

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 ПОБУДОВИ ІНТЕРНЕТ-МАГАЗИНІВ З  
ІНТЕЛЕКТУАЛЬНОЮ ПІДТРИМКОЮ ВИБОРУ ТОВАРІВ**  
(тема)

Виконав: студент 2 курсу, групи ІПЗм-18-2  
спеціальності 121 – Інженерія програмного  
забезпечення  
(код і повна назва спеціальності)

освітньо-наукової програми Інженерія  
програмного забезпечення  
(повна назва освітньої програми)

Швечіков І.С.  
(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 Швечікову Іллі Сергійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів побудови інтернет-магазинів з інтелектуальною підтримкою вибору товарів

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 штучних агентів. 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 алгоритмів штучних агентів	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 магістра: 94 стор., 11 табл., 43 рис., 3 дод., 25 джерел.

ІНТЕРНЕТ-МАГАЗИН, АГРЕГАТ, ГРАФИ, ШТУЧНІ АГЕНТИ, МУЛЬТИАГЕНТНІ ТЕХНОЛОГІЇ.

Об'єкт дослідження – процеси функціонування систем електронної комерції.

Метод – моделі й методи реалізації бізнес процесів в електронній комерції.

Метою є використання агрегатних моделей в організації й функціонуванню інтернет-магазинів та розробка прототипу ПЗ для перевірки запропонованих алгоритмів.

Методи дослідження: агрегативні моделі Бусленко, як засіб опису й аналізу алгоритмів.

В результаті розглянуті алгоритми організації електронної комерції, з використанням агентних технологій в інтернет-магазинах, розроблений прототип програмного продукту, що дозволяє перевірити коректність роботи агентів.

INTERNET SHOP, UNIT, GRAPHS, AGENTS, MULTIAGENT TECHNOLOGIES.

The object of the research is the processes of functioning of e-commerce systems.

The subject of the research is the models and methods of implementing business processes in e-commerce.

The aim of the diploma project is to increase the efficiency of business processes in the organization and operation of an online store for the sale of youth clothing.

Research methods: Buslenko's aggregative models as a means of describing and analyzing algorithms; the theory of graphs in the part of creating an agent interaction model.

The results of the diploma project can be used to improve the efficiency of the organization and functioning of existing online stores.

## ЗМІСТ

Вступ .....	7
1 Сучасний стан систем електронної комерції у світі і в Україні та постановка завдання дослідження .....	9
1.1 Основні принципи організації електронної комерції .....	9
1.2 Класифікація інтернет-магазинів .....	11
1.3 Особливості організації електронної комерції для спеціальних груп користувачів .....	13
1.4 Аналіз засобів організації електронної комерції для обслуговування спеціальних груп користувачів .....	16
1.5 Постановка завдань дослідження .....	19
2 Опис проведених теоретичних досліджень .....	21
2.1 Аналіз методів агрегатного моделювання .....	21
2.2 Аналіз використання агентних систем .....	27
2.3 Аналіз концептуальної моделі Інтернет-магазинів .....	32
2.4 Моделювання агрегатних моделей бізнес-процесів .....	36
2.5 Вибір концептуальної моделі ПЗ .....	42
3 Опис проектування алгоритмів системи.....	45
3.1 Моделі агентів пошуку .....	45
3.2 Модель бізнес-процесів електронного магазину .....	48
3.3 Розробка алгоритму використання агрегативних моделей .....	49
3.4 Проектування БД інтернет-магазину .....	54
4 Опис розробленого програмного забезпечення.....	58
4.1 Обґрунтування вибору алгоритмічних засобів .....	58
4.2 Вихідні дані на розробку прототипу ПЗ .....	60
4.3 Архітектурне проектування ПЗ .....	61
4.4 Опис структурної схеми ПЗ інтернет-магазину .....	63

	6
5 Опис можливості використання отриманих результатів.....	67
Висновки .....	70
Перелік джерел посилання .....	71
Додаток А Програмний код .....	73
Додаток Б Слайди презентації .....	79
Додаток В Апробація результатів роботи.....	93

## ВСТУП

Усе більше й більше людей у повсякденнім житті користуються послугами Інтернету для пошуку інформації, роботи й відпочинку, а також здійснення покупок і спілкування.

Швидко зростаюча інтернет-аудиторія є новим ринком збуту для компаній самого різного профілю. Відсутність географічних бар'єрів для реклами й поширення товарів і послуг залучає в інтернет-бізнес нові підприємства.

Ріст світової аудиторії інтернету випереджає навіть самі свіжі прогнози: так, на початку 2000 року Ovum Research пророкувала, що до кінця 2001 повинне бути близько 330 млн. користувачів, але вже в середині року їх число досягло майже 500 млн. Інтернет-магазини швидше прижилися в тих країнах, де вже давно існує устояна система торгівлі по каталогах. Показники електронної торгівлі вражають. Виторг Amazon в 2008 р. склала 19 млрд дол. По опитуваннях жителів Великобританії, саме через нього вони найчастіше купують музику й відео. Загальний обсяг інтернет-торгівлі в Європейському союзі – 130 млрд євро.

У просторі інтернету на території СНД «піонерам» цього бізнесу довелося кілька років пробивати бар'єр недовіри, перш ніж покупці по гідності оцінили такий спосіб здійснення покупок. Сьогодні труднощі «перехідного періоду» уже за, і відкриття нових, вітчизняних інтернет-магазинів – пересічне явище.

У представників малого й середнього бізнесу є онлайн-версії магазинів великих роздрібних мереж стільникового зв'язку, мереж продажів побутової електроніки ([www.technosila.ru](http://www.technosila.ru); [www.mvideo.ru](http://www.mvideo.ru); [www.megapixel.com.ua](http://www.megapixel.com.ua)) і т.ін.

Однак рост інтернет-торгівлі, поява нових інтернет-магазинів спричиняє збільшення конкуренції на ринку, боротьбу за покупця, за кількість продаваного товару й за престиж компанії.

Кожний електронний магазин прагне збільшити кількість відвідувачів, організувати торгівлю максимально ефективно, щоб кількість продажів

збільшувалася, для цього застосовуються останні розробки в сфері інтернет технологій.

Об'єктом дослідження атестаційної роботи магістра є процеси функціонування систем електронної комерції.

Предметом дослідження – моделі й методи реалізації бізнес процесів в електронній комерції.

Метою роботи є підвищення ефективності бізнес-процесів по організації й функціонуванню інтернет-магазину на основі агрегатної моделі.

Для досягнення мети необхідно розв'язати наступні завдання.

- виконати аналіз поточного стану проблеми організації й функціонування систем електронної комерції;
- виконати огляд і аналіз застосовності існуючих агентних технологій при розробці засобів електронної комерції;
- розробити моделі й методи реалізації бізнес-процесів в електронних магазинах на основі агентної парадигми.
- перевірити теоретичні результати досліджень за допомогою розробленого програмного прототипу.

Методи дослідження: агрегативні моделі Бусленко, як засіб опису й аналізу алгоритмів; теорія графів у частині створення моделі взаємодії агентів.

В результаті удосконалений алгоритм організації й функціонування Інтернет-магазину на прикладі магазину із продажу молодіжного одягу, шляхом застосування агентних технологій при розробці програмних засобів у процесах продажу товарів. Результати атестаційної роботи можуть бути використані для підвищення ефективності організації й функціонування існуючих інтернет-магазинів.

# 1 СУЧАСНИЙ СТАН СИСТЕМ ЕЛЕКТРОННОЇ КОМЕРЦІЇ У СВІТІ І В УКРАЇНІ ТА ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ

## 1.1 Основні принципи організації електронної комерції

Електронна комерція (e-commerce) – це прискорення більшості бізнесопроцесів за рахунок їх проведення електронним образом. У цьому випадку інформація передається прямо до одержувача, минаючи стадію створення паперової копії на кожному етапі.

Термін "електронна комерція" поєднує в собі множина різних технологій, у числі яких – iEDI, електронна пошта, Інтернет, інтранет (обмін інформацією усередині компанії), екстранет (обмін інформацією із зовнішнім світом). Таким чином, електронну комерцію можна характеризувати як ведення бізнесу через Інтернет.

В електронній комерції створюються умови для реалізації моделі зробленої конкуренції, тому що формується достатність інформації, необмежена кількість покупців і продавців, зводяться до нуля операційні витрати й практично ліквідуються всі бар'єри для нових учасників ринку. У якості гравців на ринку електронної комерції можуть виступати споживачі ( фізичні особи), бізнес (юридичні особи) і адміністрація (державні організації).

Залежно від учасників, в електронній комерції сформувалися наступні сектори:

- сектор електронної комерції Business-to-business (B2B), що включає в себе всі рівні взаємодії між компаніями;
- сектор електронної комерції Business-to-consumer (B2C), орієнтований на кінцевого споживача, основу цього напрямку становить електронна роздрібна торгівля;

- сектор електронної комерції Business-to-administration (B2A), орієнтований на угоди між компаніями й державними або урядовими організаціями;

- сектор електронної комерції Consumer-to-Administration (C2A), орієнтований на взаємодію між фізичними особами й адміністрацією;

- сектор електронної комерції Consumer-to-Consumer (C2C), орієнтований на взаємодію між більшою кількістю кінцевих споживачів.

Треба відзначити також, що багато компаній прагнуть використовувати у своїй діяльності відразу кілька моделей. На сьогоднішній день двома найбільшими секторами електронного ринку є сектори B2B и B2C.

Основна відмінність Інтернет-магазину від традиційного – у типі торговельного майданчика. Якщо звичайному магазину потрібний торговельний зал, вітрини, цінники, а також продавці, касири й досвідчені консультанти, то в його онлайн-ового «тезки» уся інфраструктура реалізована програмно. Інакше кажучи, Інтернет-магазин – це сукупність програм, що працюють на Web-сайті, які дозволяють покупцеві дистанційно вибрати товар з каталогу й оформити його замовлення. Функції вітрини й торговельного залу виконують «сторінки» з ілюстрованим каталогом товарів. Усе інше – як у звичайному магазині. Навіть в інтерфейсі Інтернет-Магазину зберігаються звичні елементи, наприклад, віртуальна «візок» («кошик»), куди можна по шляху до каси скласти обрані товари.

Усі найбільші компанії США вже перевели свою комерцію на електронний ринок. Підприємство електронної комерції, як і будь-яке традиційне підприємство в цей час розглядається як система взаємозалежних бізнес-процесів, спрямованих на досягнення певних цілей. Інформаційні технології є «локомотивом» змін, який приводить у рух усі частини компанії. Вони використовуються для забезпечення складних бізнес-процесів, що мають місце як усередині компанії, так і між декількома компаніями, що й дозволяють їм спільно брати участь у прийнятті рішень, вести документообіг, ділитися інформацією і т.д.

Не відстають від найбільших компаній і торговельні мережі, і магазини із продажу молодіжного одягу різних напрямків. Застосовуючи останні розробки в області інформаційних технологій для збільшення кількості продажів і просування інтернет-магазинів у мережі.

Такі магазини як:

- бостонський магазин із продажу одягу <http://www.karmaloop.com/> – магазин різного напрямку від звичайного, повсякденного одягу до дуже специфічних марок. Мекка вуличного одягу східного узбережжя США.

- каліфорнійський магазин із продажу хіп-хоп одягу й одягу преміум класу <http://www.drjays.com/> – має свій чіткий стиль і спрямованість.

- одна з найбільших торговельних мереж роздрібних продажів <http://macys.com/> – представляють широкий вибір брендів і стилів.

Таким чином, можна сформулювати основні принципи організації й функціонування систем електронної комерції:

- вибір вірної технологічної платформи. Передбачає застосування останніх розробок у сфері інтернет-торгівлі, для максимального підвищення ефективності функціонування інтернет-магазину;

- аналіз ринку й предметної області для того, щоб надати конкурентоспроможний товару;

- наявність необхідної інфраструктури й «простих» бізнес-процесів, тобто процеси реєстрації, пошуку, вибору й замовлення товару мають бути максимально ефективні й інтуїтивно зрозумілі користувачеві.

## 1.2 Класифікація інтернет-магазинів

Класифікувати інтернет-магазини можна за різними критеріями (рис. 1.1).

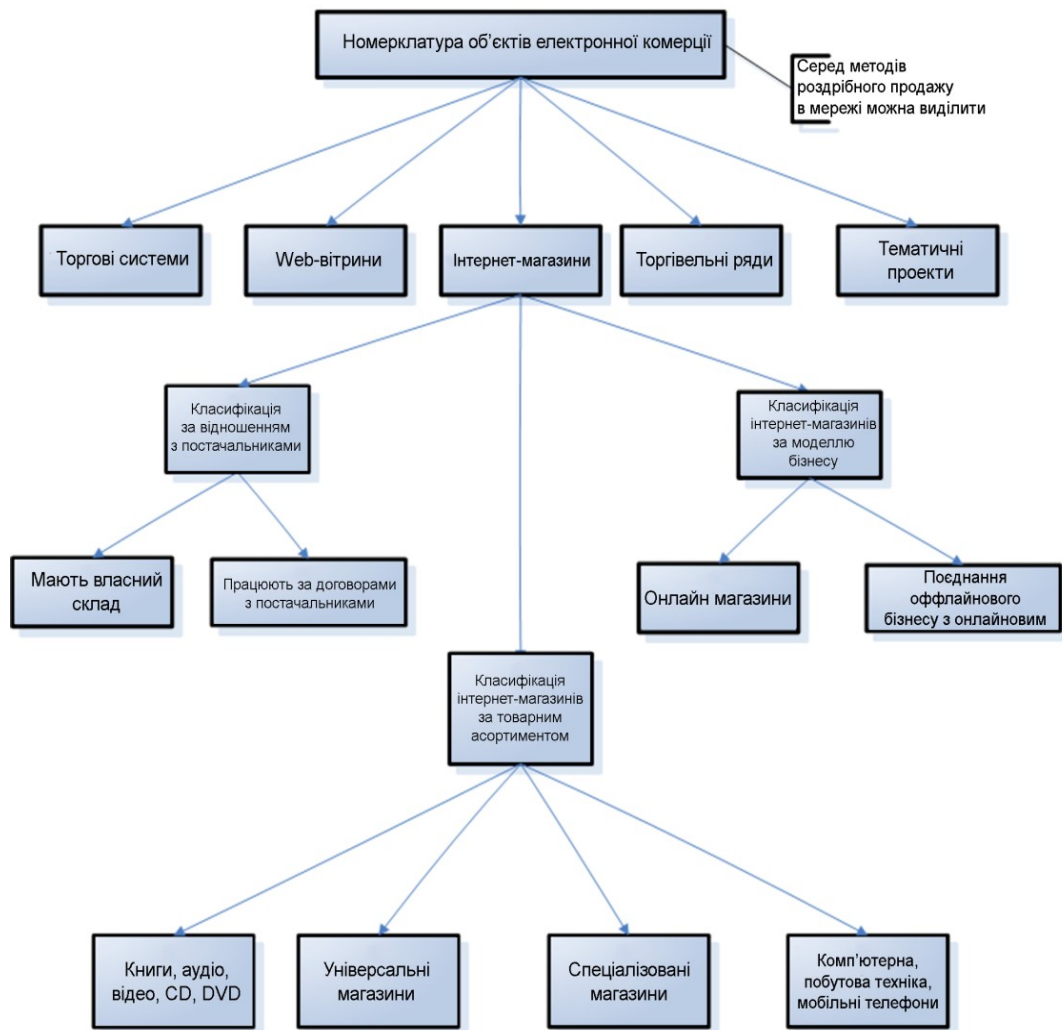


Рисунок 1.1 – Класифікація інтернет-магазинів

Класифікацію Інтернет-магазинів по товарному асортименту можна представити у вигляді сегментів у процентнім співвідношенні (рис.1.2).

На рис. 1.2 видно, що в кожного типу товару є свій сегмент ринку й спеціалізовані магазини – це окремий і найбільший клас інтернет-магазинів, який займає 41% ринку.

Існує велика кількість підкласів спеціалізованих інтернет-магазинів, наприклад:

- військовий одяг (спеціальні куртки, черевики, штани);
- антикварні речі (меблі й предмети розкоші);
- усі що пов'язане з риболовлею (вудки, гачки і т.д.);
- одяг молодіжного напрямку (lifestyle clothing, hip-hop clothing);
- весільні плаття;

- усе для автомобіля;
- спортивне спорядження.



Рисунок 1.2 –Сегменти ринку електронної торгівлі

У кожного підкласу є своя предметна область зі своєю чітко вираженою специфікою й клієнтурою. Для організації торгівлі товаром того або іншого підкласу, необхідно мати широкую виставу про ринок збуту й урахувувати специфіку продаваного товару.

Таким чином, інтернет-магазини із продажу молодіжного одягу – це окремий підклас спеціалізованих інтернет-магазинів, який займає свою нішу на основі ексклюзивності й унікальності пропонованого товару.

### 1.3 Особливості організації електронної комерції для спеціальних груп користувачів

Що стрімко розвиваються субкультури в Україні являють собою величезний ринок збуту товарів молодіжного напрямку. Компанії одягу випускають усе більше й більше моделей одягу і взуття для представників різних субкультур. Ріст популярності молодіжних плинів веде до швидкого розвитку магазинів, що пропонують товар того або іншого напрямку.

Відкриття інтернет–магазинів для такої аудиторії людей є дуже актуальним явищем. Оскільки магазини такої спрямованості знаходяться у великих містах, а побудова торговельної в мережі по всій Україні пов'язане з величезними фінансовими витратами й супутніми проблемами. Те відкриття інтернет-магазину є рішенням даної проблеми, тому що з'являється можливість роботи з усією Україною, що значно збільшує кількість клієнтів і скорочуються грошові витрати на організацію торгівлі.

Є низка факторів, які варто враховувати при організації торгівлі через інтернет для молоді:

- специфіка молодіжного плину;
- відмінності тим часом або іншим молодіжним плином;
- зміна тенденцій вуличної моди;
- психологічний фактор молоді.

Беручи до уваги ці фактори в роботі інтернет-магазину можна значно підвищити продажі й викликати великий інтерес клієнтів до магазину.

На рис. 1.3 показані різні молодіжні тренди, яким брендам віддають перевагу молоді люди й взаємозв'язок між плинами. Вулична мода дуже різностороння й багатогранна. У магазинах, що продають одяг для молоді, працюють досвідчені продавці-консультанти, які знають культуру, живуть нею й постійно стежать за новими тенденціями. Вони в будь-який момент можуть проконсультувати клієнта. Створення віртуального інтернет-консультанта, який буде мати ці знання, суттєво підвищить кількість продажів інтернет-магазину й покаже, що сайт є ексклюзивним.

Тому застосування шаблонів, стандартних форм організації торгівлі для молодіжних субкультур повною мірою не може відобразити всі тонкощі й специфіку продаваного товару.



## 1.4 Аналіз засобів організації електронної комерції для обслуговування спеціальних груп користувачів

Як правило, в інтернет-магазинах для організації процесу торгівлі використовують архітектуру клієнт-сервер. Суть її в тому, щоб розділити функції між двома підсистемами: клієнтом, який відправляє запит на виконання яких-небудь дій, і сервером, який виконує цей запит. Взаємодія між клієнтом і сервером відбувається за допомогою стандартних спеціальних протоколів, таких як TCP/IP.

Сервер являє собою набір програм, які контролюють виконання різних процесів. Відповідно, цей набір програм установлений на якомусь комп'ютері. Основна функція комп'ютера-сервера – по запиті клієнта запустити який-небудь певний процес і відправити клієнтові результати його роботи.

Клієнтом називають будь-який процес, який користується послугами сервера. Клієнтом може бути як користувач, так і програма. Основне завдання клієнта – виконання додатка й здійснення зв'язки із сервером, коли цього вимагає додаток. Тобто клієнт повинен надавати користувачеві інтерфейс для роботи з додатком, реалізовувати логікові його роботи й при необхідності відправляти завдання серверу.

Взаємодія між клієнтом і сервером починається з ініціативи клієнта. Клієнт запитує вид обслуговування, установлює сеанс, одержує потрібні йому результати й повідомляє про закінчення роботи.

Мінуси застосування клієнт-серверного додатка:

- стандартна форма організації торгівлі, яка не враховує специфіку продаваного товару;
- клієнт витрачає досить багато часу на пошук потрібного товару.

Застосування систем штучного інтелекту, дозволить зменшити затрачений час на пошук потрібного товару, процес продажу стане більш

ефективним, а також розширить можливості обслуговування клієнтів магазину. Одні з таких систем це експертні системи.

Експертні системи –це напрямок досліджень в області штучного інтелекту по створенню обчислювальних систем, що вміють ухвалювати рішення, схожі з рішеннями експертів у заданій предметній області.

Як правило, експертні системи створюються для рішення практичних завдань у деяких вузько-спеціалізованих областях, де більшу роль відіграють знання «колишніх» фахівців. Експертні системи були першими розробками, які змогли привернути велику увагу до результатів досліджень в області штучного інтелекту.

Експертні системи мають одна велика відмінність від інших систем штучного інтелекту: вони не призначені для рішення якихось універсальних завдань, як наприклад нейронні мережі або генетичні алгоритми. Експертні системи призначені для якісного рішення завдань у певній розроблювачами області, у рідких випадках – областях.

Експертне знання – це комбінація теоретичного розуміння проблеми й практичних навичок її рішення, ефективність яких доведена в результаті практичної діяльності експертів у даній області. Фундаментом експертної системи будь-якого типу є база знань, яка складається на основі експертних знань фахівців. Правильно обраний експерт і вдала формалізація його знань дозволяє наділити експертну систему унікальними й коштовними знаннями. Цінність усієї експертної системи як закінченого продукту на 90% визначається якістю створеної бази знань.

Експертна система є плодом спільної роботи експертів у даній предметній області, інженерів по знаннях і програмістів.

Jess – це оболонка для розробки експертних систем, написана Ернестом Фридманом-Хиллом (Sandia National Laboratories in Livermore, CA) повністю мовою Java. Jess споконвічно був похідною мови CLIPS, але незабаром виріс у повне, окреме, динамічне середовище. Використовуючи Jess, можна побудувати Java додаток з можливістю обробки даних на основі знань, представлених у

вигляді правил. На даний момент, Jess – і один з найбільш легких і швидких оболонок для експертних систем.

Скриптова мова Jess усе ще сполучина з CLIPS, тобто більшість скриптів Jess будуть працювати в CLIPS і навпаки. Як і CLIPS, ядро Jess використовує Rete алгоритм для обробки фактів і правил, який є дуже ефективним і швидкодіючим при рішенні завдань множинного порівняння (many-to-many matching problem). Jess також містить у собі можливість створення, керування й виклику методів Java об'єктів.

Прийняття рішень в Jess. Розуміння даного алгоритму може бути корисним для планування ЕС. Типові ЕС мають встановлену множину правил, у той час як база знань (фактів) може мінятися. Очевидним рішенням є створення списку правил і постійний його моніторинг на предмет збігу умов правила з існуючими в системі фактами. Але дане рішення неефективне, тому що в більшості випадків результат циклу буде повторюватися досить часто. Замість цього використовується більш ефективний Rete алгоритм. Він запам'ятовує результат останнього тестування знань і заново перевіряє факти, що тільки знову з'явилися.

Даний алгоритм реалізований за допомогою побудови системи вузлів, кожний з яких представляє одне або кілька перевірок фактів для кожного правила. Факт, який додається або віддаляється з бази знань, обробляється цією системою вузлів. В основі цієї системи лежать конкретні дії. Коли набір фактів доходить доценту системи, запускається дія із цього правила (якому й задовольняє система фактів). Сама по собі система складається із двох типів вузлів: з одним входом і із двома входами. Вузли з одним входом здійснюють перевірку окремих фактів, із двома – паралельні перевірки, що й групують функції.

На сьогоднішній день створена вже велика кількість експертних систем. За допомогою їх вирішується широке коло завдань, але винятково в вузько-спеціалізованих предметних областях. Як правило, ці області добре вивчені й розташовують більш менш чіткими стратегіями прийняття рішень. Зараз розвиток експертних систем трохи призупинився, і цьому є ряд причин, що викладені нижче.

Передача експертним системам «глибоких» знань про предметну область є великою проблемою. Як правило, це є наслідком складності формалізації евристичних знань експертів.

Налагодження й тестування будь-якої комп'ютерної програми є досить трудомісткою справою, але перевіряти експертні системи особливо важко. Це є серйозною проблемою, оскільки експертні системи застосовуються в таких критичних областях, як керування повітряним і залізничним рухом, системами зброї й у ядерній промисловості.

Експертні системи мають ще один недолік: вони нездатні до самонавчання. Для того щоб підтримувати експертні системи в актуальному стані необхідно постійне втручання в базу знань інженерів по знаннях. Експертні системи, позбавлені підтримки з боку розроблювачів, швидко втрачають затребуваність.

Міська мода швидко міняється, постійно з'являються, нові тенденції й міняються стилі одягу, це спричинить постійна зміна експертної системи.

Таким чином, застосування експертних систем в інтернет-магазині є не ефективним.

### 1.5 Постановка завдань дослідження

Проблема, що породила дане дослідження полягає у відсутності індивідуального підходу до клієнта інтернет-магазину, шаблонність організації торгівлі, недостатня кількість інформації про предметну область, як наслідок недостатній ефективність функціонування інтернет-магазину із продажу молодіжного одягу.

В результаті аналізу основних принципів організації й класифікації електронної комерції, також опис різних молодіжних плинів, і особливості організації торгівлі для них. Проведений аналіз придатності існуючих на ринку засобів організації електронної комерції, які й виявлені недоліки, які їм властиві.

Метою атестаційної роботи є дослідження алгоритмів підвищення ефективності бізнес-процесів по організації й функціонуванню інтернет-магазину із продажу молодіжного одягу.

Для досягнення мети необхідно розв'язати наступні завдання:

- виконати аналіз поточного стану проблеми організації й функціонування систем електронної комерції;
- виконати огляд і аналіз застосовності існуючих агентних технологій при розробці засобів електронної комерції;
- розробити моделі й методи реалізації бізнес-процесів в електронних магазинах на основі агентної парадигми.
- перевірити теоретичні результати досліджень за допомогою розробленого програмного прототипу.

## 2 ОПИС ПРОВЕДЕНИХ ТЕОРЕТИЧНИХ ДОСЛІДЖЕНЬ

### 2.1 Аналіз методів агрегатного моделювання

Агрегативні моделі Бусленко [15] відзначають, що проблема програмної реалізації агрегату – одна з основних проблем збільшення ефективності агрегативного моделювання. Дослідження класу алгоритмів функціонування агрегату проводиться з метою з'ясування наступних питань: чи мають алгоритми певну специфіку, яка б дозволила уніфікувати процес їх програмної реалізації й чи дозволяє ця специфіка створити для даного класу алгоритмів певну методика програмування, більш адекватну класу цих алгоритмів, ніж традиційні методи програмної реалізації, що існували до початку даного дослідження. Дана методика програмування повинна переслідувати наступні основні цілі:

- дати в руки розробників моделі концептуальну основу для уніфікованого опису елементів дискретних систем у вигляді кусочно-лінійних агрегатів;

- дозволити таким образом структурувати програму функціонування агрегату, щоб максимально спростити внесення змін; створити передумови для нагромадження програмних фрагментів функціонування агрегатів і стимулювати розроблювачів моделі до використання наявного програмного забезпечення в нових імітаційних моделях;

- створити можливість для автоматизації компонування нових моделей з наявних фрагментів, а також автоматизації введення параметрів у модель (бланки розробників і користувачів моделі).

Для рішення питання програмної реалізації, агрегату необхідно використовувати підходи до створення програмного забезпечення.

Для програмної реалізації алгоритмів моделей необхідно доповнити парадигму програмування наступною: структура програми моделі повинна відбивати логічну структуру процесу, що підлягає моделюванню. Рішення цього

питання забезпечується застосуванням запропонованої [16] схеми агрегату. Відповідно до відомих досліджень було встановлено, що доцільно використовувати наступну дворівневу структуру програми, що описує стрибок кусочно-лінійного агрегату. Верхній рівень – керуючий – одержує від головної програми моделі принципів координати стану й вхідного сигналу, з виходу керуючого блоку знімається номер приватної реакції, що здійснює стрибок стану. Нижній рівень, що виконує, реалізує стрибок по зазначеній приватній реакції. Керуючий блок доцільно описувати програмно у вигляді відповідної таблиці рішень блок, що виконує, – у вигляді лінійного ланцюжка типових дій, виділення яких опирається на фізичний зміст моделюємого процесу. Ті самі дії можуть входити в ланцюжки різних реакцій, що спрощує програмну реалізацію стрибків кусочно-лінійного агрегату.

Інформаційна модель агрегату в пам'яті ЕОМ представляється у вигляді відповідного рядка архіву елементів, куди заносяться: таблиця рішень для керуючого блоку, імена процедур для типових дій, адреси реєстрів, параметрів станів, вхідних до вихідних сигналів. Крім того, рядок містить номери елементів, пов'язаних з даним за схемою сполучення. Рядок архіву елементів. являє собою сукупність параметрів для настроювання програми імітації елемента на конкретний моделюємий агрегат. Принциповим є та обставина, що рядка архіву елементів запропонованої в дисертації структури можуть формуватися автоматично, за інформацією, що завдана розробником або користувачем моделі на «бланках». Пропонована Бусленко стандартна форма «бланку» й алгоритм його перетворення в стандартний рядок придатні для широкого класу елементів агрегатних моделей.

Для забезпечення автоматизації етапу розробки моделі доцільно використовувати запропонований Бусленком підхід, програмно реалізований у вигляді генератора агрегативних моделей. Генератор призначений для виконання наступних функцій: уведення уніфікованих схем функціонування агрегату, синтаксичний контроль, контроль логічної структури уніфікованої схеми (повноти, двозначності, суперечливості), побудови за уніфікованою схемою

еквівалентних таблиць рішень, еквівалентні перетворення уніфікованих схем, інтерпретації уніфікованих схем при наявності відповідних словників інтерпретації. На виході генератора можуть бути отримані змістовна, формальна й програмна інтерпретація введеної схеми. Крім того, генератор формує структуру й склад бланка користувача.

Для процесу проведення імітаційних експериментів пропонується використовувати концепцію, реалізовану у вигляді відповідного комплексу програм. Комплекс здійснює імітаційні експерименти, фіксацію й обробку проміжних результатів, послідовну зміну варіантів розрахунку, остаточну обробку й видачу результатів. Усі ці функції виконуються в рамках стратегії, що задається користувачем, рішення завдання, що забезпечує, крім зазначених вище операцій, перевірку виконання, умов, що відповідають та що диктують перехід до тієї або іншої альтернативної галузі рішень завдання. Комплекс має бути побудований по модульному принципу й містить відповідні бібліотеки варіантів типових програм.

Усі розглянуті вище підсистеми, пов'язані з автоматизацією процесу агрегативного моделювання, доцільно об'єднати в спеціальний пакет програм – автоматизований технологічний комплекс систематичного конструювання агрегативних моделей, їх складу і структури.

Узагальнення результатів практичної реалізації агрегативних моделей доцільно починати з методологічних аспектів створення агрегативних моделей і їх використання для рішення відповідних завдань. Така методологія формалізується у вигляді конвеєрної технології агрегативного моделювання (А-Технології), яка описує основні етапи, типи проведених робіт, робітників, що відповідають за роботи, і місця зі спеціалізацію виконавців. Застосування А-Технології дозволяє перейти від кустарних методів створення моделей до серійного конструювання агрегативних моделей для потреб керування.

Впровадження А-Технології дає ефект як за рахунок економії засобів і зв'язків та ліквідацією дублювання робіт, так і з ростом загальної якості моделей у зв'язку з уніфікацією.

Узагальнення методики практичної реалізації агрегативних моделей для рішення завдань керування доцільно проводити у два етапи. На першому формалізуються основні поняття, пов'язані з керуванням системами: об'єкт керування, система керування, мети керування й т.ін.

На другому з використанням введених понять визначається місце агрегативного моделювання в процесі керування і, таким чином, узагальнюються результати практичної реалізації агрегативних моделей в інформаційних системах.

Формалізацію процесів керування доцільно проводити на основі використання математичного апарата теорії агрегативних систем, що є концептуальною основою для рішення завдань моделювання інформаційних інтелектуальних процесів. Для формалізації об'єкта керування й системи керування необхідно використовувати поняття динамічної системи з дискретним втручанням випадку, доопределив його множиною можливих збурювань нормального процесу функціонування об'єкта.

Відповідно до введеної до реалізації, функціонування системи керування відбувається в такий спосіб. У момент чергового стрибка стану об'єкта керування перевіряється умова близькості стану об'єкта до загальної цільової границі системи керування. Випадок близькості до границі інтерпретується як момент часу, що вимагає керуючого впливу. У цей момент формується відповідна керуюча зміна стану об'єкта. У випадку недостатності керування (вихід за кордон керованості) формується й. видається сигнал у підсистему вищого рівня про необхідність коректування ресурсів керування /чи цільової границі.

Визначення моменту необхідності керуючого впливу є одним із принципових актів у методології імітаційного керування. Від структури цього оператора залежить, чи буде система керування системою, що попереджає, або запізненого типу. Розвиток методології керування, що попереджає, є одним з перспективних додатків методу імітації до керування складними об'єктами. Для визначення моменту необхідного керування слід ввести захід близькості траєкторії об'єкта до цільової границі. Існує два основні підходи до завдання

заходу : на підставі фактичного відхилення й на підставі прогнозу поведінки траєкторії системи. Основою другого варіанта, що відноситься до керування, що попереджає, є імітаційна модель керованого об'єкта.

Прогнозуюча здатність імітаційної моделі керування, що попереджає, базується на додатковій інформації, яка виникає трьох джерел: відомостей про поведінку об'єкта керування, вхідних модель при її створенні, параметрів і станів ситуації, що настала, у момент початку прогнозування, відомостей про передбачувані збурювання нормального процесу функціонування об'єкта, які є в розпорядженні користувача моделі.

Доцільно розглянути й завдання імітаційного планування, тобто планування діяльності об'єкта керування на імітаційній моделі. Завдання імітаційного планування – це не тільки вибрати певний план роботи об'єкта шляхом оцінки варіантів наявних планів, але й забезпечити ефективне виконання цього плану шляхом прогнозування безлічі несприятливих ситуацій, «вузьких місць», які можуть виникнути в процесі його виконання. При цьому з'являється можливість заздалегідь визначити тенденції, що призводять до критичних ситуацій, і заздалегідь підготувати відповідні керуючі впливи (підключення керуючих ресурсів). Усе це формує певну комфортність керування, роблячи його більш обміркованим і діючим.

За результатами імітаційного планування може бути підготовлено кілька варіантів планів з відповідними оцінками стійкості у відношенні до можливих збурювань, що дає підстави особі, що ухвалює рішення, вибрати один з варіантів, що більш відповідає можливостям компенсації несприятливих впливів.

Підхід до використання імітаційних моделей у керуванні (імітаційне прогнозування, імітаційне керування, імітаційне планування) припускає наявність комплексу імітаційних моделей об'єкта керування. Однак, настільки широке впровадження методу імітації в практику керування було б неможливо в рамках традиційних, кустарних методів створення імітаційних моделей. Ця обставина продиктувала необхідність розробки уніфікованого імітаційного підходу до

створення агрегативних моделей, технології, що відповідає, і програмного забезпечення.

У якості перспектив використання агрегативного моделювання в є комплексний підхід до застосування методу імітації при створенні й функціонуванні складного об'єкта, який полягає в розгляді взаємини об'єкта і його моделі на всіх стадіях існування об'єкта, починаючи від виникнення задуму через його проектування, створення й експлуатацію до реорганізації або знищення об'єкта, як такого, що виконав свої функції або свій строк. Основним критерієм ефективності об'єкта в цьому випадку є сукупний внесок у національний доход країни за весь період його існування.

Основний постулат комплексного підходу полягає в наступному: модель об'єкта, призначена для рішення конкретного завдання, на будь-якій стадії існування об'єкта повинна створюватися шляхом уточнення і коректування моделі, створеної на попередній стадії його існування, і, у свою чергу, бути основою для моделювання об'єкта надалі.

Існуюча до початку даного дослідження методика створення моделей вимагала трудомісткого етапу вивчення об'єкта, створеного іншими, на кожній стадії використання моделі. Природно, що ця обставина не сприяла винній ефективності використання методу моделювання.

Будь-яка стадія проектування об'єкта повинна бути одночасно й стадією створення його моделі, тому що автор здійснюваної розробки є найбільш компетентним фахівцем у всіх тонкощах проектного об'єкта. У цьому випадку передача готового проекту й розробленої системи керування є одночасно й передачею на наступний етап алгоритмів, що містять вичерпну модель. Такий «імітаційний проект» не тільки послужить підставою для подальшого використання моделі при функціонуванні об'єкта, але дозволить оцінити в динаміці особливості функціонування об'єкта в критичних ситуаціях, можливості керування й реорганізації.

Метод Бусленко з автоматизації агрегативного моделювання, спрямований на полегшення сприйняття моделі, внесення змін, уніфікації й стандартизації

основних складових моделі, і розроблено технологія їх виготовлення, накопичення й повторного використання. Тому саме й спрямовані на реалізацію комплексного імітаційного підходу до створення моделі об'єкта на всіх етапах його існування.

## 2.2 Аналіз використання агентних систем

Система підтримки прийняття рішень може надавати допомогу в керуванні деяким об'єктом. Однак якщо деякі параметри об'єкта керування ввійшли в критичну зону, а оператор не виправляє ситуацію (відволікся або не має можливості), то система повинна мати можливість втрутитися в роботу керованого об'єкта.

Отже, завданням, виконуваним подібними системами, повинне бути автоматичне реагування на вхідну інформацію. Це може бути й задоволення заявок, що надходять на склад, обробка запитів клієнтів електронних магазинів і багато чого іншого.

Під агентом розуміється самостійна програмна система можливість, що має, ухвалювати вплив із зовнішнього миру, визначати свою реакцію на цей вплив і здійснювати цю реакцію.

Під зовнішнім миром тут розуміється середовище, що оточує агента. Перераховані й багато інші завдання агенти можуть виконувати без використання методів штучного інтелекту. Однак ряд завдань просто не може бути вирішений без них.

Класифікуються агенти на чотири основні типи: прості, розумні (smart), інтелектуальні (intelligent) і дійсно інтелектуальні (truly intelligent).

Інтелектуальний агент цей той агент, який має рядом знань про себе й навколишньому світі й поведінка якого визначається цими знаннями.

Слід помітити, що поняття агента є деяким парафразом визначення об'єкта. Якщо обрати загальноприйнятту формулу “клас = дані + методи (функції для роботи з ними)”, то агент –це об'єкт, що володіє функціями для приймання інформації із зовнішнього миру, впливу на зовнішній мир і ухвалення рішення про рід впливу. Ухвалення рішення може відбуватися рефлексивно (тобто буде строго задане) і описуватися програмою.

Поняття інтелектуального агента є розвитком методів ООП. Тобто «інтелектуальний агент = дані + методи + знання», причому методи тут –це не тільки функції роботи з даними, але й функції роботи зі знаннями й можливі методи впливу на навколишнє середовище агентом. З жалем можна відзначити, що багато розроблювачів затверджують, що використовують у своїх продуктах технології інтелектуальних агентів, тоді як основна їхня відмітна риса – визнання –у розробках не представлені. Слід помітити, що в якості джерела інформації агент може використовувати інформацію, отриману від іншого агента.

Системи, у яких передбачена взаємодія агентів, називаються багатоагентними.

Властивості агентів:

- автономність – агенти функціонують без прямого втручання людей або кого-небудь іншого й володіють певною здатністю контролювати свої дії й внутрішній стан;
- методи (способи) спілкування – агенти взаємодіють із іншими агентами засобами деякої комунікаційної мови;
- реактивність – агенти здатні сприймати навколишнє середовище (яка може бути фізичним миром, користувачем, взаємодіючим через графічний інтерфейс, колекцією інших агентів, Internet-ом, або, можливо, всіма разом узятим) и адекватно реагувати в певних тимчасових рамках на зміни, які відбуваються;
- активність – агенти не просто реагують на зміни середовища, але й мають цілеспрямовану поведінку й здатністю проявляти ініціативу;

- індивідуальна картина миру – кожний агент має власну модель навколишнього його миру (середовища), яка описує те, як агент бачить мир. Агент будує свою модель миру на основі інформації, яку одержує із зовнішнього середовища;

- комунікабельність і кооперативність – агенти можуть обмінюватися інформацією з навколишнім їхнім середовищем і іншими агентами. Можливість комунікацій означає, що агент повинен одержувати інформацію про його навколишнє середовище, що дає йому можливість будувати власну модель миру. Більше того, можливість комунікацій з іншими агентами є обов'язковою умовою спільних дій для досягнення цілей;

- інтелектуальна поведінка – поведінка агента включає здатність до навчання, логічної дедукції або конструювання моделі навколишнього середовища для того, щоб знаходити оптимальні способи поведінки.

Отже, кожний агент – це процес, який володіє (розташовує) певною частиною знань про об'єкт і можливістю обмінюватися цими знаннями з іншими агентами. Наявність штучного інтелекту означає, що агент повинен деяким чином зберігати свої знання. За історію штучного інтелекту була розроблена множина методів вистави знань. Однак найпоширенішими на даний момент є продукції (правила виду “якщо ..., то ...”) і нейронні мережі. Перші завоювали свій успіх завдяки простоті розуміння, формалізації й реалізації. Другі – тим, що немає необхідності формалізувати знання й затягати їх у базу, а досить навчити мережа. І ті, і інші дають цілком непогані результати.

Важливим аспектом, що говорять на користь інтелектуальних агентів є те, що ми маємо можливість реалізувати повнофункціональну інтелектуальну самостійну систему не прибігаючи (повністю або майже повністю) до апаратної реалізації. Так, наприклад, віртуальний продавець-консультант, реалізований по агентної технології, може реально стежити за діями в системі й пропонувати потрібний клієнтові товар. При цьому вся його дії не вийдуть за рамки програмних впливів.

Багатоагентна система – складна система, у якій функціонують два або більш інтелектуальних агентів. Напрямок «багатоагентної системи» розподіленого штучного інтелекту розглядає рішення одного завдання декількома інтелектуальними підсистемами. При цьому завдання розбивається на декілька підзадач, які розподіляються між агентами. Ще однією областю застосування МАС є забезпечення взаємодії між агентами, коли один агент може виробити запит до іншого агента на передачу деяких даних або виконання певних дій. Також у МАС є можливість передавати знання. Побудова програмних систем за принципом МАС може бути обумовлене наступними факторами:

- деякі предметні області застосовують МАС у тих випадках, коли логічно буде кожного з учасників процесу представити у вигляді агента. Наприклад, соціальні процеси, у яких кожний з учасників відіграє свою роль;
- паралельним виконанням завдань, тобто якщо предметна область легко представляється у вигляді сукупності агентів, те незалежні завдання можуть виконуватися різними агентами;
- стійкістю роботи системи: коли контроль і відповідальність за виконувані дії розподілені між декількома агентами. При відмові одного агента система не перестає функціонувати. Таким чином, логічно помістити агентів на різних комп'ютерах;
- модульністю МАС, що дозволяє легко нарощувати й видозмінювати систему, тобто легше додати агента, чим змінити властивості єдиної програми. Системи, які змінюють свої параметри згодом можуть бути представлені сукупністю агентів, модульність обумовлює легкість програмування МАС.

Мультиагентні системи підрозділяються на кооперативні, що конкурують і змішані. Агенти в кооперативних системах є частинами єдиної системи й вирішують підзадачі одного загального завдання. Зрозуміло, що при цьому агент не може працювати поза системою й виконувати самостійні завдання. Конкуруючі агенти є самостійними системами, хоча для досягнення певних цілей, вони можуть об'єднувати свої зусилля, ухвалювати мети й команди від інших агентів, але при цьому підтримка зв'язку з іншими агентами не обов'язкова. Під

змішаними агентами розуміються конкуруючі агенти, підсистеми яких також реалізуються за агентною технологією. Крім спілкування з іншими агентами повинна бути реалізована можливість спілкування з користувачем

Отже, агентна технологія, на думку багатьох фахівців в області штучного інтелекту, є перспективною областю дослідження. За словами Джорджа Люггера, якщо вважати поява проміжного програмного забезпечення першим етапом еволюції архітектури інформаційних систем, першоосновою якої була клієнт-серверна модель, то технологія мобільних агентів – її наступний етап.

На рис. 2.1 представлена схема взаємодії сучасних технологій: АОТ, WWW і додатків е-бізнесу. АОТ – це інтегрована технологія, що використовує різні джерела й концепції: теорія рішень; розподілені системи; об'єктно-орієнтована технологія; теорія організацій; системи баз знань.

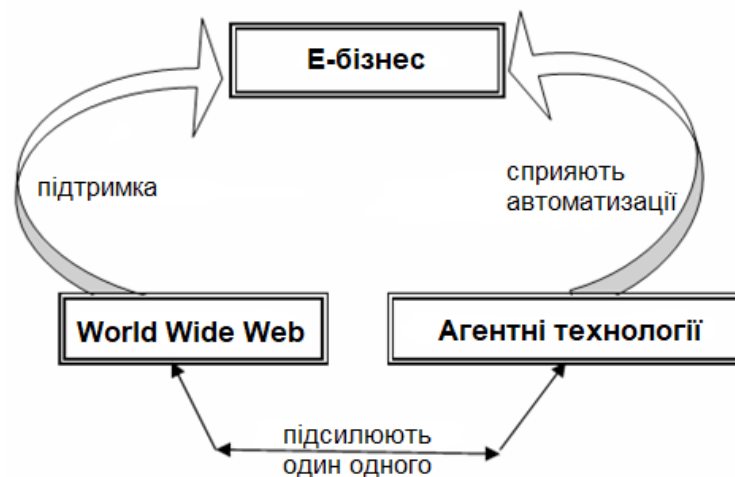


Рисунок 2.1 – Агентно-орієнтовані технології для інтелектуалізації завдань е-бізнесу

Рішення ряду завдань допоможе автоматизувати й інтелектуалізувати обробку інформації й, як наслідок, прискорити й поліпшити цю обробку. Застосування мультиагентної технології покликане до зменшення часу затрачуваного клієнтом на пошук потрібного йому товару, також це дасть можливість організувати неявний підхід до клієнта за рахунок застосування віртуального продавця (агента), який уміло направить клієнта до запитуваного товару.

Були розглянуті наступні технології: клієнт-серверний додаток, експертні системи, агентні технології. І було доведено, що застосування агентних технологій допоможе розв'язати поставлене завдання по підвищенню ефективності інтернет-магазину.

### 2.3 Аналіз концептуальної моделі Інтернет-магазинів

Для вирішення завдання вибору й обґрунтування моделей бізнес-процесів в інтернет-магазині спочатку необхідно формально представити бізнес-процеси, а потім здійснити обґрунтований вибір моделей для опису роботи. Для моделювання бізнес-процесів використовуються діаграми уніфікованого мови моделювання UML. Їхнє застосування дозволяє зробити процес моделювання більш гладким і гарантувати, що створену веб-систему буде просто підтримувати надалі.

На стадії планування необхідно розібратися в наступних аспектах системи:

- хто буде користувачами системи, і які будуть ролі цих користувачів;
- що повинен робити додаток;
- взаємне розташування сторінок і порядок переміщення між ними.

На рис. 2.2 представлена ієрархія виконавець-роль, що відображає користувачів системи.

На представленому вище рисунку зображено кілька груп користувачів (називані мовою UML «виконавцями» (actors)), які будуть працювати з вебсайтом. У цьому випадку самий загальний тип користувача ("Користувач сайту") розташований угорі діаграми. Є дві більш специфічні категорії користувачів: відвідувачі-гості й зареєстрований користувач (адміністратор).

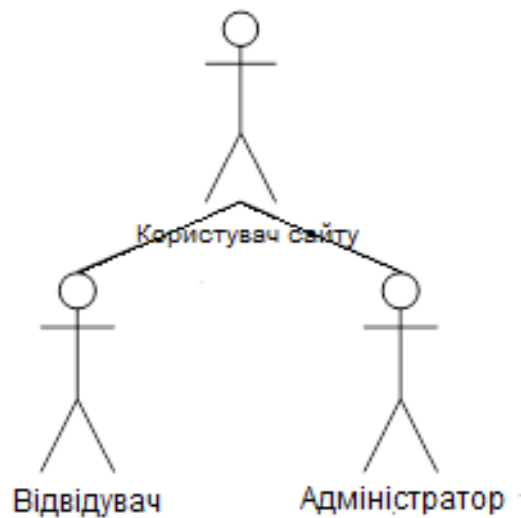


Рисунок 2.2 – Ієрархія користувачів системи

Гості будуть мати можливість переглядати вміст сайту, замовляти товари, відправляти листа з питаннями консультантам, оцінювати товари. Адміністратор сайту буде виконувати наповненням БД товарами, переглядати замовлення покупців, здійснювати деякі налаштування в роботі сайту.

Візуальне моделювання в UML можна представити, як деякий процес по-уровневого спуска від найбільше обшій і абстрактної концептуальної моделі вихідної системи до логічної, а потім і до фізичної моделі відповідної програмної системи. Для досягнення цих цілей спочатку будується модель у формі так званої діаграми варіантів використання (use case diagram), яка описує функціональне призначення системи або, інакше кажучи, те, що система буде робити в процесі свого функціонування. Діаграма варіантів використання є вихідним концептуальним представленням або концептуальною моделлю системи в процесі її проектування й розробки.

Розробка діаграми варіантів використання переслідує мети:

- визначити загальні границі й контекст моделіруємої предметної області на початкових етапах проектування системи.
- сформулювати загальні вимоги до функціональної поведінки проектованої системи;
- розробити вихідну концептуальну модель системи для її наступної деталізації у формі логічних і фізичних моделей;

- підготувати вихідну документацію для взаємодії розроблювачів системи з її замовниками й користувачами [13].

На рис. 2.3 представлена діаграма варіантів використання для відвідувача сайту.



Рисунок 2.3 – Діаграма варіантів використання для відвідувача сайту

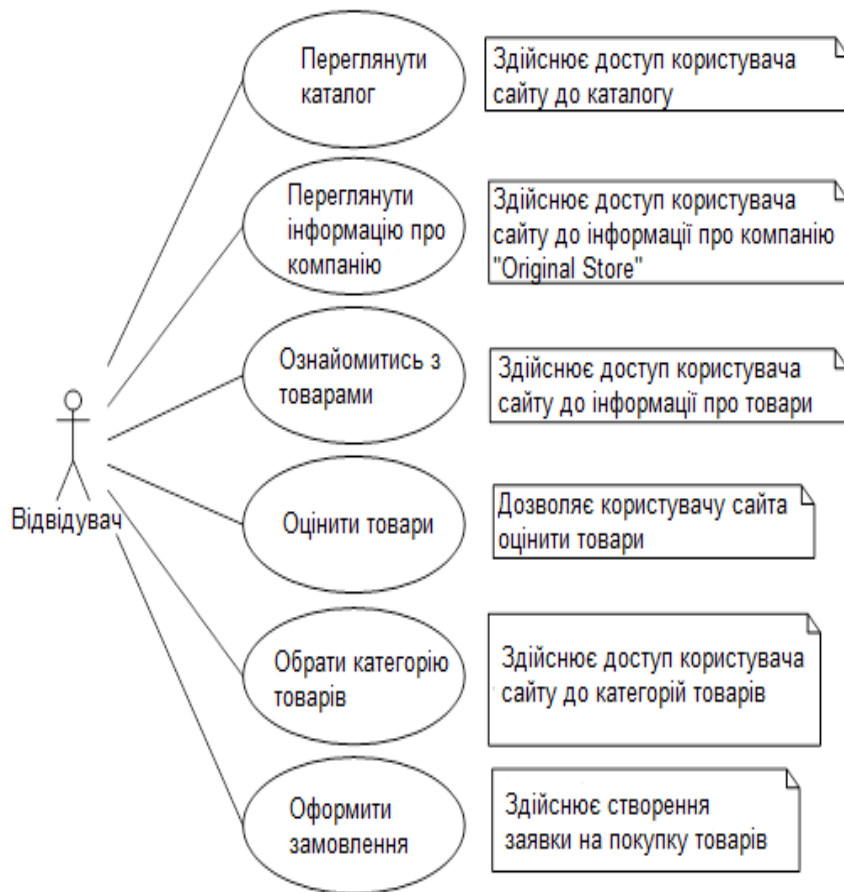


Рисунок 2.4 – Діаграма варіантів використання для відвідувача сайту

На рис. 2.5 представлена діаграма варіантів використання для адміністратора сайту.

Для планування сторінок сайту і їх взаємодії були обрана діаграми класів. Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. Діаграма класів може відбивати, зокрема, різні взаємозв'язки між окремими сутностями предметної області, такими як об'єкти й підсистеми, а також описує їхню внутрішню структуру й типи відносин. На даній діаграмі не вказується інформація про тимчасові аспекти функціонування системи. Із цього погляду діаграма класів є подальшим розвитком концептуальної моделі проектованої системи.

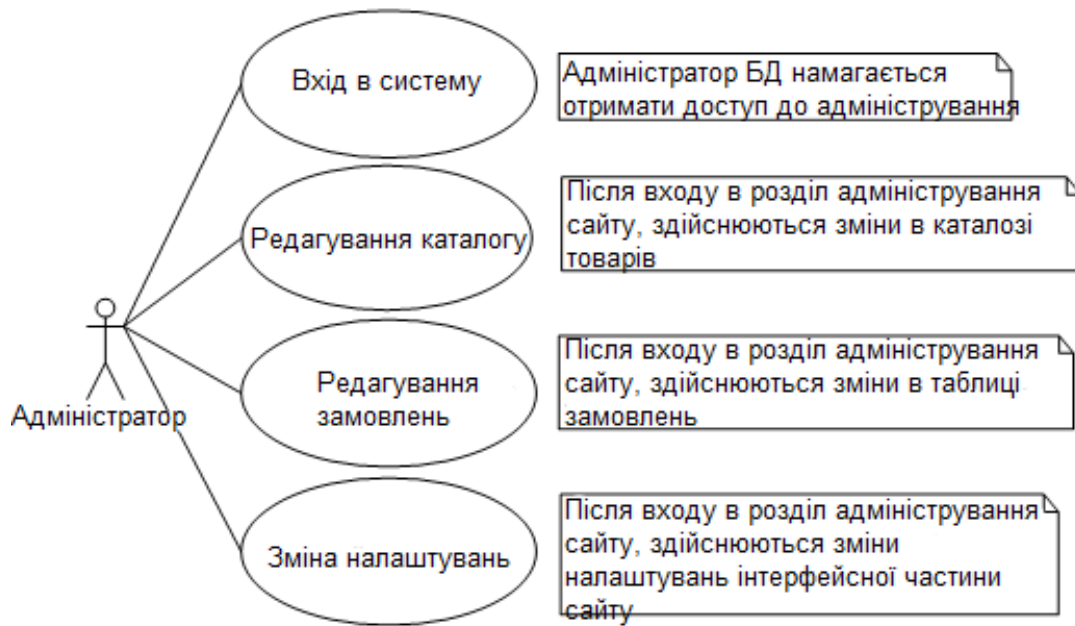


Рисунок 2.5 – Діаграма варіантів використання для адміністратора сайту

## 2.4 Моделювання агрегатних моделей бізнес-процесів

При виборі тієї або іншої схеми формалізації процесу функціонування складної системи основну роль відіграють два міркування

- забезпечити необхідну точність рішення завдання;
- одержати як можна більш просту модель.

Досить широкий клас складних систем, що мають важливе практичне значення, формалізується у вигляді систем масового обслуговування. Однак схема масового обслуговування не є настільки загальною, щоб охопити будь-які елементи сучасних складних систем. Бажання дати єдиний математичний опис усім елементам складної системи й тим самим добитися рішення ряду важливих теоретичних і практичних питань приводить до виникнення усе більш і більш загальних абстрактних схем, призначених для формалізації реальних об'єктів.

Агрегат – абстрактна схема функціонування складної системи. Принципово загальна схема в стані замінити всі частки. Однак вона складніше й для окремих випадків її застосування приводить до значного ускладнення обчислень. Проте

завдання, що існують, для яких приватні схеми не працюють або працюють погано. У цих випадках доводиться користуватися більш загальною схемою, поки не розроблена більш проста схема – приватна. Головним стимулом застосування загальних схем виявляється універсалізація математичного опису й операторів моделювання алгоритму для всіх елементів складної системи.

У кожний момент часу  $t \in (0, T)$  агрегат перебуватиме в одному з можливих станів. Стан агрегату характеризується набором символів, що  $z_1, z_2 \dots z_i^*$  є елементом деякої множини  $Z$ . Якщо стан  $z = (z_1, z_2 \dots z_i^*)$  виявляється дійсними або комплексними векторами.  $z_i$  – звичайно називають фазовими координатами.

Коли аргумент  $t$  пробігає свої значення в інтервалі  $(0, T)$  символи  $z_i$  змінюються як функції часу  $z_i(t)$ . Надалі функції  $z(t)$  ми часто будемо називати фазовими траєкторіями. Функції  $z_i(t)$  можуть залежати від ряду параметрів, які ми будемо позначатися  $\beta_m, m = 1, 2 \dots m^*$ . У початковий момент часу  $t_0$  стан  $z_1$  мають значення рівні  $z_1^0$  відповідно до цього початковий стан агрегату має закон розподілу  $L[z_1(t_0)]$ , що виходить із  $L[z_1(t)]$  при  $t = t_0$ .

Стан агрегату  $z(t)$  для довільного моменту часу  $t > t_0$  визначається по попередніх станах випадковим оператором  $H$ :

$$z(t) = H[z(t_0), t] \quad (2.1)$$

Це означає, що даному  $z(t_0)$  ставитися у відповідність у загальному випадку не одне певне  $z(t)$ , а множина значень  $z(t)$  із деяким законом розподілу, що залежать від виду оператора  $H$ .

У загальному випадку послідовності виду  $(t_j, x_j)$  реалізують випадкову послідовність  $(\theta_j, x_j)$  із законом розподілу  $L(\theta, x)$

Агрегат має особливий вхідний полюс, до якого надходять у момент часу  $t_i$  керуючі сигнали. Керуючий сигнал  $g$ , є елементом множини  $G$ .

У загальному випадку послідовності виду  $(t_i, g_i)$ , виявляються реалізаціями випадкових послідовностей  $(\theta_i, y_i)$  із законом розподілу  $L(\theta, y)$ .

На виході агрегату утворюються вихідні сигнали. Вихідний сигнал  $y$  є елементом деякої множини  $Y$ , і визначаються по станах агрегату  $z(t)$  за допомогою оператора  $G$  (рис. 2.6).

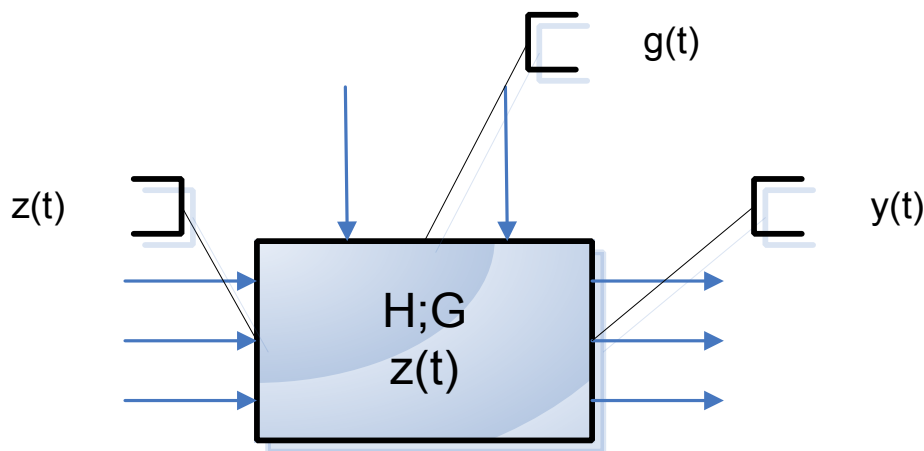


Рисунок 2.6 – Агрегат

Аналогічно слід припустити, що за кінцевий інтервал часу агрегат видає лише кінцеве число вихідних сигналів.

Оператор  $H$  зазвичай називають оператором переходів у новий стан, а оператор  $G$  – оператором переходів.

Стан агрегату  $z(t)$  для довільного моменту часу  $t$ , визначається оператором  $H$ . Вид оператора  $H$  залежить від того, чи містить розглянутий інтервал часу моменти так званих особливих станів агрегату або не містить.

З особливих станів агрегат може переходити в новий стан стрибком.

Нехай  $z(t^*)$  – може переходити в новий стан агрегату, а  $g_i$  – наступний керуючий сигнал  $g \in \Gamma$ .

Прийняте наступне позначення для операторів, що є приватними видами оператора  $H$  й визначальних стан агрегату в момент  $t_0 + 0$ .

Якщо  $t^*$  – момент вступу в агрегат вхідного сигналу  $X$ , то:

$$z(t^* + 0) = V'[z(t^*), x, g_1] \quad (2.2)$$

Аналогічно, якщо  $t^*$  – момент вступу в агрегат керуючого сигналу  $g_1$ , то:

$$z(t^* + 0) = V[z(t^*), g] \quad (2.3)$$

Нарешті, якщо  $t^*$  – момент видачі вихідного сигналу  $Y$ , тоді

$$z(t^* + 0) = W[z(t^*), g] \quad (2.4)$$

В інтервалах між особливими станами значення  $z(t)$  визначається за допомогою операторів  $U_{i^*}$ , від яких у загальному випадку залежить від особливого стану, що є для даного інтервалу часу початковим станом.

$$z(t) = U_{i^*}[z(t^* + 0, g_1, t)] \quad (2.5)$$

де  $t^*$  – момент особливого стану, що є вихідним для даного інтервалу часу.

Природно зауваження про те, що  $H$  є випадковим оператором, без зміни переноситься, на його приватні види  $U, V', V''$  и  $W$

Опис роботи оператора  $G$ . У множині  $Z$  станів  $z(t)$  агрегату видається система підмножин  $\{Z_y\}$ , що має наступні властивості.

Вихідний сигнал  $Y$  видається в момент  $t'$  у тих випадках коли:

- стан  $z(t')$  належить підмножині  $Z_y$ ;
- стан  $z(t' + 0)$  належить підмножині  $Z_y$ , але  $z(t')$  не належить

підмножині  $Z_y$ .

Таким чином, оператор  $G$  можна представити у вигляді сукупності двох операторів  $G'$  вихідний сигнал, що виробляє.

$$y = G'[z(t', g)] \quad (2.6)$$

Оператор  $G''$ , є таким, що перевіряє приналежність  $z(t)$  до однієї з підмножин  $Z_y$ . для кожного  $t$  У загальному випадку оператори  $G'$  і  $G''$  є випадковими операторами.

Агрегат функціонує в такий спосіб. У початковий момент часу  $t_0$  задані початкові стани агрегату  $z_0$  й початкове значення керуючого сигналу  $g_0$ .

Нехай  $t_1$  і  $t_2$  – моменти вступу першого  $x_1$  й другого  $x_2$  вхідних сигналів;  $\tau_1$  – момент вступу першого керуючого сигналу  $g_1$  у  $u_1$  для визначеності  $t_1 < \tau_1 < t_2$ . Розглянемо напівінтервал  $(t_0, t_1)$  стан агрегату змінюється із часом за законом

$$z(t) = U_{t_0}[z_0, g_0, t] (t_0 < t \leq t_1) \quad (2.7)$$

Доти оператор  $G''$ , поки в момент  $t'$  ( $t' < t_1$ ) стан  $z(t')$  не стане приналежним підмножині  $Z_y$ , хоча стан  $z(t' - \epsilon)$  не належав  $Z_y$  при досить малих  $\epsilon > 0$ .

У цьому випадку в момент  $t'$  видається вихідний сигнал, вироблений оператором  $G'$ . Разом з тим закон зміни стану агрегату порушується й

$$z(t' + 0) = W[z(t', g_0)] \quad (2.8)$$

Перш ніж розглядати подальші зміни станів агрегату в часі, необхідно перевірити (оператор  $G''$ ), не чи задовольняє стану  $z(t' + 0)$ . Умовам видачі вихідного сигналу, або інакше кажучи, не чи належить (у змісті умов згаданих вище) стану  $z(t' + 0)$  деякій підмножині  $Z_y$  відмінному від  $Z_y$ .

Якщо стану  $z(t' + 0)$  задовольняє умовам видачі вихідного сигналу (належить підмножині  $Z_y$ ), то в момент  $t'$  видається другий вихідний сигнал  $Z_y$  (оператор  $G'$ ), а стан агрегату описується співвідношенням:

$$z(t' + 0 + 0) = W[z(t' + 0), y_0] = W(W[z(t', y_0)]), y'_0 \quad (2.9)$$

У силу прийнятої вище угоди в момент  $t'$  може бути видане лише кінцеве число вихідних сигналів. Ця властивість агрегату є обмеженням, що накладаються на структуру підмножин  $Z_y$  і операторів  $W$ .

Розглянуто випадок, що  $z(t' + 0)$  не належить ніякій з підмножин  $Z_y$ . Тому стан агрегату змінюється відповідно до закону.

$$\begin{aligned} z(t) &= U_{t'}[z(t' + 0), g_0, t] = U_{t'}^* \{[W[z(t')], g_0], g_0, t\} \\ t' &< t \leq t_1 \end{aligned} \quad (2.10)$$

Аналогічно вирішується питання про видачу наступних вихідних сигналів і змін станів із часом. Агрегат являє собою математичну схему досить загального виду, окремими випадками якої є функції алгебри логіки, релейно-контактні схеми, кінцеві автомати, усілякі класи систем масового обслуговування, динамічні системи, описувані звичайними диференціальними рівняннями, і деякі інші об'єкти.

З погляду моделювання агрегат виступає як універсальний переробник інформації, – він сприймає вхідні й керуючі сигнали й видає вихідні сигнали.

За кінцевий інтервал часу в агрегаті надходить кінцеве число вхідних і керуючих сигналів і видається кінцеве число вихідних сигналів. У такому випадку вхідні сигнали можна пронумерувати в порядку їх набуття. Сукупність вхідних сигналів, що розташовані у порядку їх надходження є вхідним повідомленням,

або (x)-повідомленням. Вхідне повідомлення, що містить із сигналів, що надходять в агрегат протягом напівінтервалу часу  $(t_1, t_2]$  позначено  $(x]_{t_1}^{t_2}$ .

Сукупність керуючих сигналів, упорядкована щодо часу вступу в агрегат, називається керуючим повідомленням або (g)-повідомленням.

Керуюче повідомлення, що є сукупністю сигналів, що надходять в агрегат протягом напівінтервалу часу  $(t_1, t_2]$ , позначається  $(g]_{t_1}^{t_2}$ .

Для будь-якого напівінтервалу часу  $(t_1, t_2]$  можна побудувати сукупність вхідних і керуючих сигналів, упорядковану щодо моментів їх вступу в агрегат.

## 2.5 Вибір концептуальної моделі ПЗ

Оскільки передбачається, що програмний продукт повинен буде обробляти й зберігати великий обсяг даних, те доцільно зберігати цю інформацію певним чином у базах даних. Завдання бази даних полягає в зберіганні всіх даних, що представляють для деякої організації інтерес, в одному місці, до того ж таким чином, який свідомо виключає їхню надмірність. У добре спроектованій базі даних виключається надмірність даних, імовірність збереження суперечливих даних зводиться до мінімуму. Згідно з організацією зберігання даних розглянемо наступні моделі даних: мережні, ієрархічні, реляційні, об'єктно-орієнтовані.

З перерахованих вище моделей даних виберемо модель для представлення даних предметної області за наступними критеріями:

- легкість використання;
- простота реалізації.

На сьогоднішній день найбільш легкі у використанні є реляційні моделі даних, оскільки дана технологія найпоширеніша й існує розвинутий апарат

маніпулювання даними – реляційна алгебра (пошук, запам'ятовування, вибірка й ін.). Мережна модель вимагає розуміння не тільки типів записів і зв'язків, але і їх взаємин. Реалізація зв'язків «багато до багатьох» і зв'язків над трьома й більш наборами об'єктів здійснюється не просто. Подібним чином ієрархічна модель вимагає розуміння використання покажчиків (типів віртуальних записів) і породжує ті ж проблеми, що й мережна, – щодо вистави зв'язків більш складних, чому зв'язки «багато до одного» між двома наборами об'єктів. При розгляді можливостей реалізації високі оцінки одержують реляційна й об'єктно-орієнтована моделі. Через відсутність досвіду побудови об'єктно-орієнтованих моделей даних як моделі даних для нашої предметної області приймемо реляційну модель.

Логічне подання даних відповідає користувацькому уявленню фізичних даних. Відмінність між фізичною й відповідною логічною виставою даних полягає в тому, що останнє відбиває деякі важливі взаємозв'язки між елементами фізичних даних. СКБД повинні забезпечувати незалежність користувача від організації фізичних даних. Зміна користувацького представлення й/або додавання нового представлення підтримуються СКБД і не вимагає витрат на реорганізацію й зміну механізму доступу до файлів фізичних даних.

Модель даних – засіб для визначення логічної вистави фізичних даних, що ставляться до деякого додатка. Модель даних складається із трьох компонентів: структури даних для вистави точки зору користувача на базу даних, припустимих операцій, здійснених на структурі даних і обмежень для контролю цілісності.

Одиницею, що зберігається в БД інформації є таблиця. Кожна таблиця являє собою сукупність рядків і стовпців. Де рядка відповідають екземпляру об'єкта, конкретній події або явищу. А стовпці – атрибутам (ознакам, характеристикам, параметрам) об'єкта, події, явища. У термінах БД стовпці таблиці називаються полями, а її рядки – записами. Для вибору концептуальної моделі даних розглянуто два їх різновиди:

- фрейми;
- модель «сутність-зв'язок».

Фрейми виражаються структурами даних із прив'язаними процедурами обробки цих даних. Фрейми можуть бути наступних видів: собитійні, характеристики, логічні предикати. Використання фреймової моделі також недоцільно, оскільки дана модель не відбиває типи зв'язків у реляційній моделі даних.

Модель «сутність-зв'язок» описує в термінах сутність, зв'язок, значення. Сутність – поняття, яке може бути ідентифіковане. Зв'язок – з'єднання сутностей. Для представлення зв'язків і сутностей введений спеціальний метод: Ер-діаграма. Різняться сутності трьох основних класів: стрижневі, асоціативні й характеристичні. Стрижнева сутність – це незалежна сутність (їй властиво незалежне існування). Асоціативна сутність або асоціація розглядається як зв'язок між двома й більш сутностями типу "багато до багатьом" або подібні ім. Характеристична сутність (або характеристика) являє собою сутність, єдина мета якої, у рамках розглянутої предметної області, полягає в описі або уточненні деякої іншої сутності. Ер-діаграма – графічне представлення взаємозв'язків сутностей.

У зв'язку з тим, що модель «сутність-зв'язок» найбільш близька по принципах організації до реляційної моделі й реалізація останньої на основі першої найбільше зручно, то в якості концептуальної моделі обрана модель «сутність-зв'язок».

Логічна й фізична моделі даних будуть будуватися за допомогою CASE-засобу Erwin [7]. Erwin має два рівні вистави моделі – логічний і фізичний. Case-технологія [7] дозволяє створювати не тільки «правильні» продукти, але й забезпечити «правильний» процес їх створення. Основна мета Case-технології полягає в тому, щоб відокремити проектування ПЗ від його кодування й наступних етапів розробки, а також сховати від розроблювачів усі деталі середовища розробки й функціонування ПЗ.

### 3 ОПИС ПРОЕКТУВАННЯ АЛГОРИТМІВ СИСТЕМИ

#### 3.1 Моделі агентів пошуку

Агент пошуку необхідного товару усередині бренду автоматично включається при виборі користувачем будь-якого товару в електронному магазині. Агент здійснює пошук двох товарів, які максимально відповідає обраному користувачем товару.

Пошук ведеться за наступними критеріями:

- бренд;
- модель;
- колір;
- розмір;
- ціна.

У результаті пошуку агент пропонує користувачеві два товари, які максимально відповідають критеріям пошуку.

Таким чином, при виборі товару клієнтові відразу надається ще два додаткові товари, які схожі на перший і належать до обраного бренда.

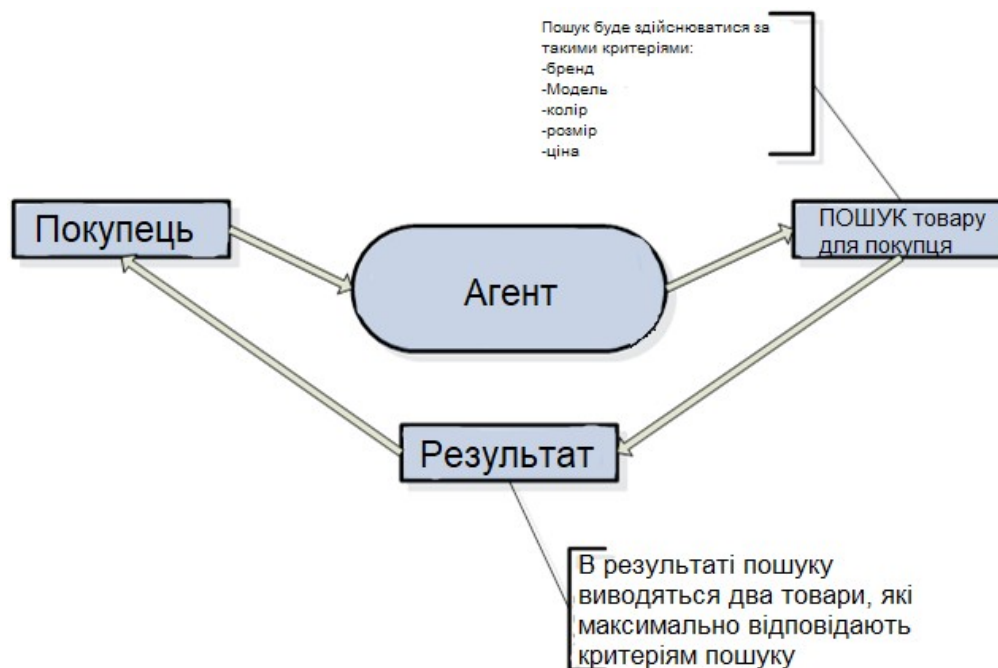


Рисунок 3.1 – Модель агента пошуку товару

Агент пошуку серед усіх брендів також включається автоматично при виборі користувачем будь-якого товару в електронному магазині. Але головна відмінність полягає в тому, що пошук здійснюється не усередині обраного бренду, а серед усього товару.

Пошук товару ведеться за наступними критеріями:

- стиль;
- модель;
- колір;
- розмір;
- ціна.

Результатом є два товари, які максимально відповідають критеріям пошуку.

При виборі товару клієнтові відразу надається ще два додаткові товари, які схожі на перший, але не належать до обраного бренду (рис. 3.2) .

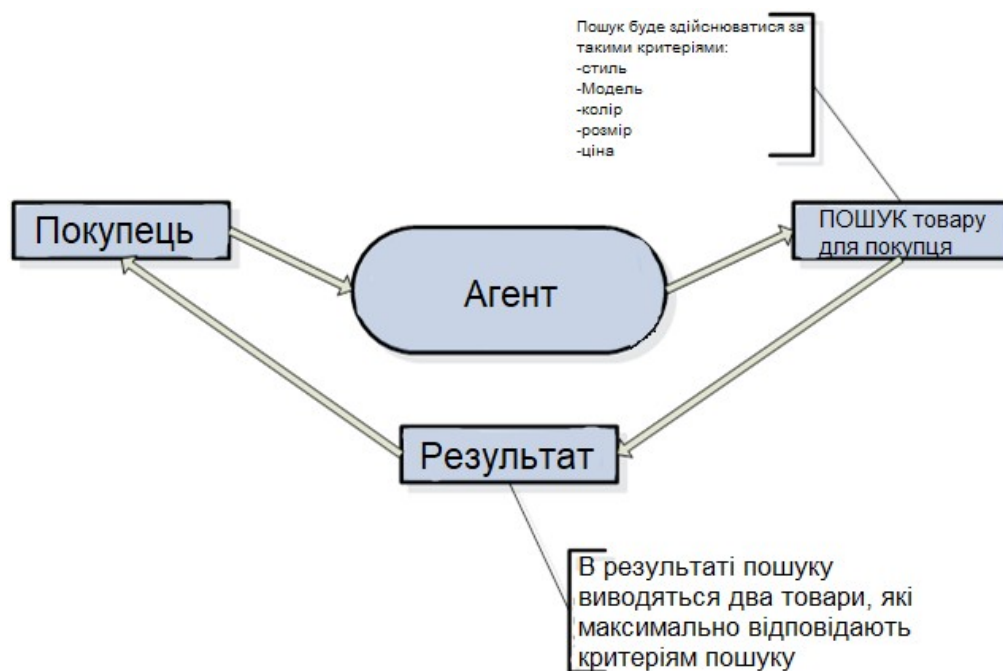


Рисунок 3.2 – Модель агента пошуку серед брендів

Агент пошуку інформації з останніх тенденцій молодіжної моди запускається щодня й проглядає список установлених інтернет-журналів на наявність новин у світі молодіжної моди. Якщо є новини щодо брендів, колекцій,

товарів, які продаються в магазині, то агент зберігає посилання на цю інформацію.

Отримані посилання потім обробляються адміністратором і інформація заноситься в блог магазину.

Також при виборі покупцем товару агент здійснює пошук інформації пов'язаної з обраним товаром у блозі електронного магазину. Інформацією може бути стаття, присвячена цьому товару, колекції або бренду.

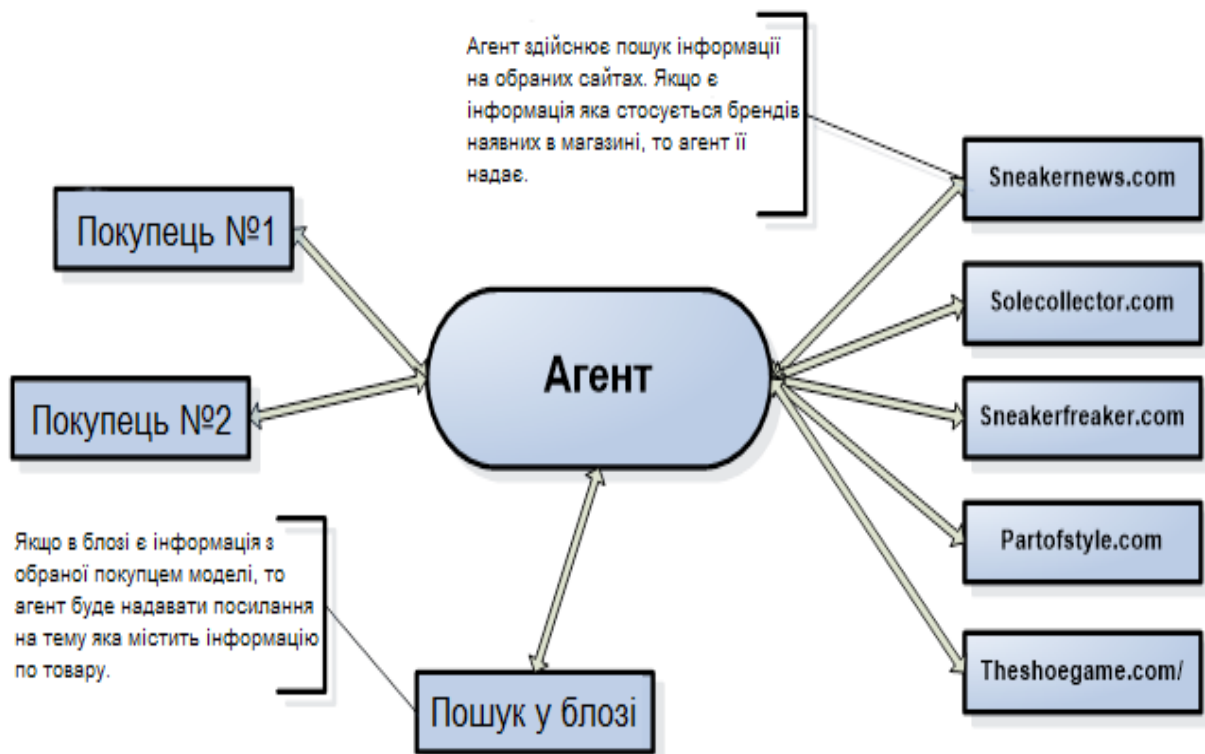


Рисунок 3.3 – Модель агента пошуку інформації трендів

У результаті роботи цих трьох агентів користувач одержує додаткову інформацію з обраного товару. Також надається схожий товар обраного бренду й товар інших брендів. Тобто якщо користувач вибрав якусь модель і прагне подивитися щось схоже, то йому не треба пролістувати увесь бренд або продивлятися увесь товар, йому досить подивитися те, що йому надав агент. При цьому користувачеві не будуть ставити ніякі запитання, агент буде все робити ґрунтуючись на діях користувача.

### 3.2 Модель бізнес-процесів електронного магазину

Розглянуто клас більш складних систем, що мають наступні властивості: існує таке розчленовування системи на елементи, при яким кожний отриманий елемент являє собою агрегат.

Надалі такого роду складні системи будуть називатися агрегативними системами або *A*-системами.

Кожний елемент *A* – системи, будучи в загальному випадку агрегатом, не обов'язково повинен мати повний комплекс властивостей агрегату; він може бути й більш простим об'єктом, що представляють собою окремий випадок агрегату. Разом з тим серед елементів *A*-системи не може бути жодного елемента, які не був би агрегатом (з повним комплексом властивостей або часткам случаємо).

Уся інформація, що циркулює в *A*-системі, ділиться на зовнішню (таку, що отримана ззовні від об'єктів, що не є елементами даної системи) і внутрішню, що вироблена агрегатами самої системи. Обмін інформацією між *A*-системою й зовнішнім середовищем відбувається через агрегати, називані полюсами системи. Так, є вхідні полюси, що представляють собою агрегати, для яких вхідна інформація, що надходить у вигляді  $(x)$  – повідомлень  $[x]$  є повністю або частково зовнішньої.

Будь-яка сукупність агрегатів називається агрегативною системою, якщо передача інформації між агрегатами відбувається миттєво й без викривлень.

Властивості *A*-системи визначаються не тільки властивостями складових агрегатів, але також і її структурою. Розгляд структури почнемо з визначення співвідношень взаємодії між агрегатами. Два агрегати *B* и *C* називаються безпосередньо зв'язаними, якщо між ними здійснюється пряма передача інформації, тобто вихідна інформація агрегату *B* є вхідною й керуючою для агента *C*, або навпаки. Крім безпосередньо зв'язаних агрегатів, ми будемо також розглядати просто зв'язані агрегати. Якщо агрегати *B* и *C* не є зв'язаними, то ми будемо їх називати незв'язаними.

Розглянуто деякі види зв'язку між агрегатами. Агрегат  $C$  безпосередньо іде за агрегатом  $B$ , якщо деяка частина вихідної інформації агрегату  $B$  є частиною вхідної інформації агрегату  $C$ .

Аналогічно, агрегат  $B$  безпосередньо управляє агрегатом  $C$  (агрегат  $C$  безпосередньо підлеглий агрегату  $B$ ), якщо деяка частина керуючої інформації агрегату  $C$  являє собою частину вихідної інформації агрегату  $B$ . Даний агрегат може безпосередньо управляти декількома агрегатами або бути безпосередньо підлеглим декільком агрегатам  $A$ -систем.

### 3.3 Розробка алгоритму використання агрегативних моделей

У математичній теорії графів і інформатиці граф – це сукупність об'єктів зі зв'язками між ними. Об'єкти представляються як вершини, або вузли графа, а зв'язки – як дуги, або ребра. Для різних областей застосування види графів можуть різнитися спрямованістю, обмеженнями на кількість зв'язків і додатковими даними про вершини або ребра.

Багато структур, що представляють практичний інтерес у математику й інформатиці, можуть бути представлені графами. Граф або неорієнтований граф  $G$  – це впорядкована пара  $G = (V, E)$ , для якої виконані наступні умови:

- $V$  – множина вершин або вузлів,
- $E$  – множина пар (у випадку неорієнтованого графа – неупорядкованих) вершин, названих ребрами.

Множина  $V$  (а значить і  $E$ ) зазвичай вважаються кінцевими множинами. Вершини й ребра графа називаються також елементами графа, число вершин у графові  $|V|$  – порядком, число ребер  $|E|$  – розміром графа. Вершини  $u$  і  $v$  називаються кінцевими вершинами (або просто кінцями) ребра  $e = \{u, v\}$ . Ребро, у свою чергу, з'єднує ці вершини. Дві кінцеві вершини того самого ребра називаються сусідніми. Два ребра називаються суміжними, якщо вони мають

загальну кінцеву вершину. Два ребра називаються кратними, якщо множина їх кінцевих вершин збігаються. Ребро називається петлею, якщо його кінці збігаються, тобто  $e = \{v, v\}$ . Ступенем  $\deg v$  вершини  $V$  називають кількість ребер, для яких вона є кінцевою (при цьому петлі вважають двічі).

Вершина називається ізольованою, якщо вона не є кінцем ні для одного ребра; висячою (або листом), якщо вона є кінцем рівно одного ребра.

Орієнтований граф (скорочено оргграф)  $G$  – це впорядкована пара  $G = (V, A)$ , для якої виконані наступні умови:

- $V$  – множина вершин або вузлів,
- $A$  – множина (упорядкованих) пар різних вершин, названих дугами або орієнтованими ребрами.

Дуга – це впорядкована пара вершин  $(v, w)$ , де вершину  $v$  називають початком, а  $w$  – кінцем дуги. Можна сказати, що дуга  $v w$  веде від вершини  $v$  до вершини  $w$ .

Змішаний граф  $G$  – це граф, у якому деякі ребра можуть бути орієнтованими, а деякі – неорієнтованими. Записується впорядкованою трійкою  $G = (V, E, A)$ , де  $V$ ,  $E$  і  $A$  визначені так само, як вище. Зрозуміло, що орієнтований і неорієнтований графи є окремими випадками змішаного.

Шляхом (або ланцюгом) у графові називають кінцеву послідовність вершин, у якій кожна вершина (крім останньої) з'єднана з наступною в послідовності вершин ребром.

Орієнтованим шляхом в оргграфі називають кінцеву послідовність вершин  $v_i$ , для якої все пари  $(v_i, v_{i+1})$  є (орієнтованими) ребрами.

Циклом називають шлях, у якому перша й остання вершини збігаються. При цьому довжиною шляху (або циклу) називають число складових його ребер. Помітимо, що якщо вершини  $u$  і  $v$  є кінцями деякого ребра, те згідно з даним визначенням, послідовність  $(u, v, u)$  є циклом. Щоб уникнути таких «вироджених» випадків, вводять наступні поняття. Шлях (або цикл) називають простим, якщо ребра в ньому не повторюються; елементарним, якщо він простий і вершини в ньому не повторюються. Нескладно бачити, що:

- усякий шлях, що з'єднує дві вершини, містить елементарний шлях, що з'єднує ті ж дві вершини;
- усякий простий неелементарний шлях містить елементарний цикл;
- усякий простий цикл, що проходить через деяку вершину (або ребро), містить елементарний ( під-)цикл, що проходить через ту ж вершину (або ребро).

Бінарне відношення на множині вершин графа, задане як «існує шлях з  $u$  в  $v$ », є відношенням еквівалентності, і, отже, розбиває ця множина на класи еквівалентності, названі компонентами зв'язності графа. Якщо в графа рівно один компонент зв'язності, то граф зв'язний. На компоненті зв'язності можна ввести поняття відстані між вершинами як мінімальну довжину шляху, що з'єднує ці вершини.

Усякий максимальний зв'язний підграф графа  $G$  називається зв'язним компонентом (або просто компонентом) графа  $G$ . Слово «максимальний» означає максимальний щодо включення, тобто, що не втримується у зв'язковому підграфі з більшим числом елементів

Ребро графа називається мостом, якщо його видалення збільшує число компонентів.

Далі моделі агентів будуть представлені й описані у вигляді орієнтованих графів.

На рис. 3.4 представлена модель агента пошуку необхідного товару усередині бренду зображена у вигляді графа.

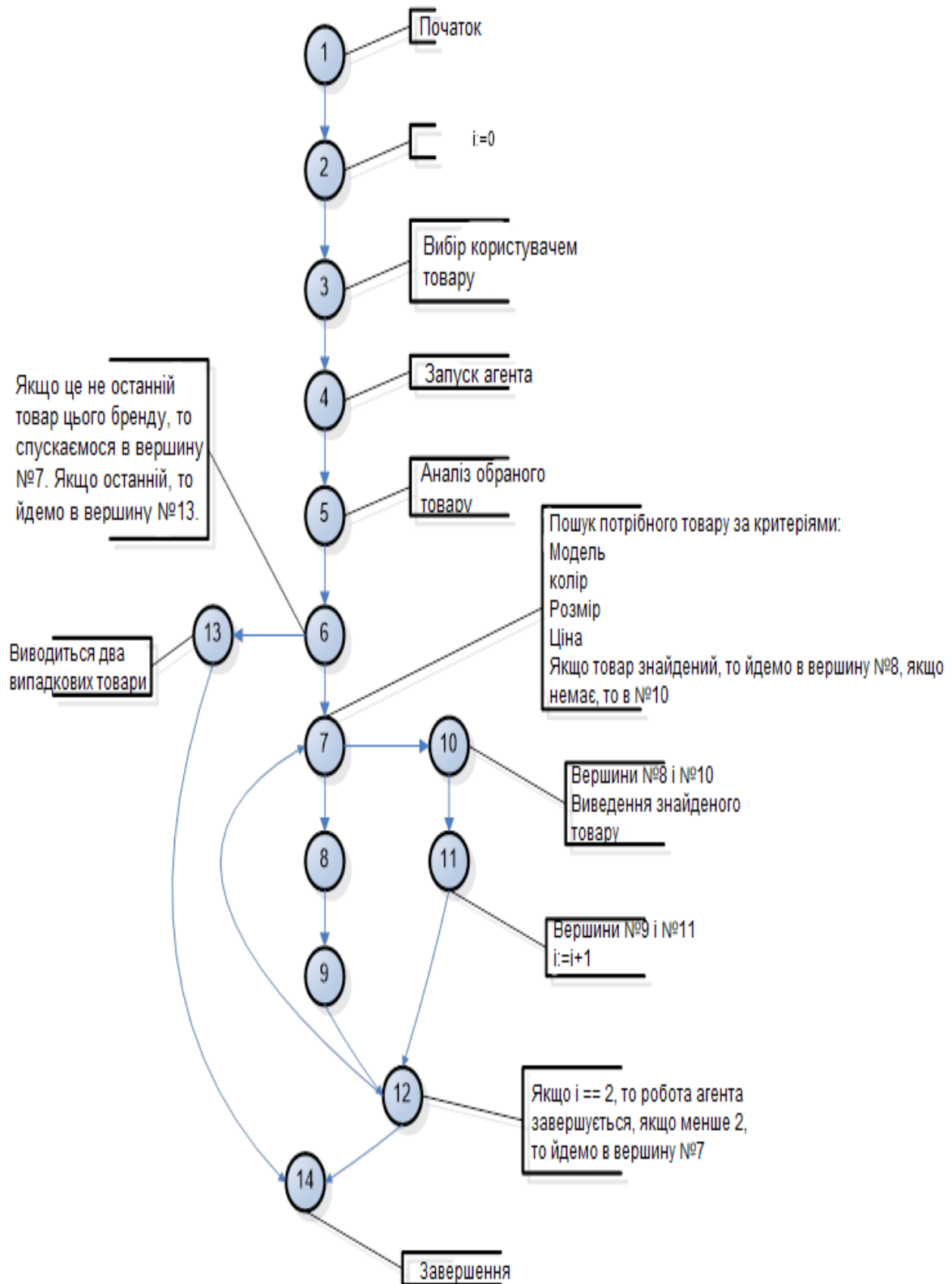


Рисунок 3.4 – Граф моделі агента пошуку товару

На рис. 3.5 представлена модель агента пошуку необхідного товару серед усього товару у вигляді графа.



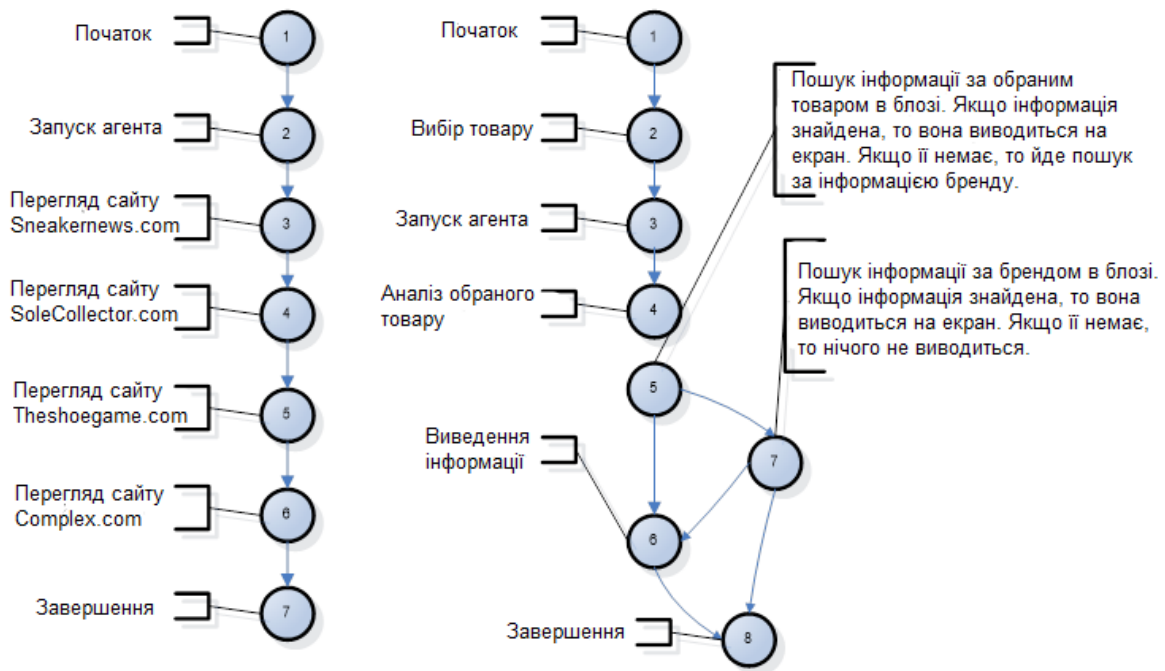


Рисунок 3.6 – Граф моделі агента пошуку інформації

### 3.4 Проектування БД інтернет-магазину

Логічний рівень – це абстрактний погляд на дані, на ньому дані представляються так, як виглядають у реальному світі, і можуть називатися так, як вони називаються в реальному світі. Об'єкти моделі, що представляються на логічному рівні, називаються сутностями й атрибутами. Логічна модель даних є універсальною й ніяк не пов'язана з конкретною реалізацією СКБД.

Після аналізу предметної області, були виділені наступні сутності БД:

- сутність «Товари» з атрибутами: номер товару, назва, повна інформація, категорія до якої він належить, оцінка покупців, справжня ціна, стара ціна, зображення, чи доступний товар на складі, кількість, що проголосували за товар, кількість проданих товарів, чи відображати товар на вітрині, короткий опис властивостей товару, внутрішній код товару;

- сутність «Замовлення» з атрибутами: номер замовлення, дата замовлення, прізвище, ім'я покупця, e-mail, країна, поштовий індекс, область, місто, адреса, телефон;
- сутність «Замовлення\_Кошик» з атрибутами: номер товару, номер замовлення, назва товару, ціна, кількість;
- сутність «Категорії» з атрибутами: номер категорії, назва, батьківська категорія, кількість товарів у даній категорії, опис, зображення;
- сутність «Спеціальні пропозиції» з атрибутами: номер пропозиції, номер продукту, порядок у списку висновку.

Після аналізу предметної області, були виділені наступні зв'язки між сутностями. На рис. 3.7 представлені зв'язки між сутностями.

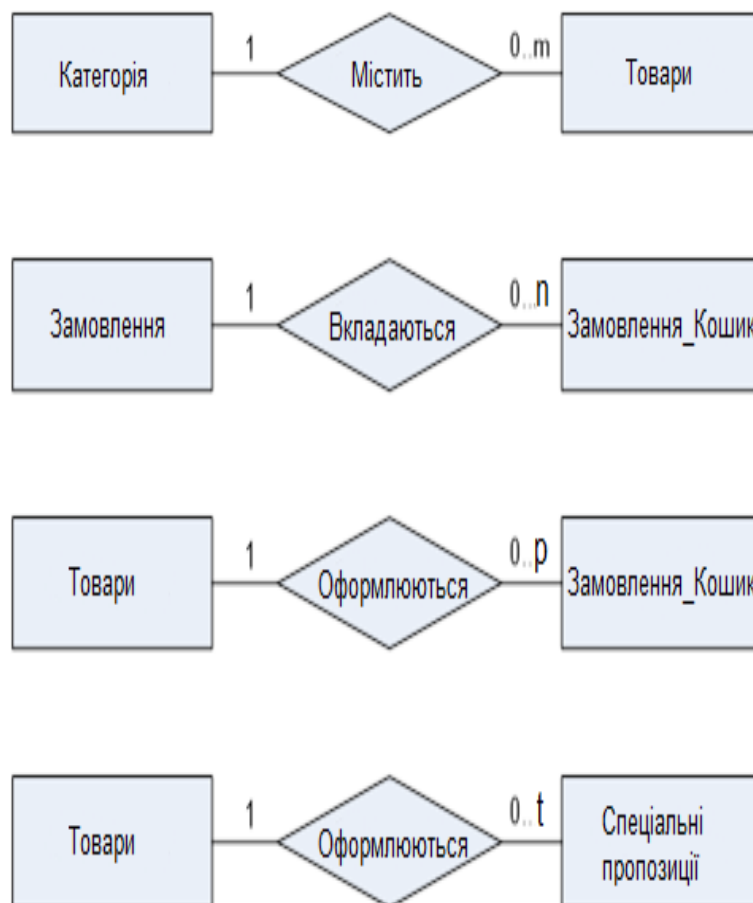


Рисунок 3.7 – Зв'язки між сутностями БД

На рисунку 3.8 представлена Ег-діаграма розроблювальної БД.

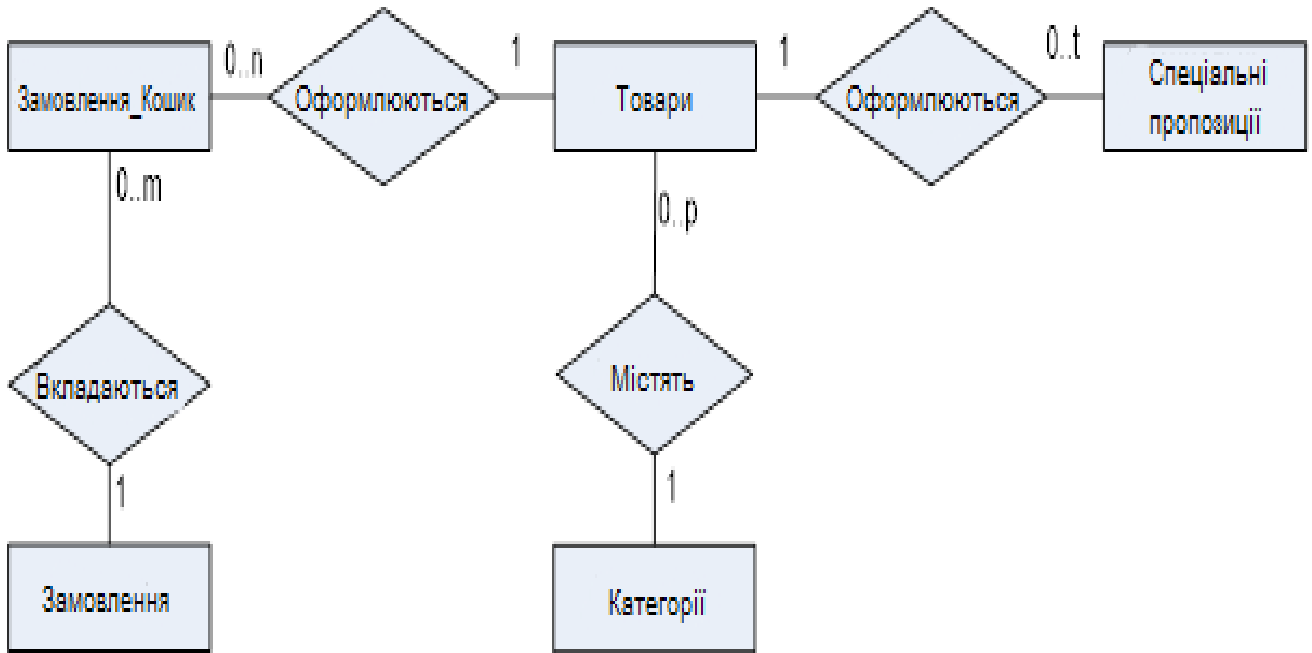


Рисунок 3.8 – Ег-діаграма БД

На рис. 3.9 представлена логічна модель БД.

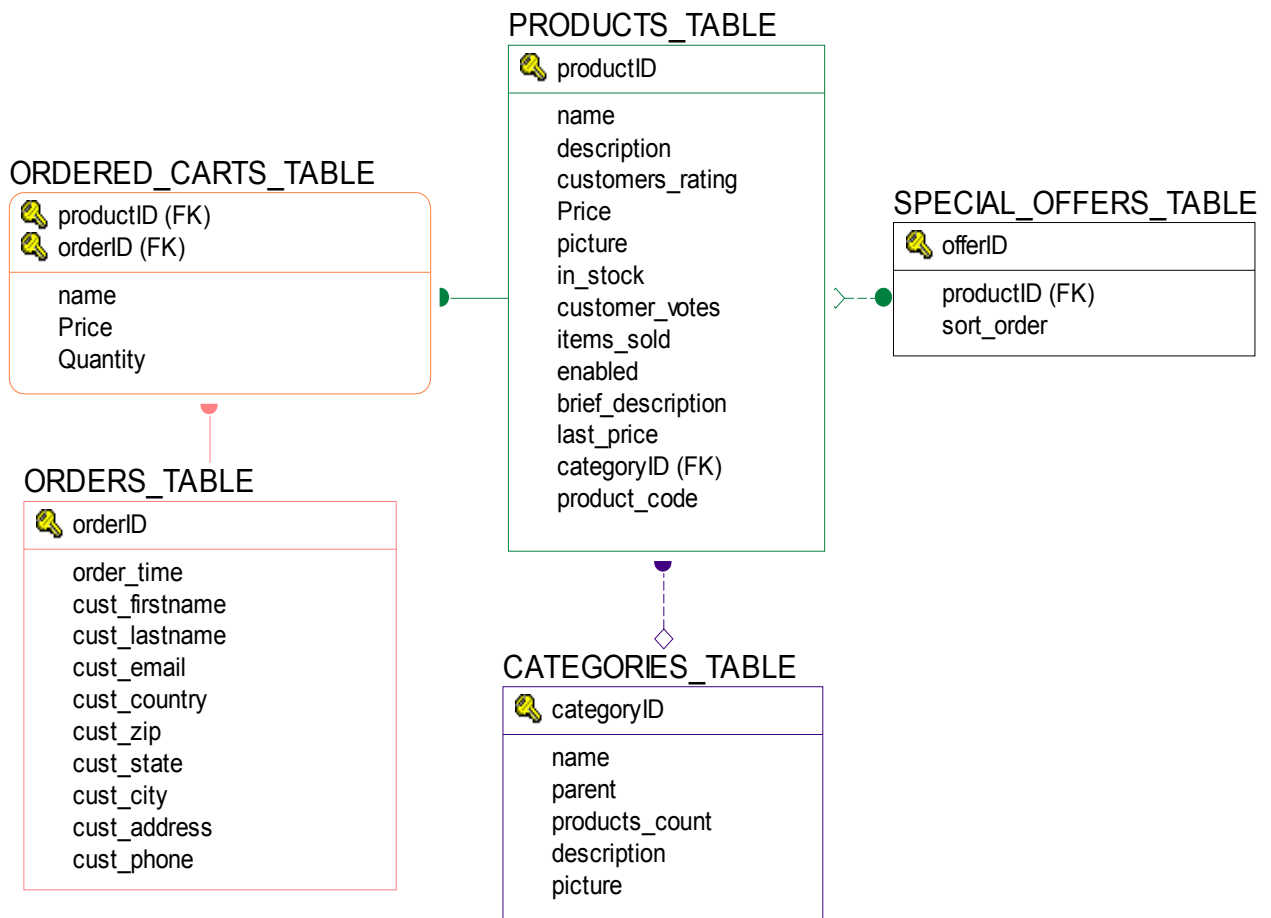


Рисунок 3.9 – Логічна модель БД інтернет-магазину

Фізична модель даних залежить від конкретної реалізації СКБД, фактично будучи відображенням системного каталогу. У фізичній моделі міститься інформація про всі об'єкти БД. Оскільки стандартів на об'єкти БД не існує (наприклад, немає стандарту на типи даних), фізична модель залежить від конкретної реалізації СКБД.

На рис 3.10 представлена фізична модель БД.

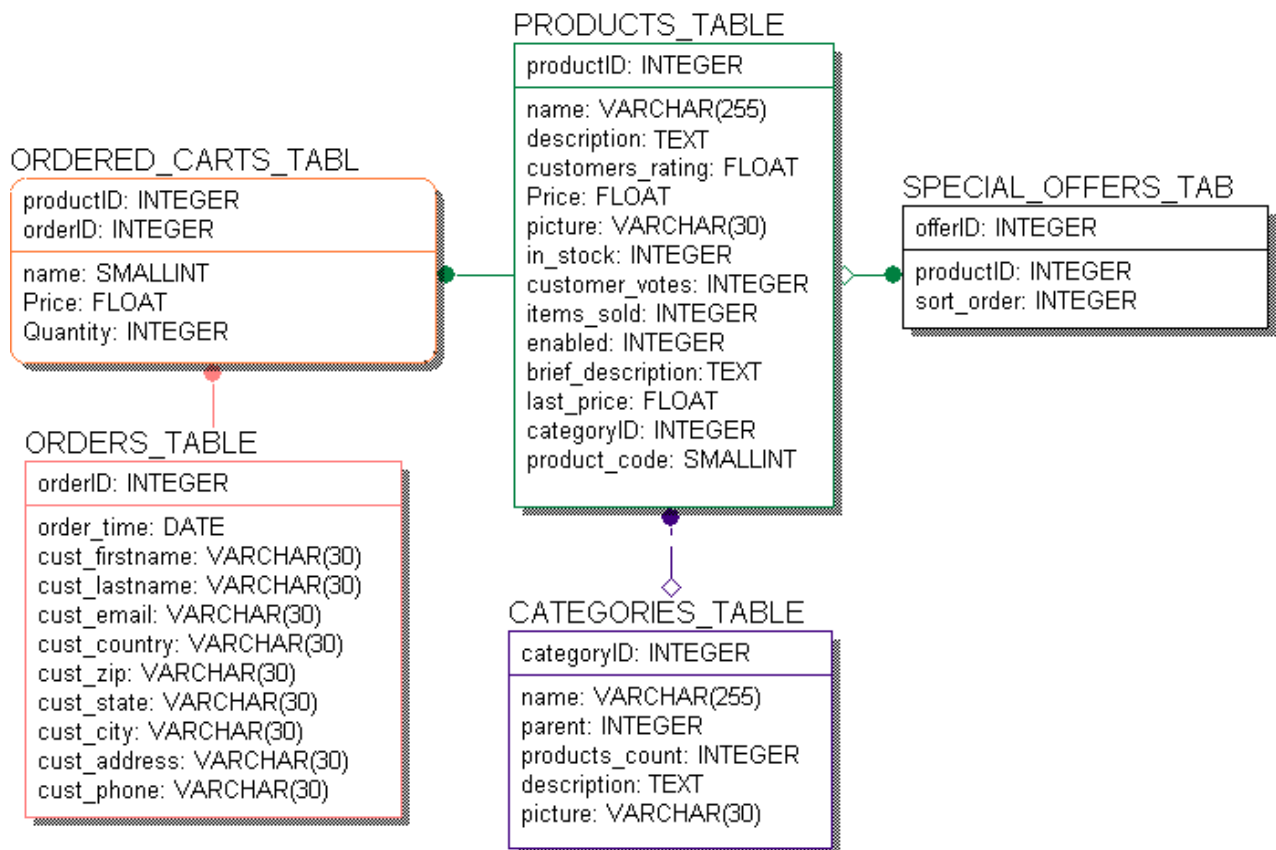


Рисунок 3.10 – Фізична модель БД

Тобто було виконане моделювання предметної області: розроблена концептуальна модель ПЗ, логічна й фізична модель бази даних. Розроблені моделі агентів, що представлені у формі агрегатів і графів.

## 4 ОПИС РОЗРОБЛЕНОГО ПЗ АГЕНТНОГО ТИПУ

### 4.1 Обґрунтування вибору алгоритмічних засобів

На рис. 4.1 представлена схема алгоритму агента пошуку товару.

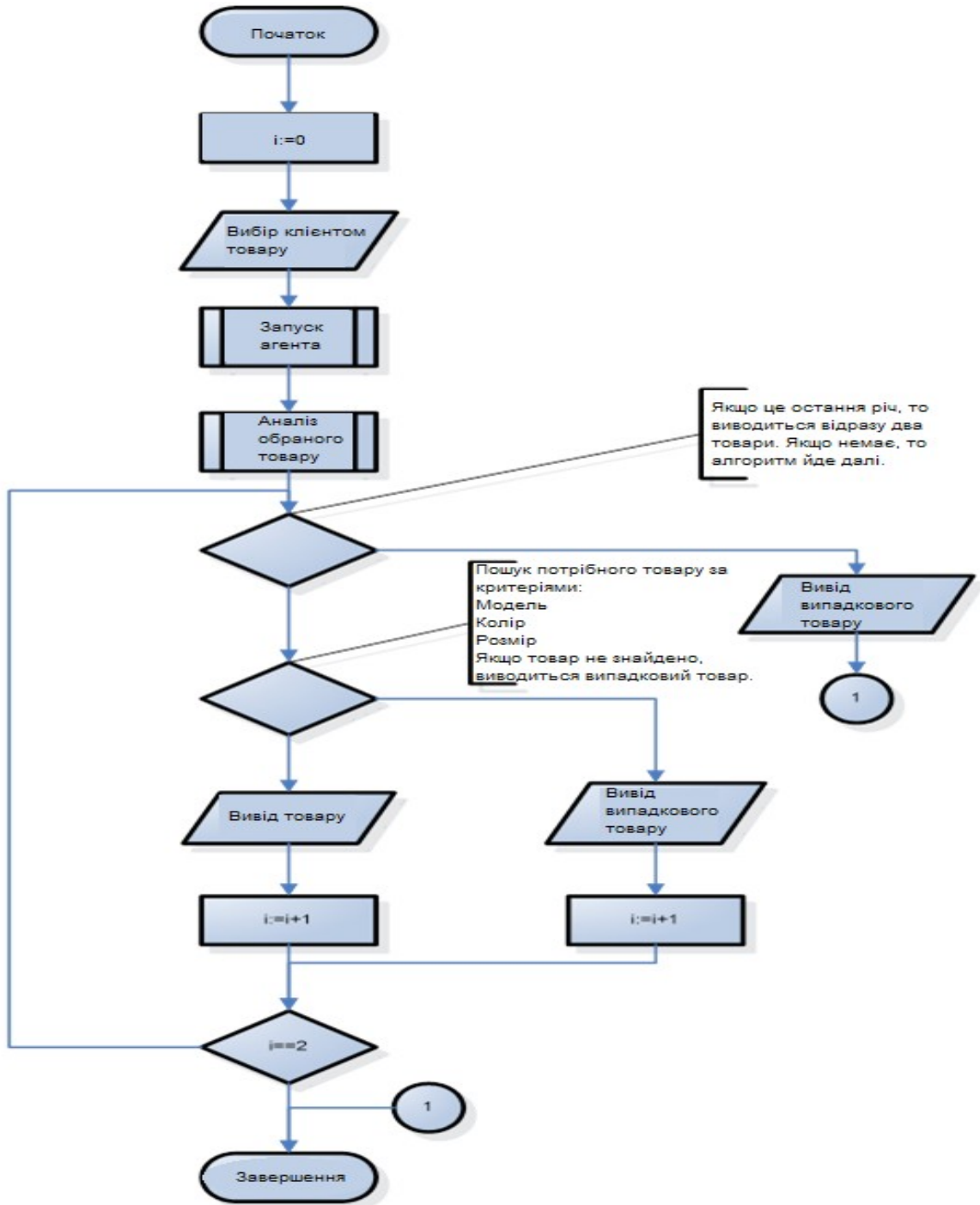


Рисунок 4.1 – Алгоритм агента пошуку товару

На рис. 4.2 представлена схема алгоритму агента пошуку моделі в межах усього товару.

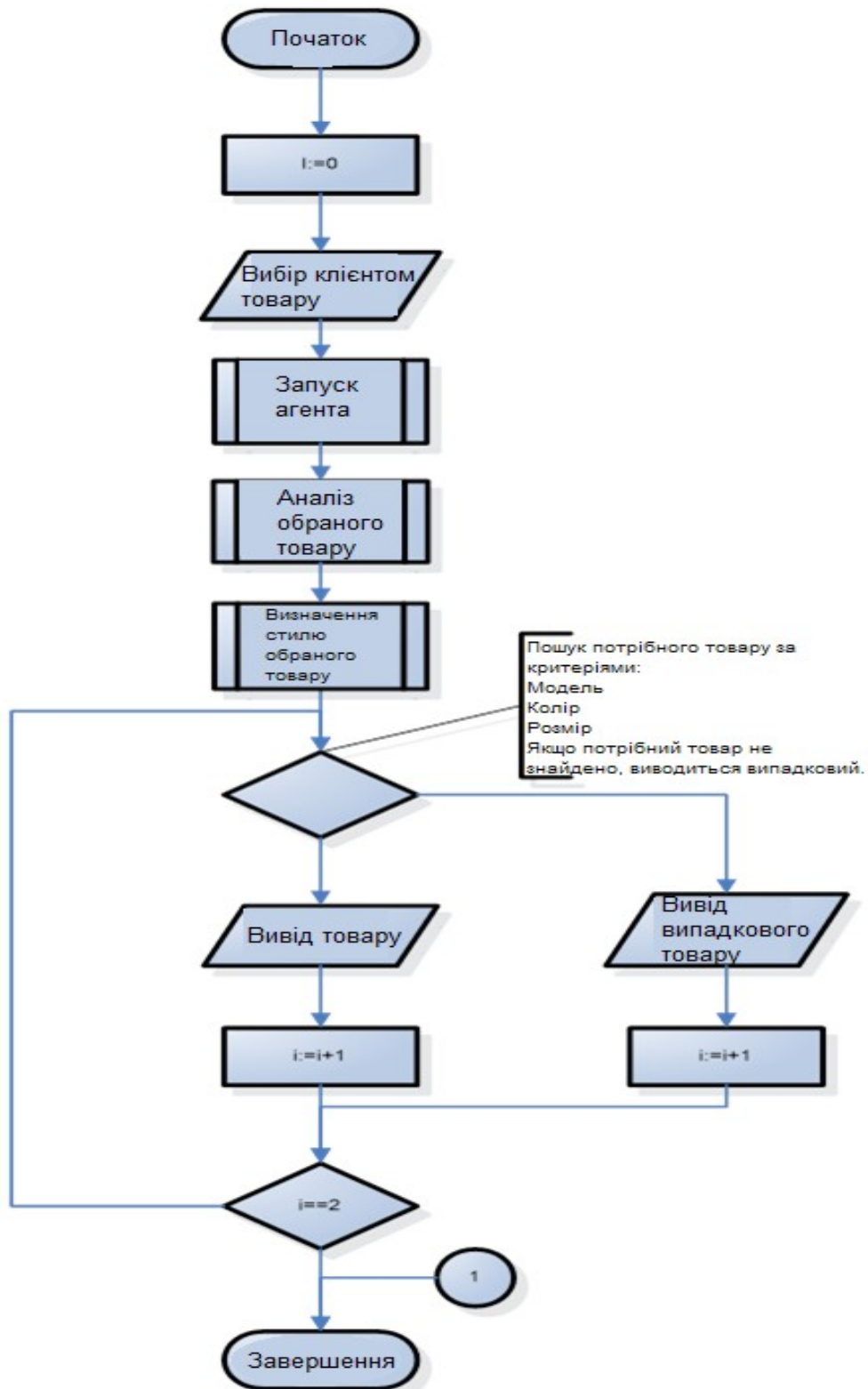


Рисунок 4.2 – Алгоритм агента пошуку товару

На рис. 4.3 представлена схема алгоритму агента пошуку інформації.

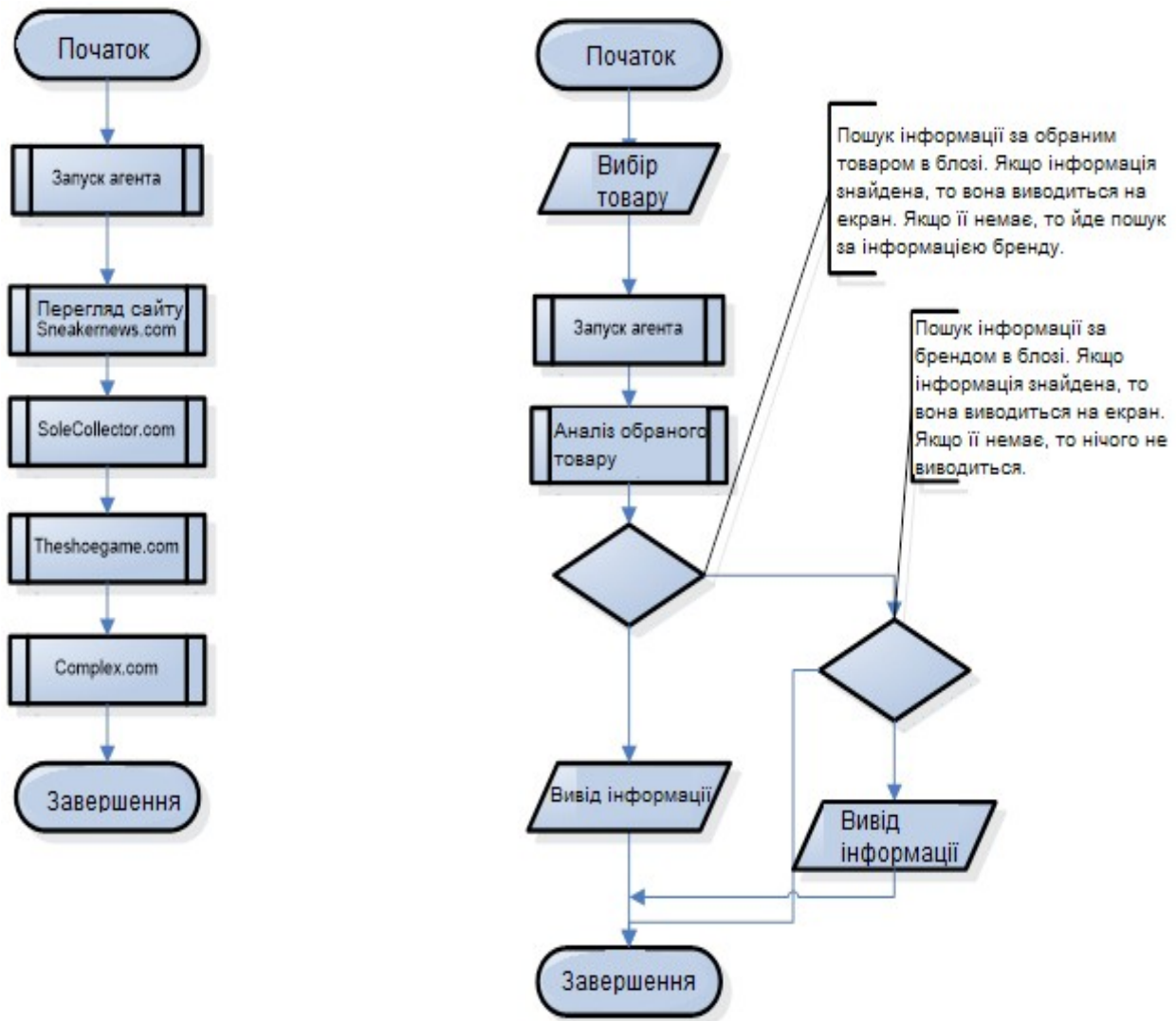


Рисунок 4.3 – Алгоритм агента пошуку інформації

## 4.2 Вихідні дані на розробку прототипу ПЗ

Даний прототип ПЗ призначений для експериментальної перевірки можливості застосування моделей клієнт-серверного додатка разом з агентною технологією. ПЗ повинне забезпечувати роботу в режимах «адміністратор», «фахівець-продавець», «користувач».

Зберігання даних про товари повинне здійснюватися за допомогою БД і містити наступні відомості: назва, повна інформація, категорія до якої він належить, оцінка покупців, справжня ціна, стара ціна, зображення, чи доступний

товар на складі, кількість, що проголосували за товар, чи відобразити товар на вітрині, короткий опис властивостей товару, внутрішній код товару.

Зберігання даних про товари повинне здійснюватися за допомогою БД і містити наступні відомості:

- назва, повна інформація,
- категорія до якої він належить, оцінка покупців, справжня ціна,
- стара ціна, зображення, чи доступний товар на складі, кількість
- що проголосували за товар, чи відобразити товар на вітрині, коротке
- опис властивостей товару, внутрішній код товару.

ПЗ, що розробляється, повинне забезпечувати можливість внесення змін у процесі роботи. Зберігання даних про замовлення повинне здійснюватися за допомогою БД і містити наступні відомості: дата замовлення, прізвище, ім'я покупця, e-mail, країна, поштовий індекс, область, місто, адреса, телефон. Додавання, зміна й видалення даних про товари із БД. Відновлення бази даних після додавання, зміни, видалення даних про товари із БД. Відображення товарів доступних для здійснення покупки й приховання тих товарів, які за якимись причинами не доступні.

Розроблювальне ПЗ повинне реалізовувати функції, що забезпечують редагування збереженої інформації

### 4.3 Архітектурне проектування ПЗ

На етапі архітектурного проектування визначається перелік компонентів ПЗ і їх інтерфейси взаємодії для встановлення загальних рамок інтегрованого середовища розробки ПЗ. На рис. 4.4 представлена загальна архітектура ПЗ.

На рис. 4.5 представлена діаграма класів розроблювального програмного забезпечення. У табл. 4.1 наведено опис класів.

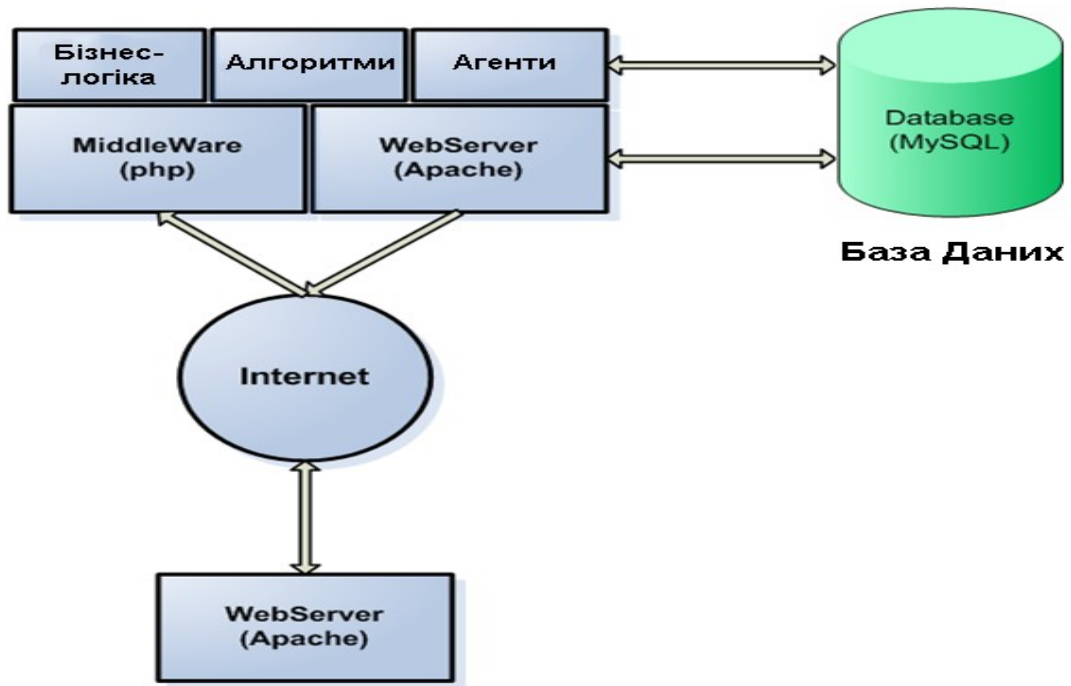


Рисунок 4.4 – Архітектура ПЗ агентного типу

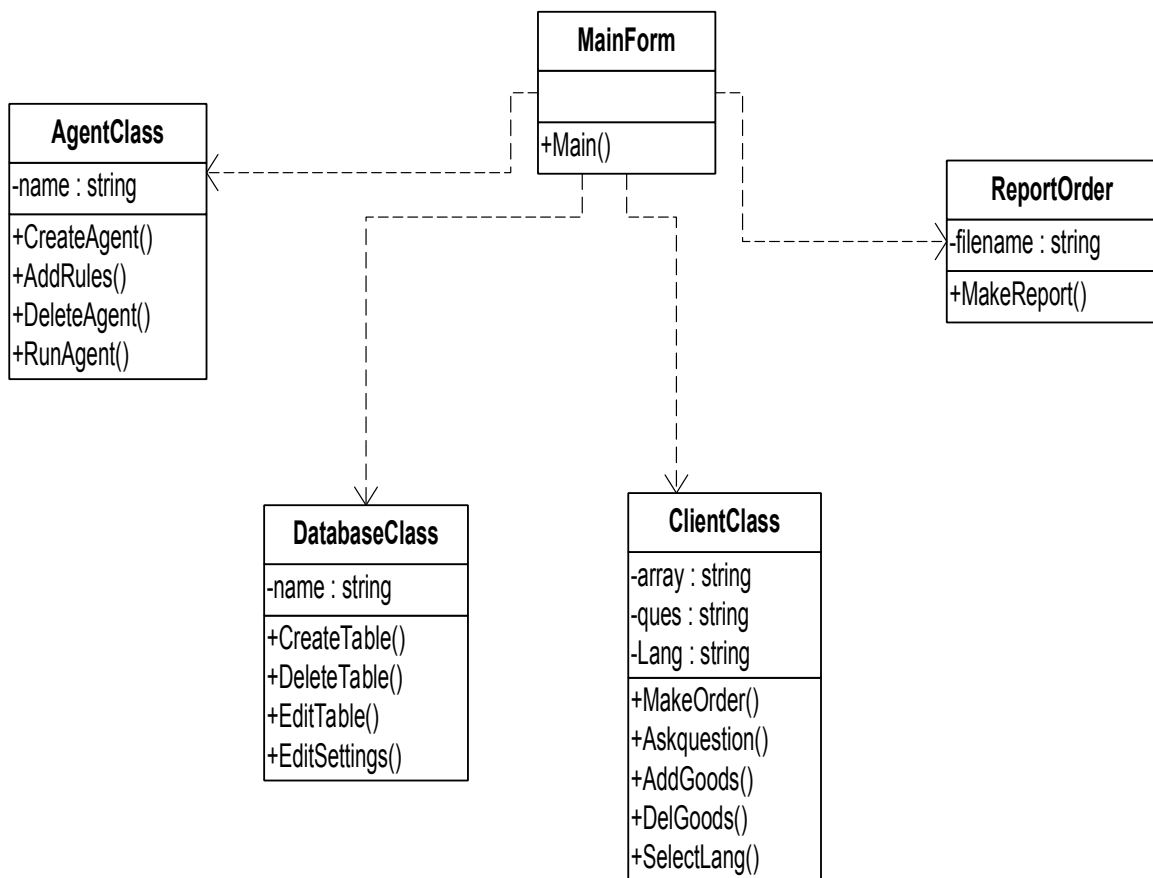


Рисунок 4.5 – Діаграма класів

Таблиця 4.1 – Опис класів

ІМ'Я КЛАСУ	ОПИС
<b>Mainform</b>	Клас головної форми додатка
<b>Agentclass</b>	Клас, що забезпечує створення, редагування, запуск і видалення агентів.
<b>Databaseclass</b>	Клас, що реалізує створення, видалення, редагування таблиць бази даних. А також редагування настроювань.
<b>Clientclass</b>	Клас, що реалізує роботу із клієнтською частиною додатка: складання замовлення, питання на замовлення, додавання й видалення товарів з кошика а також вибір мови.
<b>Report</b>	Клас, що формує структуру звіту про замовлення.

#### 4.4 Опис структурної схеми ПЗ інтернет-магазину

На рис. 4.6 представлена структурна схема ПЗ інтернет-магазину.

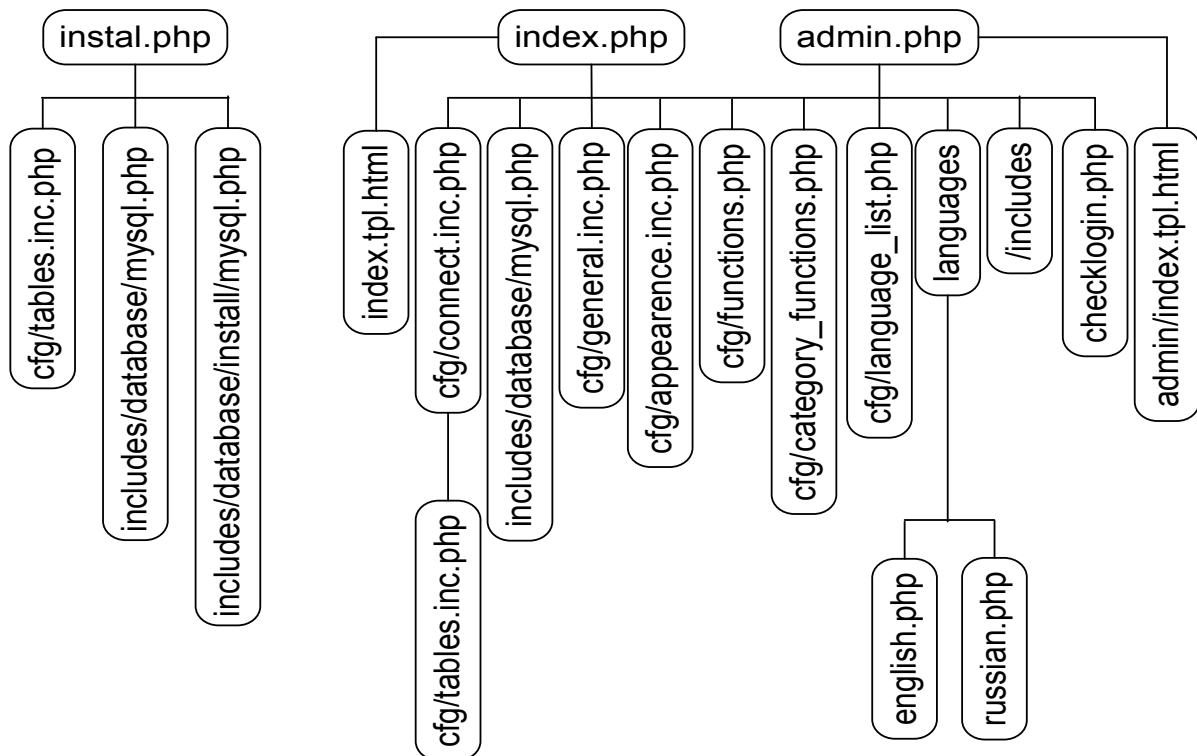


Рисунок 4.6 – Структурна схема ПЗ

У таблиці 4.2 описано модулі ПЗ інтернет-магазину.

Таблиця 4.2 – Опис модулів ПЗ Інтернет-магазину

Каталог	Назва модуля	Опис
<b>Cfg</b>	appearance.inc.php	Визначення констант, відповідальних за настроювання ПЗ
	category_functions.php	Функції заповнення дерева категорій каталогу товарів
	connect.inc.php	Визначення констант для встановлення з'єднання із БД
	functions.php	Часто використовувані функції
	general.inc.php	Визначення констант, відповідальних за настроювання ПЗ
	language_list.php	Опис списку доступних мов інтерфейсу
	tables.inc.php	Визначення констант назв таблиць
<b>Extra</b>	forgot_password.php	Генерація нового пароля адміністратора
<b>Includes\admin\sub</b>	catalog_products_categories.php	Скрипт для редагування админмом каталогу товарів
	catalog_special.php	Скрипт для редакції адміном таблиці «спец. пропозиції»
	conf_appearance.php	Настроювання висновку
	conf_aux.php	Скрипт, який виводить інформацію про компанію
	conf_general.php conf_login_pass.php	Зміна глобальна настроювання ПЗ Інтернет-Магазину
	custord_new_orders.php	Скрипт для редагування админом таблиці замовлень
<b>includes\admin</b>	catalog.php, conf.php, custord.php	Скрипти для редагування адміністратором каталогу товарів
<b>includes\database\install</b>	mysql.php	Скрипт створення таблиць БД
<b>includes\database</b>	mysql.php	Функції роботи із БД
<b>Languages</b>	english.php	Англійська мова інтерфейсу
	russian.php	Українська мова інтерфейсу
<b>Smarty</b>		Бібліотека шаблонів Smarty
	index.php	Головний модуль програми
	install.php	Скрипт процесу, що виконує, установки
	products.php	Керування каталогом продукції (адміністрування)
	style1.css	Стиль форматування

У таблиці 4.3 описуються підпрограми ПЗ інтернет-магазину.

Таблиця 4.3 – Опис підпрограм

Назва модуля	Назва підпрограми	Вхідні/вихідні дані	Опис
functions.php	show_price	Вхідні: \$price – ціна; Вихідні: \$sign_left.\$price.\$sign_right	Повертає ціну й знак обраної валюти
	get_current_time	Вихідні: рядок	Повертає поточний час і дату
	Shownavigator	Вхідні: \$a – кількість елементів у масиві, кількість пройдених сторінок; \$offset – поточний зсув у масиві (відображення елемента [\$offset ... \$offset+\$q]) \$q – кількість елементів на сторінці; \$path – посилання на сторінку	Історія сторінок, навігація по сайту
	validate_search_string	Вхідні/вихідні: \$s – рядок	Обробка рядка SqlpЗапиту
	string_encode	Вхідні/вихідні: \$s – рядок	Шифрування рядка по простому алгоритму
	string_decode	Вхідні/вихідні: \$s – рядок	Розшифрування рядка по простому алгоритму
Category_functions.php	processcategories	Вхідні: \$level – поточний рівень: 0 – ідентифікатор головних категорій, 1 – для підкатегорій; \$path – і шлях від кореневої категорії до обраної (обчислюється calculatepath()) \$sel – іcategoryid обраної категорії; Повертає масив значень (categoryid, name, level)	Повертає масив категорій, які будуть представлені в шаблону category_navigation.tpl.html
Category_functions.php category_functions.php	fillthelist	Вхідні: \$parent – батьківська категорія, \$level – і рівень; Вихідні: \$a – і масив назв категорій	Обробка категорій, заповнення дерева каталогу товарів
	update_products_Count_Value_For_Categories	Вхідні: \$parent – номер батьківської категорії; Вихідні: \$cnt – масив(кількість товарів, колич. Доступних товарів)	Виконує підрахунок кількості товарів у кожній категорії
	Setrightstoupload edfile	\$file_name	Установлює права на доступ до файлу
category.php	deletesubcategories	Вхідні: \$parent – батьківська категорія	Видаляє всі підкатегорії категорії \$parent
	category_Moves_To_Its_Subdirectories	Вхідні: \$cid – родит. категорія, \$new_parent – наступна категорія; Вихідні: true/false	Переміщення по дереву каталогу

Продовження таблиці 4.3

includes\ database\ mysql.php	db_connect	Вхідні: \$host – назва хоста, \$user – ікористувач, \$pass – пароль; вихідні: Mysql link identifier on success, or FALSE on failure.	Створення з'єднання із БД
	db_select_db	Вхідні: \$name – назва БД; вихідні: true або false при помилці	Вибір БД
	db_query	Вхідні: \$s – рядок; Вихідні: true або false	Виконання SqlpЗапиту
	db_fetch_row	Вхідні: \$q – рядок; Вихідні: true або false	Вибір рядка із запиту
includes\ database\ mysql.php	db_insert_id		повертає ID сгенерований для поля з типом AUTO_INCREMENT попереднім запитом INSERT
	db_error		Висновок повідомлення про помилку при роботі із БД

У процесі розробки ПЗ для виявлення помилок використані наступні види дослідної експлуатації:

- автономні тести забезпечують перевірку конкретних модулів і класів, проводилися методом “білого ящика”;
- інтеграційними тестами перевірена коректність роботи модулів у багатомодульній програмі;
- системні тести використані для перевірки коректності роботи всієї програми цілком.

Метою автономного тестування є порівняння функцій, що були реалізовані 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.

## 5 ОПИС МОЖЛИВОСТІ ВИКОРИСТАННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Для проведення тестування обраний метод «білого ящика», який дозволяє досліджувати внутрішню структуру програми. Здійснюється для перевірки інформаційних зв'язків між модулями, знаходження помилок в інтерфейсі між модулями, в інтерфейсах між класами й модулями.

Спочатку тестуються модулі витягу, аналізу й вистави інформації, яка використовується в ПЗ Автоматизації стендових випробувань модуль, що потім управляє модуль і, входу в систему.

Після виправлення помилок було проведено регресійне тестування з метою виключення нових помилок. Спосіб реалізації тестування – покроковий, тобто в порядку підключення певного модуля. Критерій закінчення – відсутність помилок в інформаційних зв'язках між модулями при проходженні тестування покроковим способом. Стратегія тестування – спадна, тому що більша ймовірність виникнення помилки на модулях верхнього рівня.

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.

Мета тестування клієнтської частини:

- тестування користувацького інтерфейсу;
- перевірка працездатності всіх функцій клієнтської частини.

Критерій закінчення системного тестування для серверної частини: одержання всієї необхідної інформації про робочий потік.

Критерій закінчення системного тестування для клієнтської частини: відображення всієї інформації про робочий потік в інтерфейсі користувача

(коректне відображення деревоподібної структури, зведених таблиць і звітів), відсутність модифікацій в аналізованому файлі.

Проведено тестування розробленого ПЗ на контрольному прикладі аналізу коректності робочого потоку.

На рис. 5.1 представлено екранну форму роботи агентів розміщених на сайті <http://originalstore.com.ua/>.

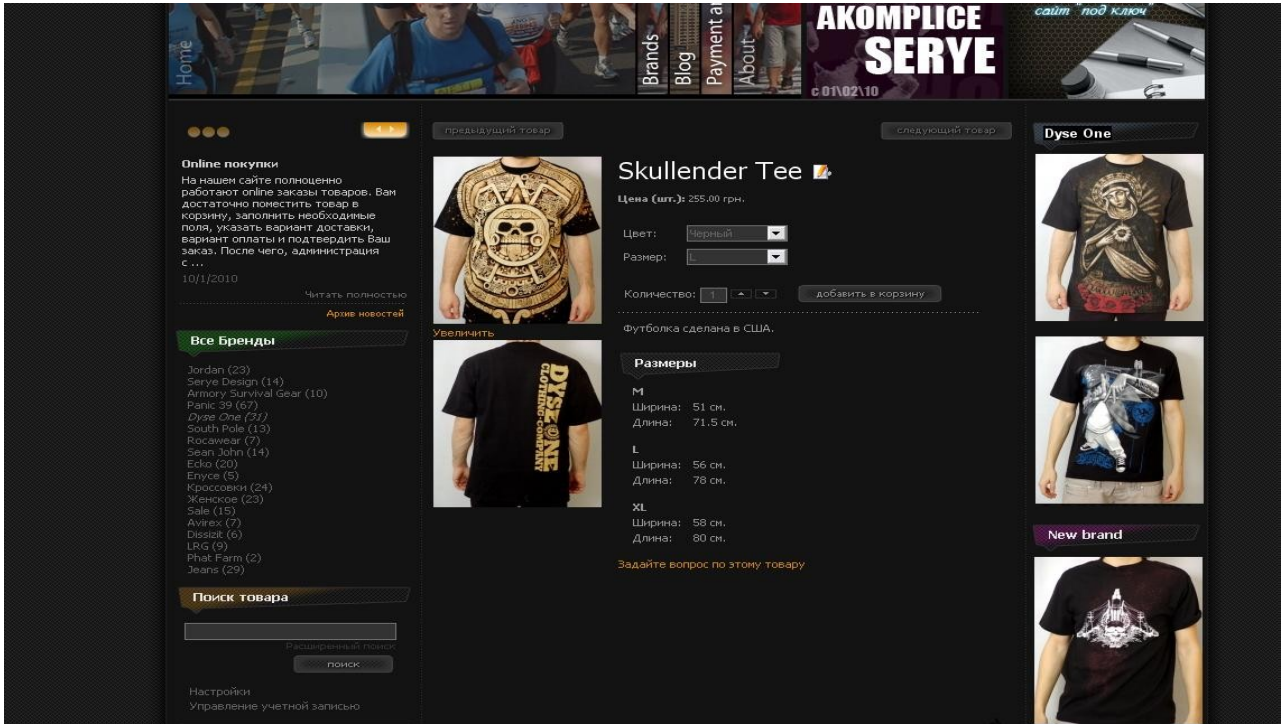


Рисунок 5.1 – Екранна форма інтернет-магазину Originalstore

Агенти були розміщені на сайті починаючи з 5 січня й відключено 5-го квітня. На рис. 5.2 – 5.4 показана порівняльна характеристика роботи інтернет-магазину до застосування агентів і після.

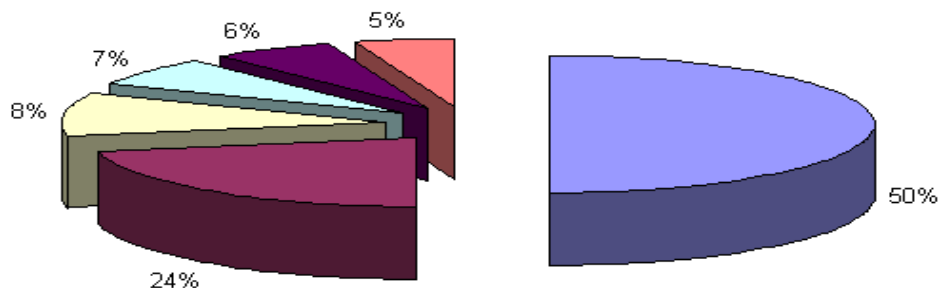


Рисунок 5.2 – Статистика глибини відвідування сайту до використання агентних технологій

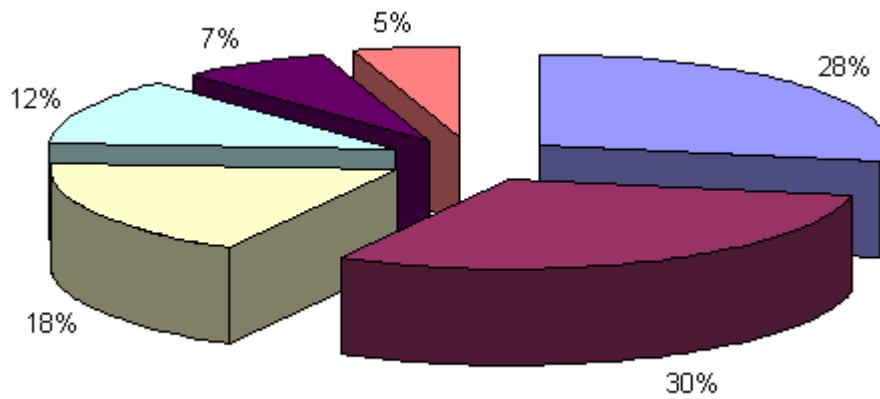


Рисунок 5.3 – Статистика глибини відвідування сайту після використання агентних технологій

	<b>1 сторінка</b>		<b>2-3 сторінки</b>		<b>4-5 сторінок</b>
	<b>6-7 сторінок</b>		<b>більше 8 сторінок</b>		<b>більше 10 сторінок</b>

Рисунок 5.4 – Характеристика значень

У результаті роботи моделі агентів підвищилася глибина перегляду сторінок сайту й час одного перегляду клієнтом інтернет-магазину.

Це свідчить про ефективну роботу агентів.

## ВИСНОВКИ

Агентна технологія дозволяє більш детально враховувати специфіку предметної області Інтернет-магазинів. Застосування агентних технологій дасть можливість легко створювати агентів для поставлених завдань; дозволить редагувати поведінка агентів і відповідно змінювати його в будь-який момент часу.

Є можливість створення агентів, які вирішували б поставлені розроблювачем завдання. При цьому працезатрати на розробку ПЗ зменшуються, що спричиняє підвищення якості кінцевого продукту.

Були розглянуті найбільш важливі алгоритми, розроблені для рішення поставленого завдання:

- алгоритм пошуку обраного товару в межах бренда;
- алгоритм пошуку обраного товару в межах усього товару;
- алгоритм пошуку інформації в мережі інтернет серед зазначених інтернет-ресурсів.

Розроблені алгоритми можуть бути використані при проектуванні програмного забезпечення інтернет-магазину із продажу молодіжного одягу.

Розроблений прототип ПЗ був протестований на предмет працездатності й рішення поставлених завдань. Тестування проводилося на контрольних прикладах, у результаті чого ПЗ показало коректну роботу. На підставі проведення тестів можна заявити, що створений прототип ПО виконує всі функції, який зазначені у вихідних даних на розробку й відповідає всім пред'явленим до нього вимогам.

Удосконалений метод організації й функціонування інтернет магазину із продажу молодіжного одягу, шляхом застосування агентних технологій при розробці програмних засобів у процесах продажу товарів.

Результати рішення завдання атестаційної роботи можуть бути використані при рефакторинге ПЗ інтернет-магазинів.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Адлер, Ю.П. Планирование эксперимента при поиске оптимальных условий. – Рипол Классик. - 2014. - 279 с.
2. Айзерман, М.А., Браверман, Э.М., Розоноэр, Л.И. Метод потенциальных функций в теории обучения машин. – М.: Наука. - 1970. - 384 с.
3. Аркадьев, А.Г., Браверман, Э.М. Обучение машины классификации объектов. – М.: Наука. - 2001. - 192 с.
4. Банди, Б. Методы оптимизации. – Рипол Классик. - 2008. - 128 с.
5. Барсегян, А. Методы и модели анализа данных: OLAP и Data Mining. БХВ-Петербург. - 2004. - 335 с.
6. Джарратано, Д., Райли, Г. Экспертные системы: принципы разработки программирование, 4-е издание. – Издательский дом Вильямс. - 2014. - 1152 с.
7. Дорофеюк, А.А. Алгоритмы автоматической классификации (обзор) // Автоматика и телемеханика. – 2011. – Т. 12. – С. 78-113.
8. Дюран, Б., Оделл, П. Кластерный анализ //М.: Статистика. – 1977. – 128 с.
9. Ершов, А.А. Стабильные методы оценки параметров (обзор) //Автоматика и телемеханика. – 1998. – №. 8. – С. 66-100.
10. Жуков, В.Г., Паротькин, Н.Ю. Исследование дифференцированного адаптивного генетического алгоритма решения задач условной оптимизации //Программные продукты и системы. – 2016. – №. 1 (105). - С. 82-86.
11. Иванов, В.К., Мескин, П.И.. Реализация генетического алгоритма для эффективного документального тематического поиска //Программные продукты и системы. – 2016. – №4(108). - С. 118-126.
12. Крманов, В.Г. Математическое программирование. – М.: Наука. - 1980. - 256 с.
13. Люгер, Д.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание. – Издательский дом Вильямс. - 2013. - 864 с.

14. Растрингін, Л.А. Адаптація складних систем. – Рига: Зинатне. - 1981. - 375 с.
15. Бусленко В. Н., Каганович В. Л., Дашкова Е. Г. Вопросы разработки технологии агрегативного моделирования//Программирование.— 1989. — Харків 1. — С. 15—24
16. Бусленко В. Н. Автоматизация имитационного моделирования сложных систем.—М.- Наука, 1977. — 240 с.
17. Рутковская, Д., Рутковский, Л., Пилиньский, М. Нейронные сети, генетические алгоритмы и нечеткие системы. – М.: Горячая линия-Телеком. - 2013. - 384 с.
18. Bahlmann, C., Haasdonk, B., Burkhardt, H. Online handwriting recognition with support vector machines-a kernel approach //Frontiers in handwriting recognition. – IEEE. - 2012. – pp. 49-54.
19. Bekios-Calfa, J., Buenaposada, J.M., Baumela, L. Robust gender recognition by exploiting facial attributes dependencies //Pattern Recognition Letters. –2016. – Vol. 36. – pp. 228-234.
20. A Guide to the Project Management Body of Knowledge (PMBOK Guide) [Text]. - 5th ed. - Project Management Institute, 2013. - 616 p.
21. Ивашко В.Г., Фін В.К. Експертні системи й деякі проблеми їх інтелектуалізації //Семіотика й інформатика. -1986. - Вып.27. - С.25-61.
22. Кук М.Н., Макдональд Дж. Формальна методологія набуття й представлення експертних знань. ТИИЭР. - Т.74, N 10. - 1986. С.145-155.
23. Представление и использование знаний / Під ред. Х. Уэно, М.Исидзука.- М.: Мир,1989. 220 с.
24. Cohen M.A., Grossberg S. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks//IEEE Trans. Syst., Man, Cybern. 2016. V. 13. N 5. P. 815 – 826.
- 25 Shostak I., Matyushenko I., Romanenkov Yu., Danova M., Kuznetsova Yu. Computer Support for Decision-Making on Defining the Strategy of Green IT, 533–559 (2018), <https://doi.org/10.1007/978-3-030-00253-4>