитування з варіантами відповідей передається для обробки студентові.

Крок 5. Контролер системи опитування одержує результат відповіді студента

Крок 6. Контролер системи опитування зтягає результат відповіді до протоколу.

Крок 7. Задане питання віддаляється з мережі питань для конкретного студента.

Крок 8. Якщо відповідь правильна, то рівень складності наступного питання буде на одиницю більше, а якщо ні, то на одиницю менше.

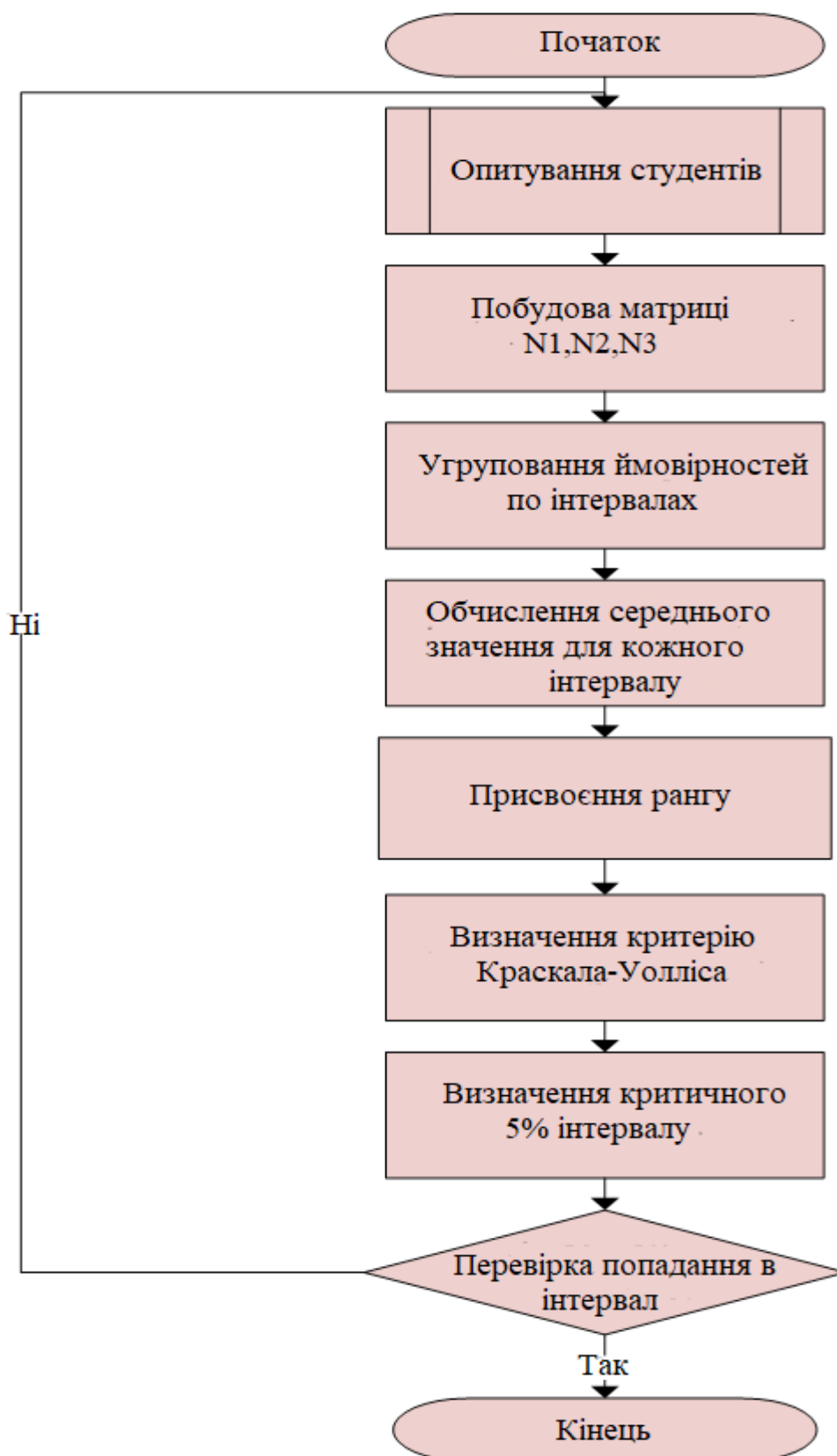


Рисунок 4.4 –Схема алгоритму статистичної обробки

Крок 9. Здійснюється перевірка кількості даних відповідей. Якщо їх більше 10, то Крок 10, інакше перехід на Крок 3.

Крок 10. Якщо перевірка тестування за критерієм Краскала-Уолиса позитивна, тобто функція розподілу ймовірності змінюється не суттєво, то закінчення опитування, інакше перехід на Крок 3.

Схема даного алгоритму наведено на рисунку 4.5.

#### 4.6 Проектування прототипу ПЗ

Метою розробки прототипу програмного забезпечення є обґрунтування переваг застосування запропонованого методу формування тесту на основі мережі Маркова з елементами мереж Петри, що включає в себе додаткові питання як частина основних питань і статистичну обробку, певну для малих вибірок даних при невідомому виді функції розподілу ймовірностей правильної відповіді з її наступної безпосередньою апроксимацією.

Розроблювальний прототип програмного забезпечення тестування знань повинне здійснювати наступні функції:

- керування багато-віконним інтерфейсом;
- введення параметрів програми тестування;
- діапазон рівнів підготовки студентів (авторитарний і латентний);
- кількість основних питань;
- діапазон рівнів складності питань (латентний, що й задається).
- діапазон кількості тем;
- діапазон рівнів складності додаткових питань;
- діапазон оцінки переваги додаткових питань;
- кількість додаткових питань, вибір яких здійснюється шляхом логічного висновку (у частках від загальної кількості додаткових питань);
- діапазон часу.

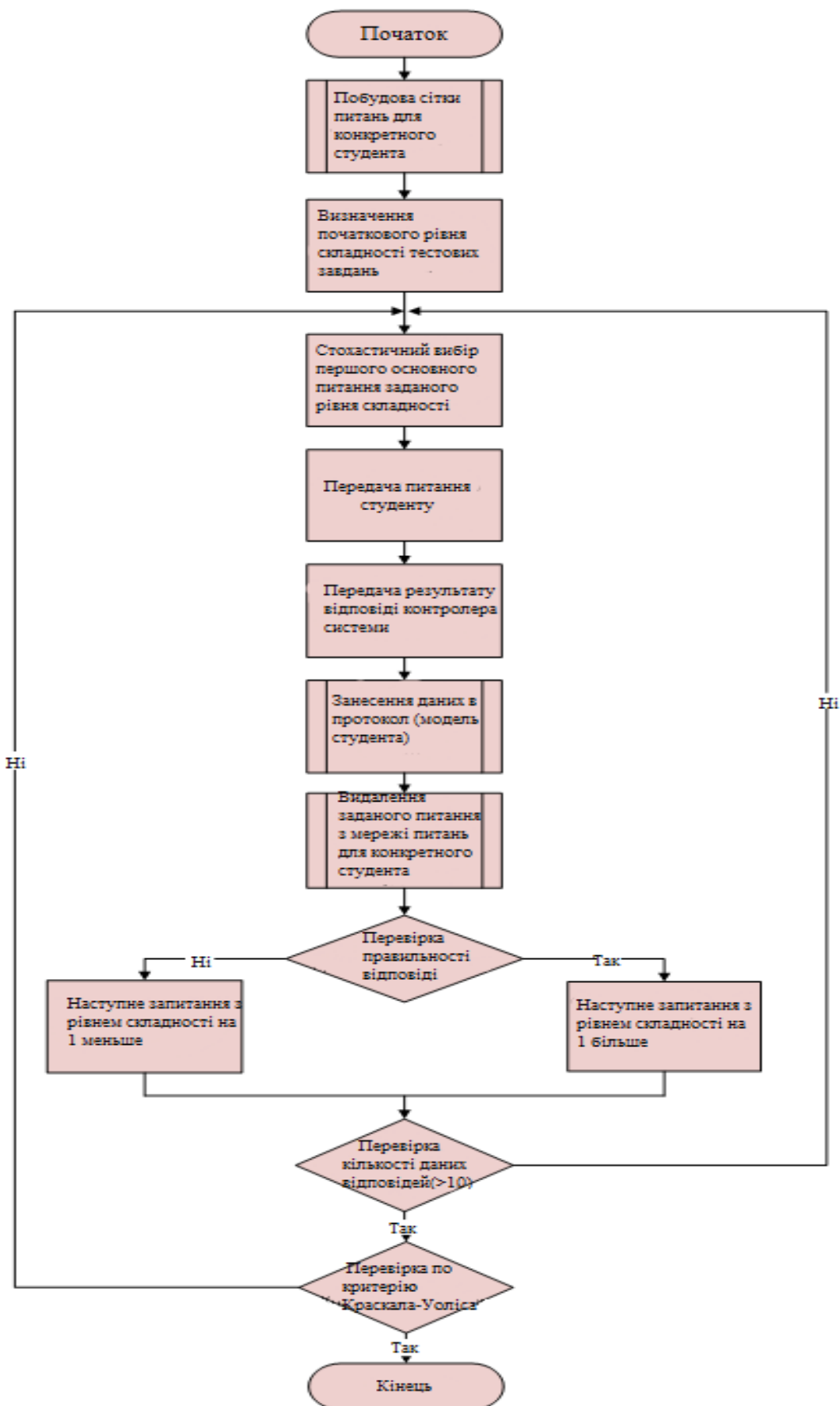


Рисунок 4.5 –Схема алгоритму опитування

Вихідна інформація має виконувати такі функції:

- зміна показників логита рівня підготовки як функції залежно від кількості заданих при тестуванні питань;
- функція ймовірностей Г. Раша для кожного випробуваного за результатами тестування;
- зміна показників функції складності питання для кожного обраного питання;
- функція складності для кожного обраного питання для середнього рівня підготовки випробуваного за результатами тестування;
- протокол опитування.

Вимоги по алгоритмах і внутрішнім методам:

- стохастичне визначення властивостей об'єктів залежно від обраного діапазону;
- стохастичний вибір застосування логічного правила переходу від додаткового питання більш високого рівня складності до більш низького, якщо це правило визначене;
- статистичний аналіз, що включає критеріальну оцінку закінчення тестування;
- апроксимацію логістичної функції ймовірності й логістичної функції складності завдання.

Генерація мережі тестових завдань:

- введення кількості завдань;
- введення кількості варіантів відповідей;
- введення теми.

Одна з основних проблем інженерії програмного забезпечення – різкий ріст сумарних витрат при завищенні вимог до якості програмного продукту. Зміна вимог до програми може відбуватися не тільки на стадії проектування, але й на стадії супроводу – давно відома проблема замовника, який остаточно усвідомлює необхідну функціональність програми в міру введення її в експлуатацію. У цьому випадку є небезпека сполучення процесу розробки ПО (створення програмного продукту) і процесу його підтримки, який характерний для експлуатації ПО, що

не припустиме. Крім цього існує вимога створення програмного забезпечення з можливістю внесення змін у програму, не міняючи раніше написаного коду. Одне з рішень даної проблеми дає технологія проектування й розробки програмних систем методами об'єктно-орієнтованого програмування (наприклад, RUP), що реалізує концепцію повної підтримки ПО на всіх стадіях проектування й експлуатації.

Складність ПЗ й невизначеність вимог замовника безпосередньо впливають на трудомісткість проекту. Завдання розроблювача – використовувати найбільш ефективну технологію проектування й розробки ПО. При цьому звичайно, більш проста технологія ефективніше в завданнях меншої складності, з добре певними, фіксованими вимогами. Правильний вибір можна зробити тільки за допомогою методик експертного техніко-економічного обґрунтування проектів програмних засобів.

Без достовірної інформації про розмір (складності) завдання досить важко оцінити обсяг працевитрат, необхідних для її виконання, тому вимір розміру (складності) розроблювальної програми передуює техніко-економічній оцінці. Недостатньо достовірні оцінки тягнуть проблеми взаємодії розроблювача із замовником і збільшують ступінь ризику проекту.

Статистика за вихідними даними реальних завершених розробок для техніко-економічної оцінки збирається, накопичується й обробляється з початку 70-х років [25]. Ці дані дозволяють одержувати й прогнозувати основні узагальнені показники процесів розробки ПЗ. Раніше почали створюватися й обґрунтовуватися різні методи аналізу трудомісткості розробки, серед яких зараз виділяють [22]-[23]:

- розмірно-орієнтовані метрики, які вимірюють ПП і процес його розробки й ґрунтуються на Лос-оцінках (Lines Of Code — кількість рядків у ПЗ);
- орієнтований<sup>^</sup>-орієнтовані-функціонально-орієнтовані метрики, які розглядають не розмір, а функціональність або корисність продукту;
- конструктивна модель вартості (COCOMO), яка використовує статистичний підхід, ураховує реальні результати величезної кількості проектів;

- структурні й модульні метрики, які дозволяють оцінити складність ПО як системи й прогнозувати кількість помилок у ПО послові закінчення його розробки [14].

#### 4.7 Архітектура розробленого ПЗ

Розробка даного ПЗ зводиться до проектування моделі ПЗ чисельного експерименту, що включає статистичні методи обробки. Вибір складових частин цієї моделі залежить від успішного застосування математичних методів, на яких вони засновані. Таким чином, як будь-яке завдання дослідження, розробка ПЗ в цьому випадкові погано прогнозована з погляду визначення включених у неї модулів на ранні й більш пізніх стадіях проектування. Тому прогнозування складності розроблювального ПЗ практично неможливо на підставі детермінованих моделей попередньої оцінки складності ПЗ. У зв'язку із цим для попереднього аналізу цієї оцінки була обрана наступна процедура:

- розроблено окремий модуль середньої складності (клас `gener`);
- оцінена його складність із погляду наявності помилок після розробки даного модуля.

Використання мовно-орієнтованих засобів і інтегрованих пакетів у цьому випадку є ризикованим тому що відсутність реалізації деяких методів, необхідність у яких може виникнути при розробці ПЗ може привести до непередбачених результатів (неможливість, наприклад, реалізації вбудованою мовою програмування).

По зазначених причинах була застосована методика оцінки складності ПЗ по структурних і модульних метриках. Враховуючи важливість відсутності помилок, як вимога до розроблювального ПЗ, яке важливо для доказової бази запропонованих методів, обрана технологія ООП, мова програмування C++ і середовище Microsoft Visual Studio.Net.

На етапі архітектурного проектування визначається перелік компонентів ПО і їх інтерфейси взаємодії для встановлення загальних рамок інтегрованого середовища розробки ПЗ. Для охоплення вимог, пропонує до програмного продукту, найкраще використовувати діаграми варіантів використання. Кожна умова використання описує ряд взаємодій системи й навколишнього середовища. Набір умов використання включає операції, які повинна виконувати система, щоб задовольнити потреби діючих суб'єктів, що використовують систему для виконання своїх ділових завдань. Діаграма варіантів використання представлено на рисунку 4.6.

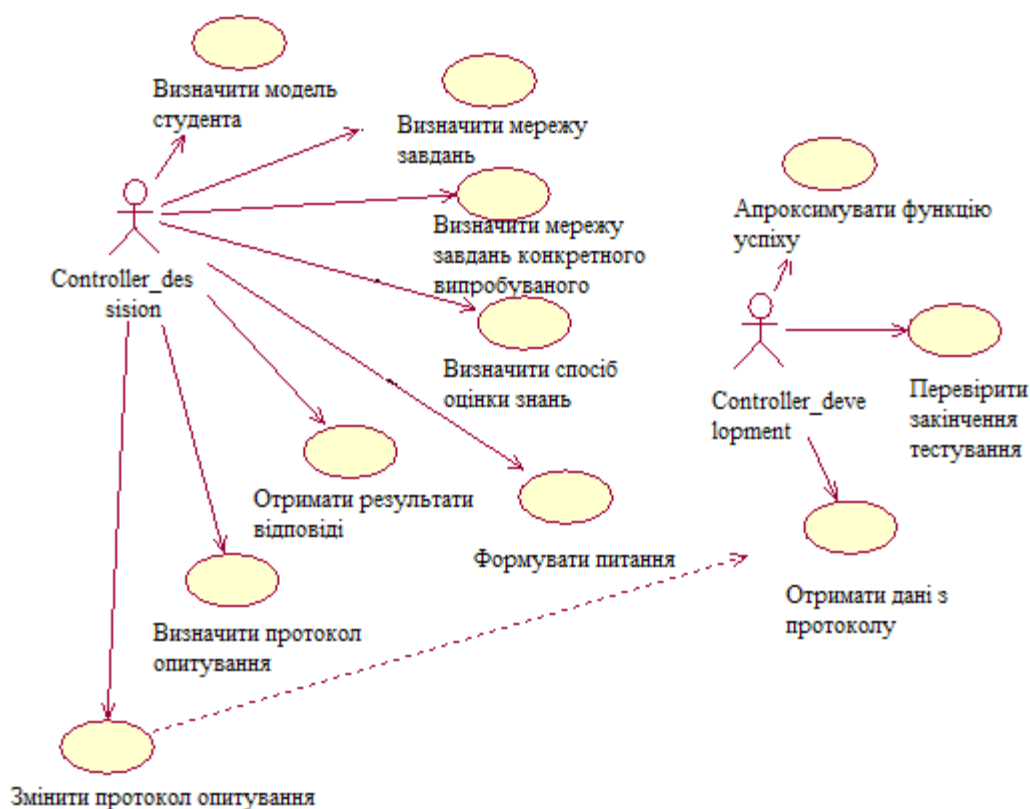


Рисунок 4.6 – Діаграма варіантів використання

Сценарій роботи системи наведено нижче.

Робота починається із використання класу `Controller_developer`, що має такі функції:

- створює задану кількість об'єктів класу `Model_student`;

- створює задані кількості основних і додаткових завдань класу Questions\_cont;
- створює об'єкт класу Definer\_dessision результат, що визначає алгоритм і дані як, відповіді об'єкта класу Model\_student на завдання класу Questions\_cont;
- передає керування об'єкту класу Net\_server.

Об'єкт класу Net\_server створює об'єкти класу (ідентичні на початку сесії тестування) Students\_own\_net для всіх об'єктів класу Model\_student.

У процесі роботи з об'єктом класу Controller\_dessision випадковим чином вибирається об'єкт класу Model\_student і з відповідної йому мережі завдань (об'єкт класу Students\_own\_net) також випадковим чином вибирає завдання, – об'єкт класу Questions\_cont (основне або додаткове з гілок додаткових завдань, що належать основному, – тільки у випадку невірної відповіді на основне завдання). Після цього передає керування об'єкту класу Definer\_dessision, що визначає відповідь обраного об'єкта класу Model\_student на обране завдання класу Questions\_cont.

Об'єкт класу Controller\_dessision передає керуванні об'єкту класу Net\_server, який видаляє обране й оброблене завдання класу Questions\_cont з відповідної до обраного студента мережі завдань об'єкту класу Students\_own\_net (у випадку, якщо завдання було додатковим, видаляється вся гілка додаткових завдань, у яку входить додаткове завдання і яка пов'язана з основним питанням).

Керування від об'єкта класу Net\_server передається об'єкту класу Controller\_dessision, який у свою чергу створює (якщо він ще не створений) і модифікує об'єкт класу Protokol\_answeres, вносячи туди всі необхідні дані про результати тестування.

Об'єкт класу Protokol\_answeres перевіряє, чи зібрана достатня статистика для проведення статистичного аналізу, і якщо так – те передає керування об'єкту класу Controller\_Development.

Об'єкт класу Controller\_Development передає об'єкт класу Protokol\_answeres об'єкту Estimator\_Krask\_Wol (а якщо він не створений – і створює його), який робить оцінку по визначенню достатності даних для статистичної обробки

функції успіху Г. Раша. Якщо так, то керування передається об'єкту класу `Approximator_Rash` (якщо він не створений — і створює його), який і робить цю оцінку й завершує виконання програми, якщо немає — через Об'єкт класу `Controller_Development` в об'єкт класу `Protokol_answers`.

Об'єкт класу `Protokol_answers` передає керування об'єкту `Controller_dessision`, який повторює процедуру опитування.

Проектоване програмне забезпечення ґрунтується на складанні по-операційного переліку робіт у рамках розглянутого проекту. Спочатку вводиться інформація про проект (характеристики проекту). У свою чергу кожне завдання теж має рядом параметрів для реалізації методів оцінки. Виходячи із цього на рисунку 4.7 представлена діаграма класів розроблювального програмного забезпечення.

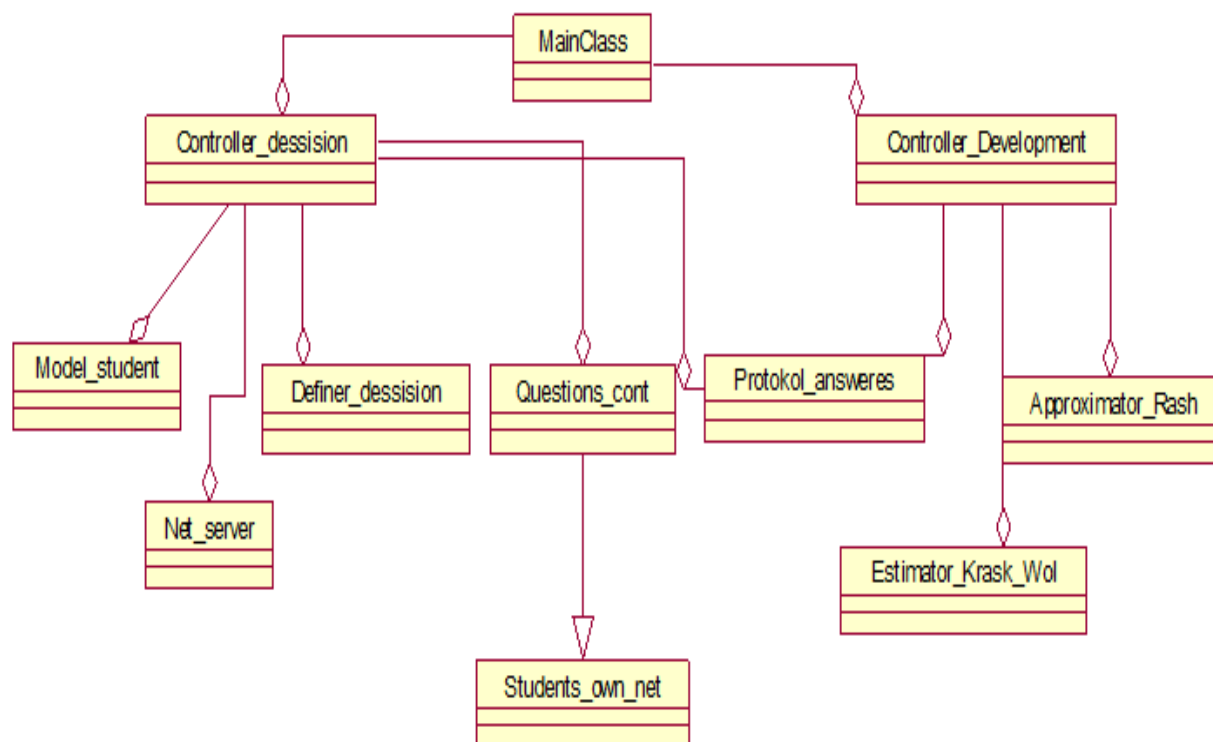


Рисунок 4.7 – Діаграма класів

На рисунку 4.8 описано схему взаємодії логічних елементів системи (класів) у вигляді діаграми послідовності.

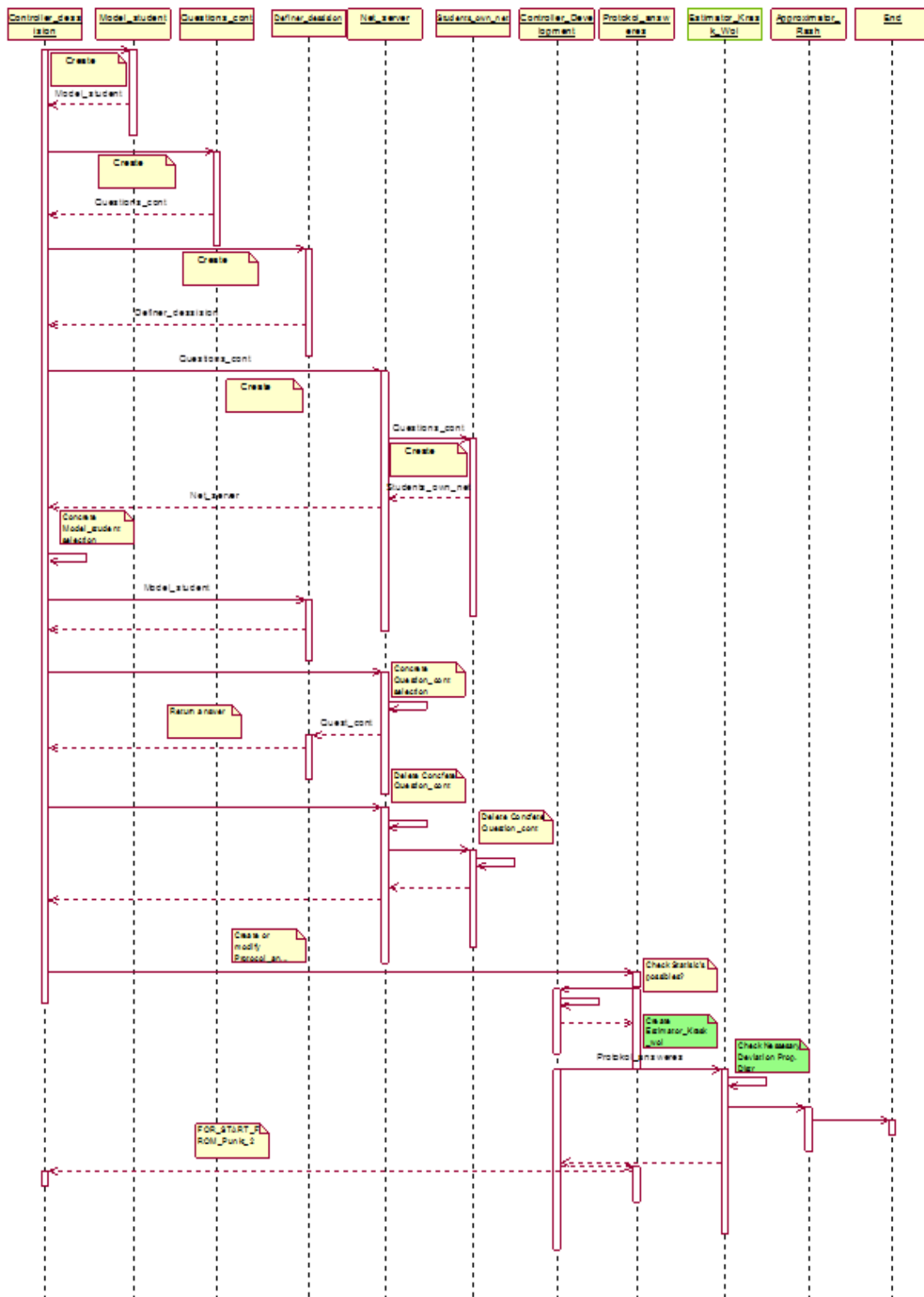


Рисунок 4.8 – Діаграма послідовностей дій

У таблиці 4.1 наведений опис класів.

Таблиця 4.1 – Опис класів

Ім'я класу	Опис
Mainclass	Клас головної форми додатка
Controller_ dessionion	Клас, що забезпечує вибір наступного питання
Controller_ development	Клас створює задана кількість об'єктів класу model_student, задані кількості основних і додаткових завдань класу questions_cont, об'єкт класу definer_dessionion
Model_student	Клас, що визначає заданий і латентний рівень підготовки випробуваних і генеруючий протокол відповідей на видані йому завдань
Definer_ dessionion	Клас, що визначає алгоритм і дані як результат відповіді об'єкта класу Model_student на завдання класу Questions_cont;
Questions_ cont	Клас містить мережі тестових завдань для кожного студента
Protokol_ answeres	Клас містить протоколи відповідей для кожного випробуваного й усієї групи
Approximator_ Rash	Клас, що реалізує апроксимацію функції успіху Г. Раша
Net_server	Клас забезпечує керування мережами тестових завдань для кожного студента
Students_own _net	Клас, що містить мережу тестових завдань для одного студента
Estimator_ Krask_Wol	Клас, що забезпечує оцінку статистичної значимості ряду відповідей випробуваних

Відповідно до завдань проектування ПЗ був розроблений користувацький інтерфейс для введення даних (рисунок 4.9).

The screenshot shows a window titled "Ввод исходных данных" (Input of initial data) with a blue background. It contains the following fields and buttons:

- Диапазон уровня подготовки (латентный): от 1 до 23
- Диапазон уровня подготовки (авторитарный): от 1 до 13
- Диапазон уровней сложности вопросов (латентный): от 1 до 8
- Диапазон уровней сложности вопросов (авторитарный): от 1 до 8
- Количество основных вопросов: от 1 до 2
- Диапазон количества тем: от 1 до 20
- Диапазон уровня сложности доп.вопр.: от 1 до 4
- Количество доп.вопр.: от 1 до 4
- Buttons: "Ввод данных" (Input data) and "Расчитать" (Calculate)

Рисунок 4.9 – Интерфейс проекowanego прототипу ПЗ

Виведення інформації здійснюється шляхом підключення через файл протоколу опитування програми. Висновку графічної інформації здійснювався за допомогою вільно розповсюджуваного спеціального ПЗ.

У процесі розробки ПЗ для виявлення помилок використані наступні види тестування:

- автономні тести забезпечують перевірку конкретних модулів і класів, проводилися методом «білої скриньки»;
- інтеграційними тестами перевірена коректність роботи модулів у багатомодульній програмі;
- системні тести використані для перевірки коректності роботи всієї програми цілком.

## 5 ОПИС МОЖЛИВОСТІ ВИКОРИСТАННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Метою автономного тестування є порівняння функцій, реалізованих у модулях, зі специфікаціями. Автономні тести повинні показати, чи виконують функції те, що передбачається, потрібним образом.

Об'єктами тестування клієнтської частини є всі модулі додатка й методи класів.

Для проведення тестування обраний метод «білого ящика», який дозволяє досліджувати внутрішню структуру програми.

Усі протестовані модулі класів написані коректно, виявлені помилки були виправлені. Найбільше кількість помилок була знайдена в модулі Dession який можна вважати найбільше потенційно небезпечним в експлуатації.

Послідовність тестування: спочатку тестується бізнес-логіка додатка, потім – клієнтська частина додатка.

Мета тестування бізнес-логіки додатка:

- перевірка сумісності системи;
- перевірка можливості витягу даних про аналізований файл.

Тестування на контрольних прикладах є невід'ємною частиною розробленого ПЗ. Так чи інакше при роботі з одними з вихідних даних є латентні (сховані) рівні складності питань або рівні підготовки випробуваних.

Метою розробленого ПО є доказ збіжності рішення від авторитарного рівня до латентного (або підготовки випробуваного або складності питань).

Побудова мережі завдань із додатковими питаннями здійснюється розробленим ПЗ стохастичним способом, тобто припускає дослідження рішень на безлічі випадково побудованих семантичних мереж, що підкоряються правилам мереж Маркова (мережа повинна бути замкнена, не мати тупикових станів і визначати набір повних станів системи).

Показником валідності системи є проведення кількарізкових випробувань прототипу ПЗ на неповторюваних, що генерується нею мереж завдань із заданими розумними обмеженнями такими як:

- кількість тем від 1 до 20;
- кількість основних завдань у темі від 1 до 20;
- кількість дерев додаткових завдань основного питання від 0 до 7;
- загальна кількість додаткових питань не більше 200;
- кількість рівнів додаткових питань у дереві від 1 до 4;
- кількість додаткових питань у кожному рівні від 1 до 4;
- кількість варіантів відповідей на завдання від 3 до 7;
- кількість рівнів складності тестових завдань від 1 до 8;
- кількість випробуваних до 100 чіл.

Під час тестування прототипу ПЗ несподіваних помилок виконання (завершення) роботи не виявлене.

Авторитарні рівні (або підготовки випробуваного або складності питань) сходилися до близьких значень латентних.

Обмеження у вихідних даних фактично продиктовані результатами налагодження й тестування програми, тому що при більших значеннях вихідних даних, чому зазначені вище були потрібні такі обсяги оперативної пам'яті й часу обробки, які можна оцінювати як критичні.

На рисунку 5.1 зображений графік збіжності рівнів підготовки випробуваного від авторитарного до латентного з використанням додаткових питань і без додаткових питань.

Виходячи з вищевикладеного, враховуючи збіжність завдання до свідомо відомих рішень по латентних ознаках, можна зробити висновок про надійну роботу прототипу ПЗ.

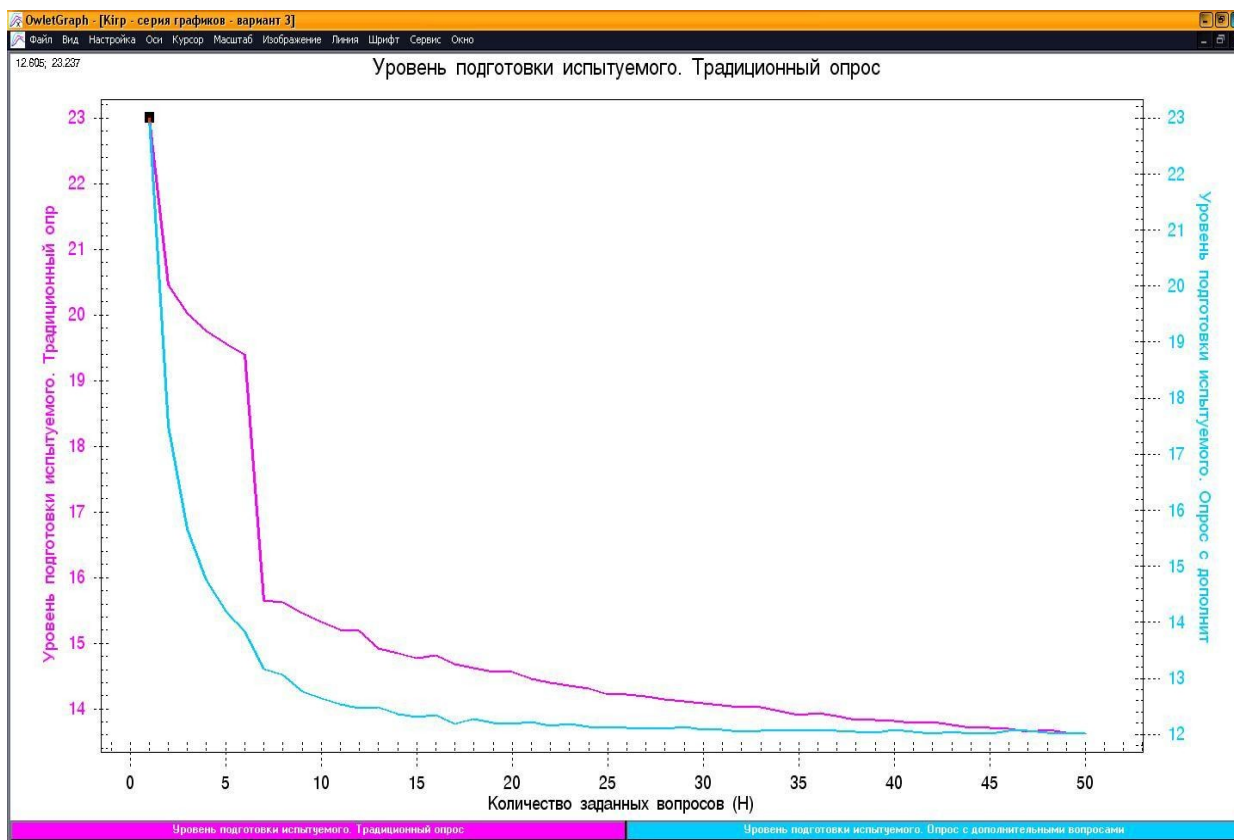


Рисунок 5.1 – Графік збіжності рівнів підготовки випробуваного від авторитарного до латентного з використанням додаткових питань і без додаткових питань

На рисунку 5.2 зображено екранну форму практичного результату апроксимації функції успіху від логіту складності одного випробуваного й експериментальні дані роботи програми.

Таким чином, в рамках дослідної експлуатації розробленого ПЗ були розглянуті найбільш важливі алгоритми для рішення поставленого завдання:

- алгоритм конгруентного генератора псевдовипадкових чисел;
- алгоритм вибору дозволеної теми питання алгоритм вибору додаткового питання (графа потоків робіт);
- алгоритм статистичної обробки;
- алгоритм опитування.

Вищеописані алгоритми є досить простими й легко реалізованими й використані при проектуванні сценарію адаптивного тестування, не дивлячись на те, більшість із них є традиційними.

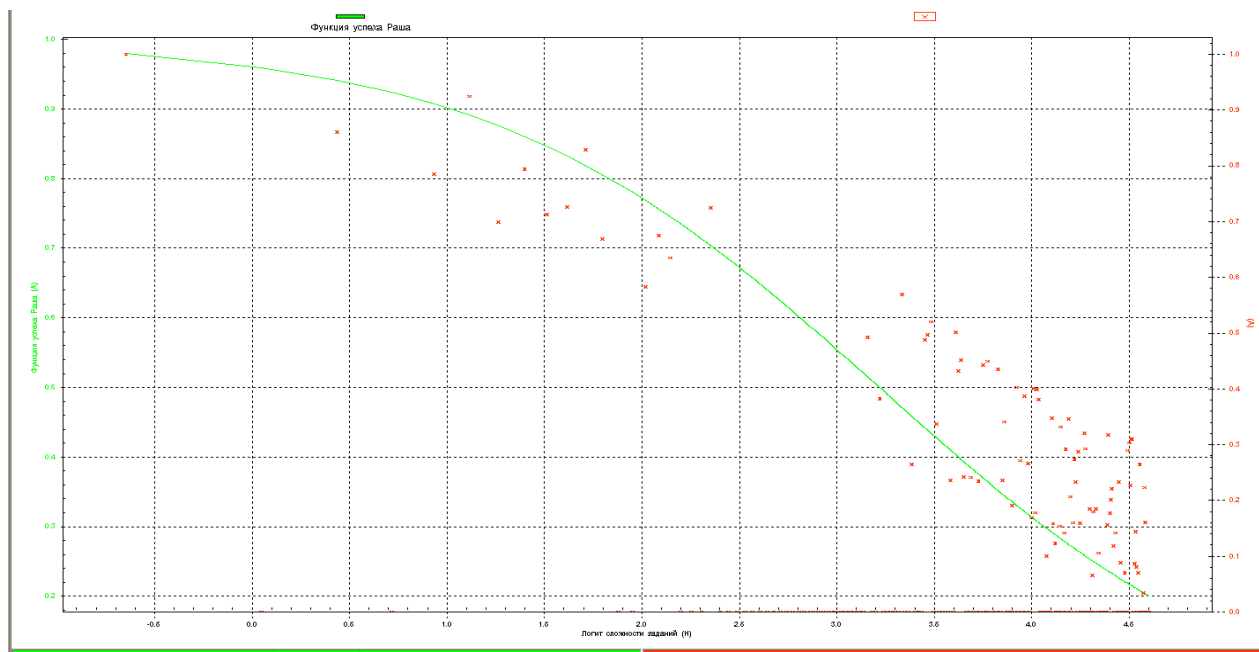


Рисунок 5.2 – Апроксимація функції успіху від логіту складності завдання одного випробуваного (експериментальний результат)

Були сформульовані вихідні дані на розробку прототипу програмного забезпечення.

У якості засобу розробки прототипу ПЗ використана технологія ООП. Застосування об'єктно-орієнтованого підходу дозволило реалізувати базові методи об'єктів у батьківських класах і, при необхідності, перевизначати й розширювати в класах спадкоємцях.

Об'єктна модель додатка представлена 11-ю класами, реалізованих в 4 модулях.

Розроблений прототип ПЗ був протестований на предмет працездатності й рішення поставленого завдання, тестування проводилося на контрольних прикладах і показало коректну працездатність розробленого ПЗ.

На підставі проведення перерахованих вище тестів можна укласти, що створений прототип виконує всі функції, зазначені у вихідних даних на розробку й відповідає всім пред'явленим до нього вимогам.

## ВИСНОВКИ

В результаті виконання магістерської роботи показано перевагу запропонованого методу побудови ПЗ адаптивного тестування на базі мереж Маркова з елементами мереж Петрі при використанні концепції основних і додаткових питань.

Різниця в рівнях складності основних і додаткових питань і здійснюваний зв'язок між основними й гілками додаткових питань дозволяє в процесі тестування як мінімізувати кількість необхідних відповідей тих, яких навчають, для визначення рівня їх знань, так і суттєво поліпшити адаптаційні властивості тестування.

Кращі результати досягнуті тоді, коли різниця між передбачуваним і дійсним рівнем знань групи тих, кого навчають, і передбачуваної й дійсною складністю завдань істотна.

Однак, при невеликій різниці між зазначеними передбачуваними й дійсними властивостями групи випробуваних застосування зазначеної методики дає позитивний ефект, із чого можна зробити висновок, що застосування запропонованого методу є ефективним для будь-яких обсягів і інших властивостей, що визначають тестування як процес.

Збільшення складності алгоритму тестування не приведе до яких або істотним витрат ресурсів ПК.

Слід зазначити, що результат по використанню зазначеної методики стосовно до 3-х параметричної моделі тестування Г. Раша довів, що апроксимація функції успіху, що припускає визначення уточнених значень дискримінаторів завдань шляхом рішення завдання оптимізації при використанні відомих алгоритмів визначило більші витрати машинного часу. Це пов'язане насамперед з тим, що застосовані алгоритми оптимізації є універсальними, а рішення виниклої проблеми вимагає додаткового дослідження метою якого є часткове заміщення

цих алгоритмів операційними методами, що використовують як властивість поставленого завдання тестування так і іншої апріорної інформації.

Застосування об'єктно-орієнтованого підходу дозволило реалізувати базові методи об'єктів у батьківських класах і, при необхідності, перевизначати й розширювати в класах спадкоємця. Об'єктна модель додатка представлена 11-ю класами, реалізованих в 8-мі модулях.

Експериментальне підтвердження засноване на множині тестових завдань, що імітують умову проведення тестування при різних вихідних даних про апріорні властивості групи суб'єктів тестування. Для цього було створено програмне забезпечення, що забезпечує постановку чисельного експерименту на моделях тестування.

Розроблене ПЗ було протестовано на предмет працездатності й рішення поставленого завдання, тестування проводилося на контрольних прикладах з відкритих джерел і показало коректну працездатність розробленого ПЗ. На підставі проведення перерахованих вище тестів можна зробити висновок, що створений прототип виконує всі функції, зазначені у вихідних даних на розробку й відповідає всім пред'явленим до нього вимогам.

У цілому, зазначене дослідження дозволяє зробити висновок про перспективність проведення робіт у цьому напрямку для одержання ще більш значимих результатів.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Learning management system // URL: [https://en.wikipedia.org/wiki/Learning\\_management\\_system](https://en.wikipedia.org/wiki/Learning_management_system) (дата звернення: 26.04.2020).
2. Бейкер, Р. Educational data mining and learning analytics/Р. Бейкер, Г. Сіменс – The Cambridge handbook of the learning sciences, 2019. – 274 с.
3. Ржеуцька С. Опыт применения методов кластеризации для анализа результатов дистанционного обучения // Информатизация инженерного образования: материалы международной науч.-практ. конф. 2016. № 56. С. 617 – 620.
4. Expectation-maximization algorithm // URL: [https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization\\_algorithm](https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm) (дата звернення: 23.03.2020).
5. K-means clustering // Wikipedia. URL: [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering) (дата звернення: 25.04.2020).
6. Кукиер, К. Big Data: A Revolution That Will Transform How We Live, Work, and Think/К. Кукиер, В. Штойнберг, 2017. – 236 с.
7. Лам Ч. Hadoop in Action/Ч. Лам, 2015. – 365 с.
8. Дашкевич О. Аналіз можливостей Apache Kafka в рамках забезпечення стримінгу Big Data // Информационные системы и технологии: материалы 7-й Международ. науч.-техн. конф. 2018. № 12. С. 34-35.
9. Data Mining with WEKA MOOC – Material // Machine Learning at Waikato University. URL: <https://www.cs.waikato.ac.nz/ml/WEKA/mooc/dataminingwithWEKA> (дата звернення: 26.03.2020).
10. A Simple Step by Step Guide to WEKA // Analyticscosm. URL: <https://analyticscosm.com/a-simple-step-by-step-guide-to-WEKA/> (дата звернення: 26.03.2020)

11. ECTS users' guide 2015. URL: <https://publications.europa.eu/en/publication-detail/-/publication/da7467e6-8450-11e5-b8b7-01aa75ed71a1> (дата звернення: 26.03.2020)
12. Distance Learning Courses and Adult Education - The Open University. URL: <http://www.open.ac.uk/> (дата звернення: 26.03.2020)
13. Online Diploma Courses – Diplomas. URL: <http://www.openuniversity.edu/courses/programmes/diplomas> (дата звернення: 26.03.2020)
14. Кодекс чести студента НОУ "ИНТУИТ". URL: <https://www.intuit.ru/content/kodeks-chesti-studenta-nou-intuit> (дата звернення: 26.03.2020)
15. OpenTEST – программа тестирования знаний. URL: <http://opentest.com.ua/> (дата звернення: 09.04.2020)
16. Появился вирус, поражающий файлы Macromedia Flash. URL: <https://lenta.ru/news/2002/01/09/macro/> (дата звернення: 11.04.2020)
17. Flash & The Future of Interactive Content. URL: <https://theblog.adobe.com/adobe-flash-update/> (дата звернення: 11.04.2020)
18. Adaptive Learning Systems: Surviving the Storm. URL: <https://er.educause.edu/articles/2016/10/adaptive-learning-systems-surviving-the-storm> (дата звернення: 29.03.2020)
19. Автоматизація. URL: <https://uk.wikipedia.org/wiki/Автоматизація> (дата звернення: 30.03.2020)
20. Самоорганізація. URL: <https://uk.wikipedia.org/wiki/Самоорганізація> (дата звернення: 30.03.2020)
21. Штучна нейронна мережа. URL: [https://uk.wikipedia.org/wiki/Штучна\\_нейронна\\_мережа](https://uk.wikipedia.org/wiki/Штучна_нейронна_мережа) (дата звернення: 23.01.2020).
22. Дерево ухвалення рішень. URL: [https://uk.wikipedia.org/wiki/Дерево\\_ухвалення\\_рішень](https://uk.wikipedia.org/wiki/Дерево_ухвалення_рішень) (дата звернення: 30.03.2020)
23. Умовний перехід. URL: [https://uk.wikipedia.org/wiki/Умовний\\_перехід](https://uk.wikipedia.org/wiki/Умовний_перехід) (дата звернення: 30.03.2020)

**24.** Штучна нейронна мережа. URL: [https://uk.wikipedia.org/wiki/Штучна\\_нейронна\\_мережа](https://uk.wikipedia.org/wiki/Штучна_нейронна_мережа) (дата звернення: 20.04.2020)

**25.** AutoDraw By Google Creative Lab. URL: <https://experiments.withgoogle.com/chrome/autodraw/> (дата звернення: 20.04.2020)

**26.** DeepDream. URL: <https://en.wikipedia.org/wiki/DeepDream> (дата звернення: 20.02.2020)

**27.** This Person Does Not Exist. URL: <https://thispersondoesnotexist.com/> (дата звернення: 23.04.2020)

**28.** A Neural Network for Machine Translation, at Production Scale. URL: <https://research.googleblog.com/2016/09/a-neural-network-for-machine.html> (дата звернення: 20.02.2020)

**29.** Добро пожаловать в блог DeepL! URL: <https://www.deepl.com/blog/20180215.html> (дата звернення: 16.04.2020)

**30.** Кластерний аналіз. URL: [https://uk.wikipedia.org/wiki/Кластерний\\_аналіз](https://uk.wikipedia.org/wiki/Кластерний_аналіз) (дата звернення: 16.03.2020)

**31.** Shostak I., Matyushenko I., Romanenkov Yu., Danova M., Kuznetsova Yu. Computer Support for Decision-Making on Defining the Strategy of Green IT Development at the State Level. In book: Green-IT Engineering: Social, Business and Industrial Applications, Vol. 171. Berlin, Heidelberg: Springer International Publishing, 533–559 (2018), <https://doi.org/10.1007/978-3-030-00253-4>

**32.** Shostak I., Kapitan R., Volobuyeva L., and Danova M., Ontological Approach to the Construction of Multi-Agent Systems for the Maintenance Supporting Processes of Production Equipment. In Proc. : IEEE International Scientific and Practical Conference «Problems of Infocommunications. Science and Technology» (PICS&T-2018). Ukraine, Kharkiv, October 9-12, 2018. P. 209 – 214