

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

Факультет прикладної математики  
(назва факультету, інституту)

Кафедра прикладної математики  
(назва кафедри)

До захисту допущено

**Завідувач кафедри**

МОЛЧАНОВ О.А.  
(підпис) (ініціали, прізвище)

“ ” 2008 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

до дипломної роботи (магістерської дисертації) освітньо-кваліфікаційного  
рівня “магістр”  
(назва ОКР)

з напрямку підготовки (спеціальності) 8.0802, Прикладна математика  
(код та назва напрямку підготовки або спеціальності)

на тему: Системне проектування інформаційно-комунікаційних систем.

Дослідження методів продукування та розробки інформаційних ресурсів  
інформаційно-комунікаційних систем

Студент групи КМ-24М  
(шифр групи)

ЛІССОВ Павло Миколайович  
(прізвище, ім'я, по батькові) (підпис)

Керівник проекту

к.т.н., доц., МАСЛЯНКО П.П.  
(вчені ступінь та звання, прізвище, ініціали) (підпис)

Консультант:

Економічно-організаційний розділ к.е.н., доц., БЕРЕЗОВСЬКИЙ К.В.  
(назва розділу ДР) (вчені ступінь та звання, прізвище, ініціали) (підпис)



технології продукування інформаційних ресурсів

б) економічна частина провести ФВА програмного продукту

5.

П

**Перелік графічного (ілюстративного) матеріалу**

1. Об'єкт, предмет, мета дослідження
2. Визначення інформаційного ресурсу
3. Типи інформаційних ресурсів
4. Класифікація інформаційних ресурсів
5. Інтеграція інформаційних ресурсів
6. Етапи та стадії продукування ресурсу
7. Порівняльний аналіз засобів продукування інформаційних ресурсів
8. Ілюстрація роботи Google PageRank
9. Модель ІКС Державної податкової адміністрації України
10. Модель інформаційного ресурсу Державної податкової адміністрації України
11. Апробація результатів дослідження

**6. Консультанти**

з економічних питань к.е.н., доцент, БЕРЕЗОВСЬКИЙ К.В.  
(вчене звання, ПІБ, посада)

7. Дата видачі завдання “01” листопада 2006р.

**Керівник дипломної роботи**

П. П. МАСЛЯНКО  
(підпис) (ініціали, прізвище)

**Завдання прийняв до виконання**

П.М. ЛІССОВ  
(підпис) (ініціали, прізвище)

ЗАТВЕРДЖУЮ

Керівник

дипломного проекту (роботи)

МАСЛЯНКО П.П.

(підпис)

(ініціали, прізвище)

“ \_\_\_ ” \_\_\_\_\_ 2008 р.

## КАЛЕНДАРНИЙ ПЛАН-ГРАФІК

виконання дипломної роботи (магістерської дисертації)

студентом

ЛІССОВИМ П.М.

(прізвище, ініціали)

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання	Позначки керівника про виконання завдань
1	Отримання завдання на магістерську роботу	1.11.2006	
2	Дослідження інформаційних ресурсів, їх типів та характеристик	10.01.2007	
3	Дослідження підходів до вимірювання інформації	1.03.2007	
4	Дослідження засобів автоматизованого продукування інформаційних ресурсів	15.04.2007	
5	Розробка технології продукування інформаційного ресурсу	1.06.2007	
6	Апробація застосування технології у проектах інформатизації	1.02.2008	
7	Проведення функціонально-вартісного аналізу	15.04.2008	
8	Підготовка результатів роботи до захисту	2.06.2008	

Студент

\_\_\_\_\_ (підпис)

## РЕФЕРАТ

**Структура та обсяг роботи.** Магістерська робота складається з вступу, п'яти розділів, висновків, списку використаної літератури з 80 найменувань, 2 додатків та містить 17 таблиць і 16 рисунків.

**Об'єкт дослідження** – інформаційні ресурси інформаційних систем як компонента інформаційно-комунікаційної системи Орг.С.

**Предмет дослідження** – бізнес-процеси системної інженерії: аналізу, проектування, конструювання та розгортання інформаційних ресурсів ІКС.

**Метою роботи** є дослідження методів та розробка технології продукування інформаційних ресурсів.

**Методи дослідження.** При виконанні магістерської роботи використовувались методи прикладного системного аналізу та компонентний підхід до проектування інформаційного ресурсу інформаційно-комунікаційної системи Орг.С. Моделювання проводилось за допомогою методів графічного моделювання, в тому числі UML. Використовувались методи математичного моделювання.

**Результати роботи та наукова новизна одержаних результатів.** В результаті виконання наукових досліджень було запропоновано компонентний підхід до моделювання та розробки інформаційного ресурсу. При цьому:

- досліджені сучасні системи автоматизованого проектування інформаційних ресурсів;
- проведено аналіз та визначені межі застосування підходів до вимірювання кількості інформації у ресурсах;
- запропоновано технології продукування ресурсу як системи компонентів з визначеними інтерфейсами.

Результати роботи застосовуються при розробці інформаційного ресурсу як компоненти ІКС Орг.С.

**Апробація результатів магістерської дисертації.** Основні наукові результати були представлені та обговорювались на ІХ, Х Міжнародній

науково-практичній конференції “Системний аналіз та інформаційні технології” (м. Київ, 2006, 2007, 2008); XIII Всеукраїнській науково-практичній конференції “Сучасні проблеми прикладної математики та інформатики”, (м. Львів, 2006); I, II Всеукраїнській науково-практичній конференції “Сучасні тенденції розвитку інформаційних технологій в науці, освіті та економіці” (м. Луганськ, 2007, 2008); II, III Міжнародній науково-практичній конференції “Інформаційні технології в економічних та технічних системах”, (м. Кременчук, 2007, 2008), VII міжнародній конференції «Інтелектуальний Аналіз Інформації» IAI-2007 (Київ, 2007).

За результатами роботи опубліковано 13 наукових праць, із них 6 у фахових наукових виданнях (5 з яких входять до переліку ВАК). Результати роботи успішно застосовані при реалізації декількох проектів інформатизації.

**Можливі напрямки подальших наукових досліджень** включають інтеграцію показників ефективності у систему проектування, організацію оптимізації проекту, удосконалення підходів до вимірювання інформації.

#### **Ключові слова**

ІНФОРМАЦІЙНИЙ РЕСУРС, ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНА СИСТЕМА, СИСТЕМНЕ ПРОЕКТУВАННЯ ІКС, ГРАФІЧНЕ МОДЕЛЮВАННЯ, ПРОДУКУВАННЯ ІНФОРМАЦІЙНИХ РЕСУРСІВ, ІНФОРМАЦІЙНИЙ КОЛЕКТОР

## SUMMARY

**The structure and the scope of the thesis.** The master thesis consists of introduction, five chapters, conclusion, list of references with 80 items, 2 appendixes and includes 17 tables and 16 figures.

**The object of research** – an informative resource of informative-communicative system of organization system

**The subject of research** – system engineering business-processes for informative resource: analysis, design, creation and deployment.

**The goal of the thesis** is the research of methods and the development of the technology for informative resources production.

**Research methods.** Applied system analysis and component approach were used for design process of informative resource of informative-communicative system. Modeling was made with visual modeling methods, such as UML. Mathematical modeling was used.

Research results and the scientific novelty. As a result of the given scientific research component approach for informative resources design and development was offered. The results of research are the following:

- modern automatized design tools for informative resources were studied;
- the analysis of information measurement methods was made and the possibilities of their application for several types of resources were studied;
- the technology of resource production as a set of components with defined interfaces was offered.

**The thesis results are recommended to use** for development of informative resource as a component of an informative-communicative system.

**Master thesis result approbation.** The main scientific results were presented and discussed on: IX, X International scientific-practical conference “System

analysis and information technologies” (Kyiv, 2007, 2008); XIII Ukrainian scientific-practical conference “Modern problems of applied mathematics and informatics”, (Lviv, 2006); I, II Ukrainian scientific-practical conference “Modern information technologies trends in science, education and economics” (Lugansk, 2007, 2008); II, III International scientific-practical conference “Information technologies in economical and technical systems”, ( Kremenchuk, 2007, 2008).

By the results of research 13 scientific papers were published, including 6 articles in specialized scientific journals. Effect of technology usage was proved with development of several informatisation projects.

**Further research** may focus integration of efficiency parameters in the production system, project optimization and enhancement of information measurement approaches.

### **Keywords**

INFORMATIVE RESOURCE, INFORMATIVE-COMMUNICATIVE SYSTEM, ICS SYSTEM DESIGN, VISUAL MODELLING, INFORMATIVE RESOURCE PRODUCTION, INFORMATIVE COLLECTOR.



## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ .....	12
ВСТУП .....	13
РОЗДІЛ 1. ІНФОРМАЦІЯ. ВИМІРЮВАННЯ ІНФОРМАЦІЇ. ІНФОРМАЦІЙНІ РЕСУРСИ.....	16
1.1 Дані, інформація, знання .....	16
1.2 Інформаційні ресурси .....	17
1.3 Вимірювання інформації .....	18
1.3.1 Алфавітний підхід .....	18
1.3.2 Статистичний підхід .....	20
1.3.3 Семантичний підхід .....	23
1.4 Порівняльний аналіз підходів до вимірювання інформації.....	25
1.5 Об’єкт дослідження. Предмет дослідження .....	26
ВИСНОВКИ ДО РОЗДІЛУ 1 .....	28
РОЗДІЛ 2. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ОСНОВНИХ ТИПІВ ІНФОРМАЦІЙНИХ РЕСУРСІВ .....	30
2.1 Класифікація .....	30
2.2 Типи інформаційних ресурсів.....	31
2.2.1 Файлові системи.....	32
2.2.2 Бази даних .....	34
2.2.3 Інформаційні сховища .....	35
2.2.4 Інформаційні колектори .....	35
2.2.5 Веб-ресурси.....	36
2.2.6 Порівняння основних типів інформаційних ресурсів. ....	36

2.3 Організація пошуку інформації в інформаційних ресурсах .....	38
2.3.1 Структура і класифікація пошукових систем.....	39
2.3.2 Оцінка релевантності результатів пошуку .....	42
2.3.3 Додаткові механізми пошукових систем.....	48
2.4 Інтеграція та інтероперабельність інформаційних ресурсів.....	49
ВИСНОВКИ ДО РОЗДІЛУ 2 .....	53
РОЗДІЛ 3.    ПРОДУКУВАННЯ    ІНФОРМАЦІЙНИХ    РЕСУРСІВ ІНФОРМАЦІЙНО - КОМУНІКАЦІЙНИХ СИСТЕМ.....	54
3.1 Ручні засоби .....	54
3.2 Автоматизовані засоби .....	56
3.2.1 Побудова баз даних.....	56
3.2.2 Побудова інформаційних сховищ .....	57
3.2.3 Побудова інформаційних колекторів.....	58
3.2.4 Побудова веб-ресурсів.....	67
ВИСНОВКИ ДО РОЗДІЛУ 3 .....	68
РОЗДІЛ 4.    ЕТАПИ    ТА    ПРАКТИЧНЕ    ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ ПРОДУКУВАННЯ ІНФОРМАЦІЙНИХ РЕСУРСІВ.....	69
4.1 Етапи продукування інформаційного ресурсу .....	69
4.2 Застосування технології продукування інформаційного ресурсу для Державної податкової адміністрації України.....	75
ВИСНОВКИ ДО РОЗДІЛУ 4 .....	79
РОЗДІЛ 5.    ТЕХНІКО-ЕКОНОМІЧНЕ    ОБГРУНТУВАННЯ ПРОЕКТУ. ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ .....	80
5.1 Постановка задачі техніко-економічного дослідження .....	80
5.2 Аналіз виявлених варіантів ПП .....	81

5.2.1	Формування функцій ПП .....	81
5.2.2	Варіанти реалізації функцій .....	81
5.2.3	Морфологічна карта .....	83
5.3	Обґрунтування системи параметрів .....	85
5.3.1	Опис параметрів .....	85
5.3.2	Кількісна оцінка параметрів .....	85
5.3.3	Визначення коефіцієнтів вагомості параметрів .....	87
5.3.4	Перевірка ступеня придатності експертних оцінок .....	88
5.3.5	Результати розрахунку попарного пріоритету показників .....	89
5.3.6	Розрахунок пріоритету параметрів .....	90
5.3.7	Визначення рівня якості .....	91
5.4	Економічний аналіз варіантів ПП .....	92
5.4.1	Визначення трудомісткості розробки програмного продукту .....	92
5.4.2	Визначення витрат на розробку і розрахунок вартості ПП .....	94
5.4.3	Розрахунок показників економічної ефективності кожного з варіантів програмного продукту .....	97
5.5	ВИСНОВКИ ДО РОЗДІЛУ 5 .....	98
	ВИСНОВКИ .....	99
	ПЕРЕЛІК ПОСИЛАНЬ .....	100
	ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ .....	106
	ДОДАТОК Б. ФРАГМЕНТИ ПРОГРАМНОГО КОДУ .....	120

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

ІКС	Інформаційно-комунікаційна система
ІКТ	Інформаційно-комунікаційні технології
Орг.С.	Організаційна система
БД	База даних
СУБД	Система управління базою даних
UML	Unified Modeling Language, Уніфікована мова моделювання
XML	Extensible markup language, розширюваний формат розмітки
ФС	Файлова система
ПЗ	Програмне забезпечення
ПП	Програмний продукт
ФВА	Функціонально-вартісний аналіз
PR	Google PageRank
ДПА	Державна податкова адміністрація України
OAI	Open Archives Initiative, Ініціатива “Відкриті архіви”
ETL	Extract, Transform, Load, засіб інтеграції через витягання, перетворення та завантаження даних
SEO	Search engine optimization, оптимізація для пошукових систем
OCLC	Online Computer Library Center, мережевий центр комп’ютерних бібліотек
API	Application programming interface, програмний інтерфейс взаємодії
MARC	Machine readable cataloging, машинно-орієнтовані каталоги

## ВСТУП

### **Оцінка сучасного стану проблеми системного продукування інформаційних ресурсів ІКС Орг.С.**

На сучасному етапі розвитку цивілізації однією з найбільших цінностей стає інформація. Управління потоками інформації без застосування спеціальних технічних і програмних засобів стає неможливим. Тому виникає проблема створення таких засобів.

Під терміном “організаційна система” ми розуміємо об’єднання автономних підприємств, організацій або структурних підрозділів з економічної, соціальної або державної сфери діяльності, які функціонують під централізованим керівництвом і вирішують спільні завдання [1]. Сьогодні одним із ефективних засобів підвищення конкурентоздатності організаційної системи є її інформатизація. Під визначенням “Інформатизація організаційних систем” ми розуміємо необхідну і достатню множину правових, організаційних, економічних, наукових та науково-технічних рішень і процесів, спрямованих на створення інформаційно-комунікаційних систем з метою задоволення інформаційних потреб, забезпечення та автоматизації бізнес-процесів, підтримки прийняття рішень та підвищення ефективності управління організаційною системою із застосуванням інформаційно-комунікаційних технологій [2, 3, 4]. Це стосується всіх напрямків інформатизації всіма засобами інформатизації, в тому числі засобами інформаційно-комунікаційних технологій [5].

Створення інформаційно-комунікаційної системи передбачає створення інформаційного ресурсу як одного із її компонентів. Він забезпечує зберігання та обробку даних, інформації та знань, які сьогодні є однією з основних цінностей Орг.С. [6]. Для того, щоб створити ресурс, який буде максимально ефективно задовольняти потреби організації, необхідно провести ґрунтовний аналіз як згаданих потреб, так і характеристик та особливостей інформації.

Основою для проектування ІКС є фундаментальні положення системного аналізу та теорії систем [1, 7, 8].

### **Актуальність проблеми**

Конкурентоспроможність будь-якої організаційної структури, підприємства, фірми прямо залежить від кількості та якості інформації, якою вона володіє, а також від наявності засобів оперування такою інформацією. Тому проблема створення інформаційних ресурсів є важливою для конкурентоспроможності, а отже у сучасних умовах і існування організаційної системи.

### **Мета роботи, галузь застосування результатів**

**Метою роботи** є дослідження методів та розробка технології продукування інформаційних ресурсів. Необхідно запропонувати підхід, що дозволить обґрунтовано обрати засоби та технології для створення інформаційних ресурсів.

Для цього необхідно проаналізувати дані, інформація та знання, які можуть зберігатись в інформаційних ресурсах.

На основі компонентного підходу та прикладного системного аналізу необхідно розробити підхід до проектування та створення ресурсу як системи компонент, певним чином пов'язаних між собою та з іншими компонентами ІКС.

Мають бути досліджені інформаційні ресурси, їх класифікація, основні типи. Результати аналізу засобів продукування інформаційних ресурсів є основою для формулювання рекомендацій по застосуванню тих чи інших засобів для продукування окремих компонентів ресурсу ІКС.

Результати роботи спрямовані на застосування при розробці проектів інформатизації організаційних систем, зокрема для системного проектування інформаційних ресурсів.

### **Взаємозв'язок з іншими науковими роботами**

Розробка інформаційного ресурсу є одним з етапів розробки інформаційно-комунікаційної системи організаційної системи [9]. Підходи до дослідження процесів інформатизації організаційних структур описані в [2, 3].

Проектування інформаційного ресурсу базується на системному проектуванні інформаційних систем [10]. Принципи моделювання організаційної системи описані в [11, 12].

Для ефективного використання інформаційного ресурсу мають бути розроблені засоби доступу до нього. На сьогодні для доступу до ресурсів організаційних систем використовуються корпоративні портали [13]. При проектуванні ресурсу має бути передбачена інтеграція із засобами доступу, організація належного рівня безпеки [14], інтеграція, тощо.

## РОЗДІЛ 1. ІНФОРМАЦІЯ. ВИМІРЮВАННЯ ІНФОРМАЦІЇ. ІНФОРМАЦІЙНІ РЕСУРСИ

### 1.1 Дані, інформація, знання

Однозначного та завжди коректного визначення понять дані, інформація, знання, нажаль, немає. Сам термін “інформація” є порівняно молодим: до II половини XX сторіччя в російській та українській мовах поняття “інформація” не вживалось [9]. У перекладі з латини *informatio* – відомості, роз’яснення. Можна навести багато визначень інформації, від філософського “Відображене різноманіття, яке виникає в результаті взаємодії між об’єктами” [15] до визначення Лебедєва – “послідовність нулів та одиниць”. Однак перше визначення не є конструктивним а друге не зовсім відповідає сучасному розумінню інформації.

Згідно законодавства України інформація – відомості, подані у вигляді сигналів, знаків, звуків, рухомих або нерухомих зображень чи в інший спосіб [16]. Законодавство України визначає дані як інформація у формі, придатній для автоматизованої обробки її засобами обчислювальної техніки [16].

За європейськими стандартами, знання – це комбінація даних та інформації, до яких додається точка зору, навички та досвід експерта, що дає вагомий результат, який може бути використано для прийняття рішень. Знання може бути вичерпним та/або вузьким, індивідуальним та/або колективним. Нажаль, українське законодавство взагалі не визначає поняття “знання”.

Згідно інших джерел, “Data is information before it has been given any context, structure and meaning” [17] – Дані це інформація до того, як їм надається контекст, структура та значення.

Найбільш вдалим, на мою думку, є визначення, за яким дані – це результат простого збору визначених фактів; інформацією вони стають лише



при зв'язуванні у щось корисне, комбінацію хто, що, де і як. У свою чергу знання – це розуміння, як і чому щось відбувається [18]. Саме такі значення понять “дані”, “інформація” та “знання” будуть використовуватись далі.

За ступенем упорядкованості інформація поділяється на структуровану і неструктуровану. Відповідно до цього системи можна розділити на [19]:

1. системи для роботи зі структурованими даними для роботи з неструктурованими даними (системи управління знаннями)

Більша частина інформації, з якою працює організація (близько 80%) є неструктурованою.

Системи управління знаннями в свою чергу можна поділити [19]:

По об'єму функціональності:

1. Інформаційні
2. Пошукові
3. Пошуково-аналітичні
4. Повнофункціональні

По типу використання:

1. Сервіси
2. Відчужуємі програми

## 1.2 Інформаційні ресурси

Дані, інформація та знання – сутності, які можуть відображати об'єкти, процеси та явища навколишнього середовища, вони є абстрактними об'єктам [9]. Засобом їх матеріалізації є інформаційний ресурс. Згідно законодавства України інформаційний ресурс – сукупність документів у інформаційних системах (бібліотеках, архівах, банках даних тощо) [20].

Документ – це упорядкована сукупність даних, інформації та знань, яка надає можливості доступу, передачі, обробки, тощо. Прикладом документа може бути паперовий документ, фільм, комп'ютерний файл, тощо [9].

Середовищем зберігання документів є інформаційно-комунікаційна система, яка забезпечує доступ, обмін інформацією та її обробку. Така система

не обов'язково повинна бути комп'ютеризована. Прикладами комп'ютеризованих інформаційно-комунікаційних систем можуть бути, операційне середовище, база даних, інформаційне сховище, інформаційний колектор.

Продуктування інформаційних ресурсів є одним з основних завдань інформаційного суспільства [21]. Під продуктуванням інформаційних ресурсів ми розуміємо створення нових видів інформаційних ресурсів на основі існуючих даних, інформації, знань.

Для ефективного продуктування інформаційних ресурсів необхідно провести аналіз характеристик даних, інформацій та знань, які будуть у ньому зберігатись [22].

### 1.3 Вимірювання інформації

#### 1.3.1 Алфавітний підхід

Алфавітний підхід до вимірювання інформації оцінює розмір інформації як суму структурних елементів повідомлення. Найчастіше такий підхід застосовується при передачі даних. Це пов'язане з тим, що саме така оцінка дозволяє визначити час передачі повідомлення або необхідні параметри каналу для досягнення визначених його характеристик.

Традиційна і найпростіша міра у рамках такого підходу – геометрична. Одиниця виміру – інформаційний елемент. Міра може бути використана для визначення інформаційної ємності пам'яті комп'ютера. У даному разі в якості інформаційного елементу виступає мінімальна одиниця збереження – біт [23].

Геометрична міра також може бути застосована для визначення ефективності алгоритму кодування. Повідомлення 55556666888888 має розмір 14 байт (112 біт). Якщо закодувати його одним з ефективних методів – методом повторів – отримаємо повідомлення 5(4)6(4)8(6), розмір якого 12. Можна представити те ж саме повідомлення іншим чином: 454668 (якщо обмежити максимальну кількість символів, що повторюються). Розмір такого

повідомлення 6. Ефективність методу кодування визначається середнім відношенням довжини вихідного повідомлення і закодованого. Однак дана міра не враховує того, що для запису повідомлення можуть бути використані різні набори символів [23].

Удосконаленням геометричної міри є адитивна міра, запропонована Хартлі у 1928 році (міра Хартлі). Хартлі вперше ввів спеціальне позначення для кількості інформації –  $I$  і запропонував наступну логарифмічну залежність між кількістю інформації та потужністю вихідного алфавіту:

$$I = L \log h, \quad (1.1)$$

де  $I$  – кількість інформації, що міститься у повідомленні;  $L$  – довжина повідомлення;  $h$  – потужність вихідного алфавіту.

При вихідному алфавіті  $\{0,1\}$ ;  $L = 1$ ;  $h = 2$  і основі логарифму, рівному 2, маємо

$$I = 1 * \log_2 2 = 1 \quad (1.2)$$

Формула 1.2 дає аналітичне визначення біту (BIT - BInary digiT) за Хартлі: це кількість інформації, яка міститься в одній бінарній цифрі. Одиницею виміру в адитивній мірі також є біт [24].

Комбінаторна міра оцінює можливість представлення інформації за допомогою різних комбінацій інформаційних елементів в заданому об'ємі. Використовуються типи комбінацій елементів і відповідні математичні співвідношення, які приводяться в комбінаториці.

Комбінаторна міра може бути використана для оцінки інформаційних можливостей деякого автомату, який може генерувати дискретні сигнали (повідомлення) у відповідності до певного правила. Комбінаторна міра використовується для визначення можливостей кодуєчих систем, які широко використовуються в інформаційній техніці [23].

### 1.3.2 Статистичний підхід

Найбільш відомим і широко вживаним на практиці підходом є імовірнісний підхід до вимірювання інформації. На основі даного підходу розроблено великий розділ кількісної теорії інформації, яка за ім'ям її основоположника називається “теорія інформації Шеннона”. Головною відмінною особливістю імовірнісного підходу від комбінаторного є той факт, що він базується на імовірнісних припущеннях відносно знаходження певної системи в різних станах. При цьому загальне число елементів (мікростанів, подій) системи не враховується. За кількість інформації тут приймається знята невизначеність вибору із множини можливостей, які мають, в загальному випадку, різну ймовірність [25].

Нехай маємо  $n$  подій із ймовірностями  $p_1, p_2, \dots, p_n$  і необхідно визначити кількість інформації у повідомленні за імовірнісною мірою (позначимо її  $H$ ). Для такої міри  $H$  Шеннон висуває наступні вимоги:

1.  $H$  повинна бути неперервною відносно  $p_i$ .
2. Якщо всі  $p_i$  рівні, то  $H$  повинна бути монотонно зростаючою від  $n$ .
3. Якщо вибір розпадається на два послідовних вибора, то первісна  $H$  повинна бути сумою індивідуальних значень  $H$  кожного з виборів.

В процесі подальших досліджень Шеннон довів теорему, згідно якої існує єдина функція, що задовольняє всім трьом вимогам, і вона має вигляд:

$$H = -K \sum_{i=1}^n p_i \log p_i \quad (1.4)$$

де  $K$  – певна додатня стала [26].

Очевидно, що при рівній імовірності  $p_i$  міра Шеннона зводиться до міри Хартлі.

Удосконаленням даного підходу є міра, запропонована у синергетичній теорії інформації. Так Вяткин В.Б. вводить поняття “негентропія”, як сутність, протилежна ентропії [25]. Даний підхід є логічним продовженням ідей

Бріллюена, згідно якого інформація являє собою від'ємний вклад у ентропію [27]. Математично величина негентропії визначається за двома різними мірами. Перша міра визначає негентропію як різницю між ентропією до і після отримання повідомлення. Друга міра визначає негентропію як відношення між цими значеннями [25].

Друга міра очевидно є кращою. Це пов'язано з тим, що будь-яке повідомлення зменшує ентропію саме в певну кількість разів а не на абсолютну величину. Таким чином при появі даного повідомлення першим або другим в системі повідомлень абсолютна величина зміни ентропії буде різною, а відносна – сталою (за умови що повідомлення незалежні).

Відмінний від поглядів Хартлі, Шеннона, Вінера і Бріллюена підхід до визначення поняття “кількість інформації” запропонував академік А.Н. Колмогоров, який він назвав алгоритмічним.

Виходячи з того, що найбільш ємним є представлення о кількості інформації “в чомусь” (X) або “про щось” (Y), Колмогоров для оцінки інформації в одному скінченому об'єкті відносно іншого запропонував використовувати теорію алгоритмів. За кількість інформації в одному скінченому об'єкті відносно іншого скінченого об'єкту приймається значення певної функції від складності кожного з об'єктів і довжини програми (алгоритму) перетворення одного об'єкту в інший [28].

Розв'язання задачі визначення кількості інформації в алгоритмічному підході має загальний вигляд і описується наступним чином.

“Відносною складністю” об'єкта Y при заданому X будемо вважати мінімальну довжину “програми” P отримання Y з X. Сформульоване так визначення залежить від “методу програмування”. Метод програмування є не що інше, як функція, яка ставить у відповідність програмі P і об'єкту X об'єкт Y” [28].

Так як кожний з об'єктів не може бути нескінченно складним, то можна довести теорему, згідно якої відносній складності об'єкту Y, при заданому методі програмування, може бути поставлена у відповідність інша відносна

складність, яка отримана при іншому методі програмування, така, що виконується нерівність:

$$K_A(Y|X) \leq K_\varphi(Y|X) + C_\varphi \quad (1.3)$$

де  $C_\varphi$  - певна стала програмування, яка не залежить від  $X$  і  $Y$ .

Враховуючи, що при довільних  $X$  і  $Y$  відносна складність  $K_A(Y|X)$  є скінченою величиною, а  $K_A(Y) = K_A(Y|I)$  можна вважати просто складністю об'єкту  $Y$ , А.Н. Колмогоров для оцінки алгоритмічної кількості інформації в об'єкті  $X$  відносно об'єкту  $Y$  запропонував використовувати формулу:

$$I_A(X:Y) = K_A(Y) - K_A(Y|X) \quad (1.4)$$

при чому  $K_A(X|X) \approx 0$  і, відповідно,  $I_A(X|X) \approx K_A(X)$  [28].

Алгоритмічна інформація найбільш близька до визначення негентропії відображення системних об'єктів в порівнянні з розглянутими раніше інформаційними мірами і навіть, на принциповому рівні суджень, може бути прийнята за її кількісну характеристику. Однак вона не може бути застосована до системних об'єктів. Справа в тому, що коли відображуємий об'єкт є відкритим, ми будемо мати від'ємні значення, негативізм яких може бути доповнений тим, що, як не важко помітити, можливі ситуації, в яких адитивна негентропія відображення сукупності відображуючи об'єктів буде рівна нулю. Таким чином, наприклад, після проведення спостереження і виявлення сукупності об'єктів, які мають безпосередній взаємозв'язок з відображуємим (досліджуємим) об'єктом, в результаті будемо мати, що загальна кількість інформації, яку ми отримали в процесі досліджень, рівна нулю, а це вже нонсенс. Таким чином, ми бачимо, що алгоритмічний підхід, так же як комбінаторний і імовірнісний, не дозволяє отримати розрахункову формулу для негентропії відображення системних об'єктів [25].

### 1.3.3 Семантичний підхід

Алфавітний та статистичний підхід дуже важко застосувати до вимірювання знань. Обмеження алфавітного підходу в даному випадку очевидні. Статистичний підхід оперує такими поняттями, як імовірність тої чи іншої події, таким чином вимагаючи наявності певних очікуваних результатів досліду. Коли мова йде про знання, дуже часто воно стосується речей, ймовірність яких оцінити апріорно не можна. Тим більше якщо говорити про певні загальні істини та загальні закони. Крім того інколи і для інформації визначення ймовірностей є дуже складним. Спробою запропонувати методи вимірювання таких знань і інформації є семантичний підхід.

Семантичний підхід враховує доцільність та корисність інформації. Застосовується для оцінки ефективності інформації, яка одержується, та її відповідності до реальності.

В рамках даного підходу розглядаються такі міри, як доцільність, корисність, (враховують прагматику інформації) і істинність інформації (враховує семантику інформації).

Кількість інформації  $I$  з позицій її доцільності визначається формулою:

$$I = \log \frac{p_1}{p_2} \quad (1.5)$$

де  $p_1, p_2$  – імовірності досягнення цілі після і до отримання повідомлення, відповідно [23].

Така міра дозволяє, з одного боку, виміряти знання і інформацію. З іншого боку вона, так як і статистичні міри, зводиться до оцінки ймовірностей. Тут стає очевидною характерна особливість семантичного підходу – визначена величина кількості інформації є суб'єктивною. Дійсно, ймовірності досягнення цілі залежать не лише від інформації у даному повідомленні, але в дуже великій мірі від самого суб'єкта, що сприймає повідомлення, від інших його знань, умінь навичок. Це не можна вважати недоліком такої міри, так як

дійсно, інформація, відома всім, навряд-чи має якусь цінність і багаторазове її відтворення не є корисним.

Як вже було вказано, кількість інформації  $I_{\text{засв}}$ , яку “засвоює” користувач, тісно пов’язана із тими знаннями, які користувач має до моменту отримання інформації – з тезаурусом (ТЗ) користувача. На цьому заснована міра корисності інформації. Наприклад, для засвоєння тих знань, які студент отримує у ВНЗ, необхідна середня освіта, інакше студент ці знання не зрозуміє і тому не засвоїть. Тому кожна дисципліна базується на знаннях, які студент повинен отримати на попередніх курсах. Цим пояснюється послідовність учбових дисциплін по рокам навчання.

Залежність інформації, що засвоюється, від тезаурусу, демонструє крива, зображена на рис 1.1.

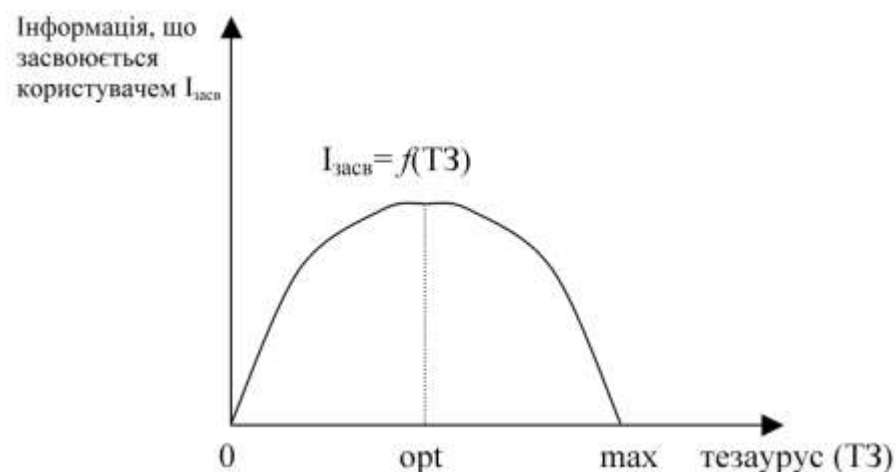


Рисунок 1.1 – Крива засвоєння інформації в залежності від тезаурусу

Очевидно, що при тезаурусі, рівному нулю, інформація не засвоюється, так як є незрозумілою для користувача. З іншого боку при максимальному тезаурусі інформація також не засвоюється, так як вона вже відома користувачу. Максимально засвоюється інформація (тобто, вона максимально корисна) в точці opt, коли користувач має достатній (але не максимально можливий) тезаурус для розуміння інформації, яку він отримує. При значенні тезаурусу  $i$ -го користувача  $\text{ТЗ}_i$  кількість інформації, яку він засвоює,



визначається як  $I_{засв} = f(TЗ_i)$ . Сам тезаурус може бути визначений як результат інтелектуального тестування, яке проводиться, наприклад, перед поданням інформації. При такому тестуванні людині виставляється певний бал, який може розцінюватись як його тезаурус [23].

Міра істинності інформації оцінює інформацію з позицій її відповідності джерелу інформації, тобто реальному світу. Оцінка може бути проведена по шкалі [0, 1]. Тоді для оцінки складного повідомлення (частково істинного) його істинність можна прийняти як функцію від істинності простих повідомлень (які в свою чергу можуть бути істинними 1 чи хибними 0). При цьому функція визначає “вимогливість” щодо істини: середнє арифметичне забезпечить лояльне відношення до неправди, мінімум – визначить що навіть частково хибне повідомлення є все-ж таки хибним [23].

#### 1.4 Порівняльний аналіз підходів до вимірювання інформації

Можливості застосування розглянутих підходів для різних типів ресурсів наведено у табл. 1.1 [22].

Таблиця 1.1 – Порівняльний аналіз підходів до вимірювання інформації

Тип ресурсу	Призначення	Алфавітний підхід	Статистичний підхід	Семантичний підхід
Файлова система	Персональний ресурс	Автоматично	Не застосовується	Не застосовується
База даних	Зберігання упорядкованих масивів даних	Автоматично	Може бути застосований, можливо автоматизований	Вимагає експертного оцінювання
Інформаційне сховище	Зберігання та аналітична обробка інформації	Автоматично	Може бути застосований, можливо автоматизований	Вимагає експертного оцінювання
Інформаційний колектор	Зберігання знань	Автоматично	Не застосовується	Вимагає експертного оцінювання

Алфавітний підхід використовується для всіх типів інформаційних ресурсів та виконується засобами ресурсу автоматично. Він дозволяє адекватно оцінити апаратні засоби, необхідні для роботи з ресурсом, а також висловити певні вимоги до програмних засобів. Однак така оцінка не може адекватно визначити цінність ресурсу.

Статистичний підхід вимагає визначення параметрів оцінюваної інформації, таких як, наприклад, імовірність тих чи інших подій. Тому автоматичне вимірювання поки що неможливе. Для файлових систем, враховуючи різні типи і завдання даних що зберігаються, така оцінка є неактуальною. Для інформаційних колекторів, які призначені для зберігання знань, статистичні параметри практично не можуть бути визначені, тому така оцінка також не застосовується.

Семантичний підхід може бути використаний для вимірювання інформації та знань. Він не відповідає характеру інформації, яка зберігається у файлових системах. Для баз даних такий підхід може бути використаний із залученням експертів. Наприклад, оцінка істинності даних у базі дозволяє визначити її цінність. Для інформаційних сховищ і колекторів такий підхід також може бути застосований. Необхідність залучення до оцінювання людини викликана, передусім, тим, що комп'ютер (штучний інтелект) поки що не може зрозуміти суть інформації для оцінювання.

### 1.5 Об'єкт дослідження. Предмет дослідження

Виходячи з проведеного аналізу визначимо **об'єкт дослідження** – інформаційні ресурси інформаційних систем як компонента інформаційно-комунікаційної системи Орг.С.

Термін “організаційна система” означає об'єднання автономних підприємств, організацій або структурних підрозділів з економічної, соціальної або державної сфери діяльності, які функціонують під централізованим керівництвом і вирішують спільні завдання [10].

**Предмет дослідження** – бізнес-процеси системної інженерії, аналізу, проектування, конструювання та розгортання інформаційних ресурсів ІКС.

## ВИСНОВКИ ДО РОЗДІЛУ 1

Кожен з розглянутих підходів до вимірювання кількості інформації може бути застосовано до певних процесів обробки інформації. Так, алфавітний підхід найбільш адекватно може бути використаний при розробці систем збереження і передачі повідомлень. Алфавітний підхід можна найбільш ефективно застосовувати для вимірювання даних. Для вимірювання інформації і знань він практично неприйнятний враховуючи вказані його недоліки [22].

Статистичний підхід дозволяє частково розв'язати проблеми алфавітного підходу. Його можна застосувати для вимірювання інформації, але він ще недостатній для вимірювання знань. Суттєвим недоліком у порівнянні з алгоритмічним підходом є неможливість проведення вимірювання без втручання людини, тому що він базується на ймовірностях, які машина поки що не може оцінити навіть приблизно.

Семантичний підхід формує основні методи, які можуть бути застосовані для оцінки знань, хоча з успіхом може бути застосований і для оцінки кількості інформації. Однак, на відміну від попередніх підходів, отримана в результаті величина сильно залежить від суб'єктивних особливостей людини, яка дану інформацію (знання) сприймає. Таким чином дана оцінка не може бути використана для визначення абсолютної “об'єктивної” кількості інформації. З іншого боку дана проблема пов'язана саме із суб'єктивним характером знань, що впливає із визначення поняття “знання”.

Однак це не означає, що при сучасному рівні продуктивності комп'ютерної техніки слід обмежитись вимірюванням лише даних і інформації як таких, для яких може бути запропонована певна кількісна оцінка [22].

З іншого боку, всебічне поширення ресурсів, які зберігають саме інформацію і знання (таких як інформаційні сховища, колектори, бібліотеки), існує необхідність у створенні методів визначення принаймні приблизної кількості знань у документі. При цьому важливо створювати саме

автоматизовані засоби вимірювання, тому що людина фізично вже не має змоги розібратись в тих об'ємах інформації, які постійно створюються сьогодні.

Таким чином вдосконалення підходів та розробка технологій автоматизації вимірювання інформації в інформаційних ресурсах інформаційно-комунікаційних систем є актуальною і важливою для розвитку інформаційно-комунікаційних технологій [22].

## РОЗДІЛ 2. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ОСНОВНИХ ТИПІВ ІНФОРМАЦІЙНИХ РЕСУРСІВ

### 2.1 Класифікація

Класифікація інформаційних ресурсів дозволяє обґрунтовано визначити необхідний тип інформаційного ресурсу, обрати інструменти та технології для його продукування.

Інформаційні ресурси можна класифікувати за такими ознаками [29].

1. Приналежність ресурсу до певної організаційно-технологічної системи (наприклад, бібліотечної мережі, ЗМІ, корпоративної системи).
2. Спосіб виділення об'єктів обліку (твори, документи, видання, бази даних, інтернет-сторінки, сайти, тощо).
3. Призначення ресурсу (масова інформація, освіта, бізнес, особиста переписка, тощо).
4. Зміст ресурсу:
  - тематичний;
  - об'єктний;
  - функціональний.
5. Видовий склад ресурсу (види документів).
6. Джерело інформації:
  - національне або закордонне;
  - офіційне або неофіційне.
7. Правовий статус ресурсу (публічні документи, об'єкти інтелектуальної власності, спам, таємні документи, тощо).
8. Структурний тип ресурсу, що включає:
  - можливість відділення даних від програм та представлення;
  - формати;

- кодування;
  - інше.
9. Відкритість ресурсу (відкритий або з обмеженим доступом).
10. Рівень структурованості:
- структуровані;
  - неструктуровані.
11. Спосіб розповсюдження і носій.
12. Мова ресурсу.

Для опису вимог до інформаційного ресурсу, що розробляється, ми розглядаємо ряд його характеристик [30]. Класифікація дозволяє розбити ресурс на компоненти, визначити характеристики, обрати оптимальні засоби для їх реалізації.

1. Характеристики продуктивності:
  - пропускна спроможність;
  - час реакції;
  - час затримки.
2. Характеристики надійності.
3. Характеристики масштабованості.
4. Характеристики розширюваності.
5. Характеристики прозорості.
6. Повна вартість володіння.

Повна вартість володіння інформаційного ресурсу визначається не тільки (і не стільки) вартістю використаних при його створенні апаратних і програмних засобів, а і вартістю інформації, яка в нього закладена.

## 2.2 Типи інформаційних ресурсів

Комп'ютеризовані інформаційні діляться на такі основні типи [30]:

1. файлові системи;
2. бази даних;
3. інформаційні сховища;

4. інформаційні колектори;
5. веб-ресурси.

### 2.2.1 Файлові системи

Файлові системи (ФС) є найпростішим і найбільш розповсюдженим типом інформаційних ресурсів. Вони дозволяють зберігати дані, інформацію та знання довільного типу та довільної структури. ФС з одного боку є персональним інформаційним ресурсом, а з іншого є базою для інших типів ресурсів.

Розвиток ФС почався із створенням носіїв даних достатньо великого розміру, та виникнення задач, які вже не дозволяли зберігати на носії дані одного типу у визначеному форматі. Можна виділити такі типи ФС [31, 32]:

1. Дисккові. Орієнтовані на зберігання розрізнених даних на одному жорсткому диску персонального комп'ютера користувача. Здебільшого використовуються на ПК для вирішення персональних задач користувача.
2. Кластерні. Орієнтовані на зберігання даних на масиві дисків, що забезпечує вищу швидкість, надійність, ємність. Як правило використовуються в серверах, хоча все частіше застосовуються в ПК (RAID-масиви).
3. Мережеві. Дозволяють пов'язати в єдину ФС носії на різному апаратному забезпеченні, поєднані в межах мережі. Розвиток таких систем дозволяє створити інформаційний ресурс підприємства із забезпеченням максимальної надійності і швидкості роботи при уникненні дублювання.
4. Спеціального призначення. До цієї групи відносяться ФС, оптимізовані для роботи на різноманітних носіях (таких як компакт-диски), системи доступу до ftp, системи файлових пристроїв, тощо.



Ми розглядаємо два основні класи задач, які вирішує ФС: організація розміщення даних на фізичному носії та організація файлового простору користувача (ієрархії файлів та каталогів).

Фізична організація зберігання даних на диску передбачає, що для користувача (програми) створюється інтерфейс за певними правилами, тобто певний образ організації файлів і каталогів. Реальне місце кожного файлу встановлюється ФС за певним алгоритмом. Такий алгоритм повинен забезпечити швидкий доступ до файлів, пошук, запис, операції переміщення/копіювання, тощо. Це завдання перш за все пов'язане із фізичною організацією пристроїв для збереження даних.

Другим аспектом розробки ФС є організація ієрархії файлів і каталогів. Вона має забезпечити структурованість інформації, швидкість та зручність її обробки, підтримку стандартних рішень. Як правило ієрархію файлів кожен користувач визначає самостійно, але інколи існують рішення, орієнтовані на певну визначену ієрархію. Прикладом може бути організація ФС для ОС Windows. Користувач може створювати ієрархію для своїх даних але для ОС використовуються ієрархія, створена розробником. Аналогічно, якщо у межах організаційної структури використовуються певні загальні програмні рішення, може з'явитись необхідність розробки ієрархії ФС. Така розробка має проводитись на етапі проектування, а в подальшому за допомогою організаційних засобів має контролюватись підтримка встановленої ієрархії.

Останнім часом на файлові системи покладаються нові задачі, такі як контроль доступу до даних, підтримка політик безпеки, архівація, тощо. Це викликано тим, що ФС у сучасних системах вже не є лише основою операційної системи (ОС) а являє собою окремий інформаційний ресурс, доступ до якого може проводитись і "в обхід" ОС. Тому функції, які раніше покладались на ОС, тепер мають бути перекладені на саму ФС.

### 2.2.2 Бази даних

База даних (БД) являє собою інформаційний ресурс, призначений для зберігання структурованих даних. Класифікація БД за структурою інформації, що зберігається:

1. Каталоги. Найпростіший вид БД у якому використовується підхід, аналогічний до бібліотечних каталогів – вся інформація зберігається у одній таблиці з визначеним набором полів. Такі БД були історично першими і почали розвиватись у 60-х роках минулого сторіччя.
2. Ієрархічні. У таких БД дані організовані у деревовидну структуру, у якій кожен запис є вузлом. Така організація БД забезпечує швидкий пошук даних (лише за одним критерієм).
3. Мережеві. Такі БД зберігають дані із зв'язками з іншими даними. Зв'язки організуються як вказівники (pointer) на інші записи. Як правило для зберігання записів організовується схожа до ієрархічної схема.
4. Реляційні. Найпопулярніші сучасні БД, які базуються на реляційній моделі, запропонованій Е. Коддом у 1969 р. [33]. У таких БД дані представлені у вигляді таблиць, кожен рядок яких представляє один запис а кожний стовпчик – визначений атрибут. Робота з БД відбувається за допомогою запитів, результатом яких є таблиці.
5. Пост-реляційні. Такі бази використовують деякі принципи реляційних БД але не обмежуються реляційною моделлю. Часто вони використовують принципи ієрархічних та мережевих БД.
6. Об'єктні. Враховуючи розвиток об'єктно-орієнтованого програмування виконуються спроби створити схему БД, яка б дозволила працювати з об'єктами. В такому випадку БД і програма, яка її використовує, мають єдині імена і це дозволяє спростити взаємодію і уникнути проблем із перетвореннями даних.

Варто зазначити, що при розробці сучасних систем інколи доречно використовувати БД із змішаною структурою [34].

Класифікація баз даних за характером інформації, яка зберігається:

- фактографічні (картотеки),
- документальні (архіви);

Класифікація баз даних за способом зберігання даних:

- централізовані (зберігаються на одному комп'ютері,
- розподілені (використовуються в локальних і глобальних комп'ютерних мережах);

### 2.2.3 Інформаційні сховища

**Інформаційні сховища** являють собою ресурси, призначені для аналітичної обробки інформації. За визначенням Білла Інмона, сховище даних – це предметно-орієнтований, хронологічний, стабільний набір даних, призначений для підтримки прийняття управлінських рішень [35]. З іншого боку, інформаційне сховище являє собою ієрархічно організовану сукупність БД, призначених для збереження та обробки архівних даних.

Бази даних дозволяють ефективно працювати із оперативними даними, але для аналітичних запитів час обробки може бути великим, так як доводиться перероблювати великі об'єми інформації. Натомість у інформаційне сховище додаються оброблені певним чином дані, тобто первинна аналітична обробка проводиться постійно і поступово. Таким чином отримується багатовимірна структура, яка не містить оперативних даних але містить сировину для аналітичної роботи [36]. За запитом експерта сховище дозволяє швидко побудувати так званий “зріз”, який містить необхідну інформацію.

### 2.2.4 Інформаційні колектори

На сьогодні більшості організаційних структур вже недостатньо обробки лише даних та інформації, необхідною є робота зі знаннями. Для зберігання та обробки знань використовуються інформаційні колектори [37].

Першими інформаційними колекторами були бібліотечні системи. Завданням таких систем було зберігання текстів документів у сховищах бібліотек [21]. Документи були класифіковані за стандартними бібліотечними класифікаторами. В сучасних інформаційних колекторах для пошуку документів використовуються метадані [38]. Це дозволяє більш ефективно та швидко знаходити документи.

У сучасних умовах дуже важливим стало завдання інтеграції колекторів. Це пов'язано з характером знань, що зберігаються у колекторах [37]. Більшість колекторів зберігає саме відкриту інформацію. Створення єдиної мережі колекторів [39, 40] дозволить забезпечити можливості пошуку інформації у різних колекторах без необхідності проводити пошук у кожному окремо.

На сьогодні інформаційні колектори використовуються для збереження знань у різних формах. Сучасні системи підтримують зберігання не лише текстів, а і відео, аудіо та інших мультимедійних документів.

#### 2.2.5 Веб-ресурси

Веб-ресурси представляють собою ресурси, які зберігаються на різних елементах, розподілених у гетерогенній мережі. Прикладом такого ресурсу є мережа Інтернет. Перевагою даного типу ресурсу є надійність та забезпечення зручного доступу. Недоліком є складність організації управління і пошуку.

#### 2.2.6 Порівняння основних типів інформаційних ресурсів.

Порівняння основних типів інформаційних ресурсів наведено на табл. 2.1.

Таблиця 2.1 – Порівняльна характеристика типів інформаційних ресурсів

Тип ресурсу	Призначення	Функції
Файлова система	Збереження персональних даних, інформації, знань; Основа для інших ресурсів та систем	Базовий тестовий пошук
База даних	Збереження структурованих	Швидкий доступ та

	даних	пошук даних
Сховище даних	Збір та аналітична обробка інформації	Швидкий аналіз інформації, архівних даних
Інформаційний колектор	Зберігання та розповсюдження знань	Пошук за метаданими; Індексція документів

Як видно з аналізу, кожен з розглянутих типів інформаційних ресурсів призначається для вирішення певних завдань. При побудові ІКС Орг.С. необхідно вирішувати різні типи завдань, тому у більшості випадків використовується комбінований ресурс, що поєднує у собі систему окремих взаємодіючих ресурсів. Компонентний підхід дозволяє описати такі ресурси та взаємодію між ними на рівні інтерфейсів. Типова модель організації ресурсів у складі інформаційного ресурсу ІКС представлена на рис. 2.1.

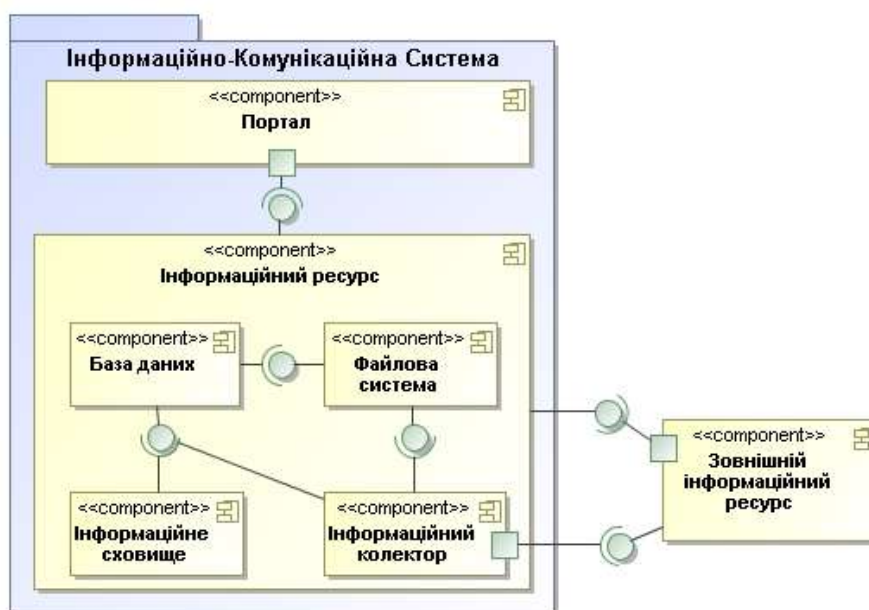


Рисунок 2.1 – Модель організації інформаційного ресурсу ІКС. Діаграма компонентів у нотації UML.

### 2.3 Організація пошуку інформації в інформаційних ресурсах

При розробці ІКС важливою задачею є організація пошуку у інформаційному ресурсі. Можна виділити такі методи пошуку [30]:

1. Повний перебір
2. Індексція
3. Класифікація
4. Тегування

Метод повного перебору є єдиним методом, який можна застосувати для пошуку у неструктурованих інформаційних ресурсах, таких як файлова система. Він полягає в перегляді усіх документів. Недоліком методу є великий час пошуку. Ефективність даного методу в першу чергу залежить від того, наскільки упорядковано зберігає інформацію користувач (наскільки він знає, де слід шукати). Перевагою методу є те, що його застосування не вимагає попередньої обробки інформації. Однак, якщо провести спеціальну попередню обробку, наприклад текстову індексцію, то досягається певна структуризація і час пошуку значно знижується.

Метод індексції полягає в присвоєнні документам (або атрибутам документів) певних індексів. Такі індекси можуть бути впорядковані, що дозволяє швидко знаходити необхідний документ. Даний метод використовується для структурованих інформаційних ресурсів.

Метод пошуку за допомогою класифікації полягає у визначенні певної ієрархічної структури класів документів. Кожен документ відноситься до певного класу. Прикладом може бути універсальний десятковий класифікатор (УДК). Недоліком такого підходу є те, що для пошуку за класифікатором необхідно вміти точно визначити, до якого класу віднести певну інформацію. Крім того, дуже часто інформація може бути віднесена до декількох областей знань. Для вирішення цієї проблеми в системі Yahoo запропоновано наступний підхід – вказівники на підрозділи класифікатора додаються до інших підрозділів. Однак в результаті ми отримуємо хаотичну систему [41].

Тегування полягає в тому, що кожному документу автор ставить у відповідність набір тегів (*ключових слів*), які характеризують його зміст. Недоліком такого підходу є неоднозначність визначення ключових слів. Модифікацією підходу, яку запропонував Ширкі [41] є те, що теги визначають користувачі системи. В такому випадку декілька найбільш часто використаних тегів повинні коректно описувати документ. Однак це не позбавляє тегування його основного недоліку – різні люди схильні використовувати різні терміни для опису одних понять.

Для організації пошуку у окремих типах ресурсів, зокрема у глобальній мережі Інтернет, застосовуються спеціалізовані пошукові машини.

В сучасних умовах важливим завданням при розробці ресурсу стає SEO (Search Engine Optimization) – оптимізація для пошукових систем. Вона передбачає таку організацію і наповнення ресурсу, яка забезпечить високу позицію ресурсу у результатах пошуку за темами, описаними у документах ресурсу.

### 2.3.1 Структура і класифікація пошукових систем

Кожна пошукова система складається з декількох компонентів [42, 43].

1. Spider – павук.
2. Crawler – подорожуючий павук.
3. Індексатор.
4. База даних.
5. Система видачі результатів.
6. Веб-сервер.

Коротко розглянемо кожен компонент та його характеристики.

Spider являє собою браузероподібну програму, яка скачує веб-сторінки для подальшої їх обробки. Crawler знаходить всі посилання на сторінці та визначає подальший шлях для павука. Таким чином, Crawler знаходить нові сторінки, невідомі системі.

Індексатор розбирає сторінку на складові частини та досліджує їх, виділяє основні частини документу, такі як текст, заголовки, структурні особливості, службові теги. Результати розміщуються у базі даних.

Система видачі результатів за запитом користувача вибирає з бази даних ті записи, які найкраще відповідають запиту користувача. Алгоритм. Що покладено у основу такого вибору, є визначальним для пошукової системи, і саме від нього залежить популярність системи.

Веб-сервер забезпечує відображення користувачу сторінки, у якій можна ввести запит. Веб-сервер також відповідає за представлення результатів пошуку у html-форматі.

Враховуючи швидкість змін у Інтернеті та кількість запитів до пошукових систем, постає питання забезпечення необхідної продуктивності. У сучасних пошукових машинах для забезпечення продуктивності використовується розподілена архітектура [44]. Так, кожен із компонентів системи працює на системі окремих серверів. Павукам, краулерам та індексаторам видається різне, максимально незалежне завдання. Результати індексації накопичуються, поєднуються, і у моменти найменшого навантаження записуються у розподілену БД. Зазначимо, що для забезпечення такої паралельності проводиться початкова класифікація за формальними ознаками, яка дозволяє чітко визначити який сервер відповідає за певну інформацію.

Таким чином пошукова система являє собою спеціалізовану базу даних та засоби для її наповнення та вибору даних. Від традиційного інформаційного ресурсу вона відрізняється тим, що зберігає не самі документи а посилання на них і містить алгоритми організації і видачі за запитом таких посилань.

Класифікація систем індексації ресурсів за рівнем функціональності представлено на рис. 2.2 [45].



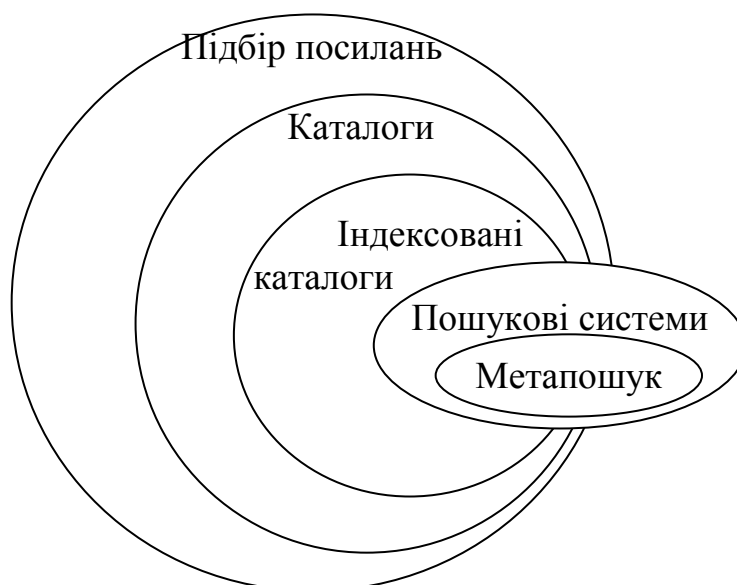


Рисунок 2.2 – Класифікація систем пошуку у інформаційних ресурсах

До систем підбору посилань відносяться системи, що містять лише список посилань на Інтернет-ресурси. Такий список відсортований за формальними ознаками (наприклад, за алфавітом) і класифікований за великими групами. Додання посилань у такі списки проводиться адміністратором ресурсу, як правило, вручну.

Каталоги також містять інформацію про спеціально додані у них ресурси. Однак у них інформація упорядкована по категоріям спеціально розробленого дерева-рубрикатора та наявні механізми пошуку. Як правило, інформація про сайти заноситься самими авторами із зазначенням певних відомостей, за якими і відбувається пошук в подальшому.

Індексовані каталоги відрізняються тим, що містять вдосконалені системи пошуку, які базуються на частковій або повній індексації посилань у спеціалізованій базі даних. У такому випадку користувач може створювати складні запити, які швидко обробляються.

Основною відмінністю пошукових систем є наявність спеціального робота, або павука, який постійно сканує Інтернет у пошуку нових ресурсів. Таким чином індексована база даних постійно поповнюється новими ресурсами і користувач може знаходити інформацію і у тому випадку, коли автор ресурсу не докладав спеціальних зусиль для індексації свого ресурсу.

Метапошукові системи відрізняються тим, що здійснюють пошук по підготованим за спеціальними стандартами метаданим.

### 2.3.2 Оцінка релевантності результатів пошуку

Термін релевантність (від англ. *relevant*, той, що відноситься до справи) означає відповідність знайденого документу запиту користувача [46]. Оцінка релевантності передбачає визначення двох характеристик документу.

По-перше визначається ступінь відповідності документу до ключових слів, за якими потім визначається ступінь відповідності запиту користувача.

По-друге визначається статус документу для того, щоб визначити які документи є найбільш “вагомими” і мають бути на перших місцях у результатах пошуку, а які навпаки варто приховати так як імовірно вони не містять корисну для користувача інформацію.

Оцінка релевантності визначає популярність пошукової машини, так як у сучасних умовах запиту користувача можуть відповідати тисячі документів, а якість пошуку користувач оцінює по тим, які потрапили на першу сторінку.

Зараз багато веб-ресурсів, особливо рекламного характеру, використовують спеціальні прийоми для того, щоб посилання на них потрапляли на перші рядки результатів пошуку за популярними запитамі. Вони не містять корисної інформації, що відповідає запиту користувача, і тому мають бути відфільтровані. Пошукові машини намагаються оцінювати, наскільки корисним для користувача може бути документ, і відповідно розподіляють місця у результатах пошуку.

Для оцінки відповідності документу ключовим словам використовуються внутрішні дані документу, і в першу чергу сам текст документу [45]. Оцінюється розподіл фраз і окремих термінів, їх густина та розміщення у тегах. Аналіз густини дозволяє з одного боку відкинути документи, які містять дуже мало необхідної інформації, а з іншого боку відкинути “ловушки”. При аналізі розміщення у тегах вважається, що розміщення ключового слова або фрази у заголовку або виділення у тексті вказує на те, що текст присвячено

даній темі. До уваги також береться текст, розміщений у тегах <alt> для зображень.

Раніше пошукові машини активно використовували спеціальні теги та метадані, такі як елементи <description> та <keywords> html-коду. Зараз така інформація або зовсім не використовується (наприклад у системах “Рамблер” [44, 47] та “Google” [47]) або мають дуже низьку вагу (наприклад, “Апорт” [48]). Це пов’язано з тим, що значення метаданих не відображається користувачу і розробники документів часто розміщують у них інформацію, що не стосується тексту і направлена на підвищення статусу документу для пошукових систем. Тому метадані зараз використовуються тільки для керованих ресурсів, таких як інформаційні колектори з певними обмеженнями на публікацію, бо в такому випадку відповідні особи стежать за коректністю метаданих, вони повністю відповідають документу і пошук за ними значно ефективніше, ніж повнотекстовий пошук.

Сучасні пошукові системи виконують граматичний аналіз тексту для того, щоб результат пошуку не сильно залежав від правил граматики мови запиту. Врахування синонімів дозволить знайти ті результати, які не містять слів запиту в явній формі але містять синоніми і тому бути корисними для користувача. Врахування морфології, тобто відмінювання слів, необхідне для коректного аналізу текстів кирилицею. Ці задачі можуть бути вирішені сучасними пошуковими системами.

Складнішим є визначення омонімів, тобто слів з однаковим написанням та різним смислом. Значення омоніму можна приблизно визначити за сусідніми словами, однак конкретних рішень даної проблеми на сьогодні немає.

Не вирішеною на сьогодні задачею є синтаксичний аналіз. Так, при необхідності знайти вірш Шевченко, за словами «вірш» та «Шевченко» будуть знайдені документи на яких є такий текст, а сторінка з текстом вірша без зазначення слова «вірш» не буде знайдена. На сьогодні цю проблему може вирішити лише автор документу, зазначивши в документі інформацію, що забезпечить його пошук.

Для визначення статусу документу визначається ранг, який залежить від таких факторів:

1. Відповідність тексту посилань до запиту.
2. Релевантність ресурсів, на яких виставлені посилання на обраний ресурс.
3. Популярність посилань.
4. Вихідні посилання зі сторінки.
5. Інші фактори.

Таким чином вищий ранг буде мати той ресурс, який обмінюється посиланнями з ресурсами високого рангу.

В системі Google використовується система ранжування PageRank (PR). Принцип її роботи полягає в тому, що кожна сторінка має PR, яким вона голосує за інші сторінки за допомогою посилань на них, збільшуючи їх PR.

Формула для розрахунку PageRank має вигляд:

$$PR_a = (1 - d) + d \cdot \sum_{i=1}^n \frac{PR_i}{C_i}, \quad (2.1)$$

де  $PR_a$  – ранг сторінки, що розглядається;

$d$  – коефіцієнт затухання;

$PR_i$  – ранг  $i$ -ї сторінки, що містить посилання на сторінку, що розглядається;

$C_i$  – загальна кількість посилань на  $i$ -й сторінці.

Розрахунок PageRank графічно продемонстровано на рис. 2.3. Для спрощеного пояснення коефіцієнт  $d$  покладено рівним 1. Для кожного документу зазначено його PR. Стрілки відповідають посиланням, для кожної зазначено який PR дане посилання передає документу, на який посилається. PR ресурсу рівномірно розподіляється між посиланнями, які на ньому містяться.

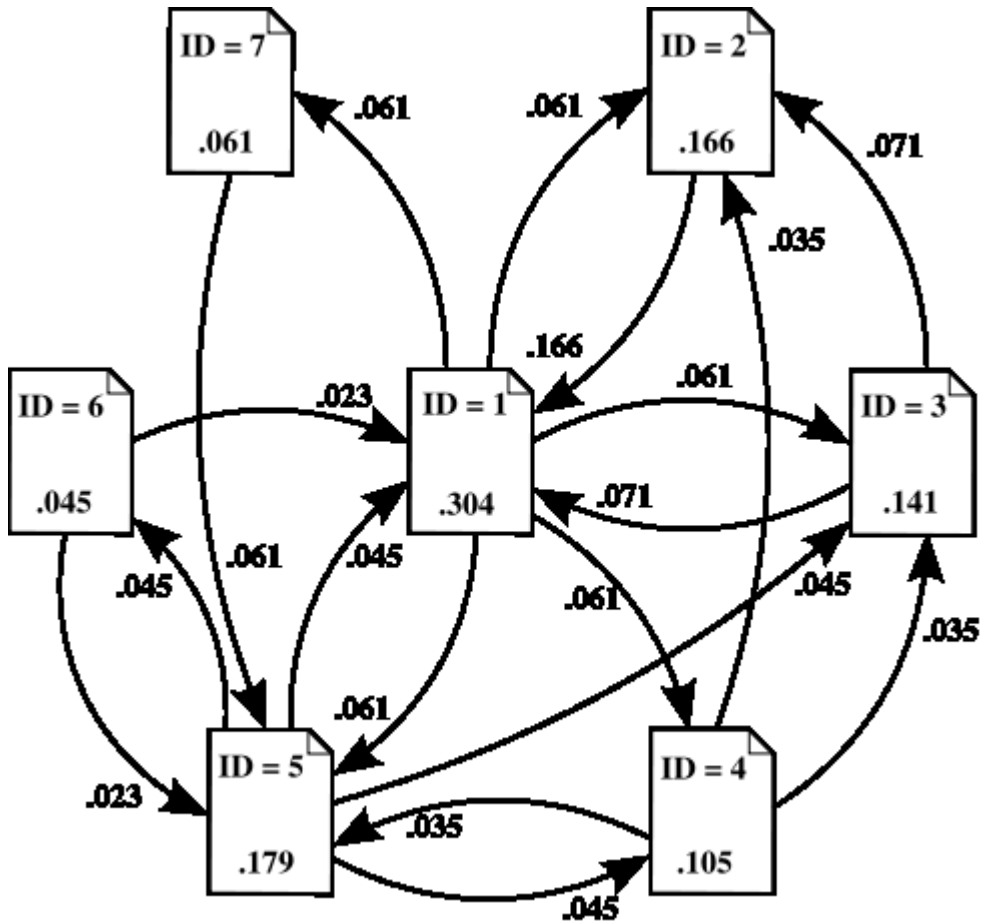


Рисунок 2.3 – Ілюстрація підрахунку Google PageRank

Модель рангів проектувалася таким чином, що PageRank відповідає ймовірності знаходження користувача на сторінці. Коефіцієнт затухання задає ймовірність того, що користувач буде продовжувати перехід за посиланнями. За статистичними підрахунками його значення приблизно рівне  $d=0.85$  [49]. Крім того, з математичної точки зору, значення менше 1 забезпечує збіжність оцінки.

Таким чином ранг сторінки розподіляється між посиланнями на інші сторінки. Обчислення PageRank є циклічним процесом, який проводиться до тих пір, поки значення на сусідніх ітераціях не будуть розрізнятися менше встановленого порогу. Так як ранг сторінки відповідає ймовірності

знаходження користувача на сторінці він може адекватно описати авторитетність сторінки і бажаність її як результату запиту.

Значення PR є елементами власного вектору модифікованої матриці суміжності. Це робить PageRank ефективною метрикою, яка може бути швидко обрахована. Власний вектор  $\mathbf{r}$  є розв'язком рівняння (2.2).

$$\mathbf{r} = \mathbf{M} \mathbf{r}, \quad (2.1)$$

$$R = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} l(p_1, p_1) & l(p_1, p_2) & \dots & l(p_1, p_i) \\ l(p_2, p_1) & \ddots & & \vdots \\ \vdots & & l(p_i, p_j) & \\ l(p_N, p_1) & \dots & & l(p_N, p_i) \end{bmatrix}, \quad (2.2)$$

де  $\mathbf{r}$  – вектор значень PageRank,  $F_i$  – PageRank сторінки  $i$ , значення

$l_{ij}$  рівне 0 якщо посилання зі сторінки  $i$  на сторінку  $j$  немає і 1, якщо таке

посилання є,  $N$  – кількість сторінок, що розглядається,  $d$  – коефіцієнт затухання.

До початку обчислень матриця нормалізується так, щоб виконувалось рівняння

$$\mathbf{M} \mathbf{1} = \mathbf{1}, \quad (2.3)$$

Власний вектор швидко збігається і необхідні лише декілька ітерацій, які дають значення PR одразу для набору документів.

Так як Google розцінює PageRank як голосування одних сторінок за інші, з'являються нові механізми, що управляють таким голосуванням. Так розробник сторінки може зазначати, які посилання не будуть використовуватись для обчислення рангу.

Деякою модернізацією PageRank є оцінка LocalRank, запатентована Google у 2003 році [50]. Даний алгоритм відрізняється тим, що для розрахунку рангу вибираються групи документів за певною ознакою. У такому випадку ранжування проводиться серед документів, що відповідають запиту і має бути точним. Однак залишається невирішеною проблема автоматичного розбиття набору документів на групи. Даний алгоритм не було реалізовано.

Алгоритм TrustRank, який використовується "Yahoo", визначає "рівень довіри до сторінки" [51]. Для цього експерти оцінюють "рівень довіри" початкових сторінок вручну а потім система автоматично оцінює інші ресурси Інтернету проходячи за посиланнями. Перевагою алгоритму є висока надійність так як експерти легко і надійно визначають чи є сторінка спамом. Але при цьому аналіз достатньої кількості сторінок є надто дорогим і довгим.

Різні пошукові системи використовують різну вагу розглянутих параметрів [42]. Так, наприклад, MSN більшу увагу приділяє текстовому наповненню сторінки. Google акцентує увагу на посиланнях на дану сторінку і оптимізує алгоритми для максимальної швидкості індексації.

Yandex також оцінює ранг сторінки за кількістю посилань але в даному випадку більшу вагу мають тематичні посилання. При цьому час індексації дещо вищий ніж у Google але, особливо якщо врахувати направленість Yandex на російський сегмент Інтернету, швидкість індексації є прийнятною.

Rambler є найменш документованою системою із популярних публічних пошукових систем. Він надає велику вагу термінам у простому тексті документів, без виділення у спеціальні теги. З результатів пошуку ефективно виключаються спам-ресурси та ресурси, які займаються "розкруткою" за рахунок посилань.

Багато пошукових систем використовують результати бази даних Google. В тому числі Google пропонує спеціальну пошукову машину, яка може бути встановлена для пошуку у межах окремого ресурсу. Наприклад, mail.ru використовує результати пошуку Google з внесенням невеликих змін.

### 2.3.3 Додаткові механізми пошукових систем

Останнім часом почалася розробка засобів для оптимізації пошуку по файловій системі ПК, які дозволяють замінити повний перебір на швидший метод пошуку (Google Desktop Search, „Персональний Поисковик „Яндекса”). Принцип роботи таких засобів полягає в індексації файлів на жорсткому диску. Таким чином, файлова система певним чином структурується і пошук за текстом файлів відбувається практично миттєво (так саме, як і для пошукових систем в інтернеті). Недоліком таких засобів є те, що вони підтримують обмежену кількість форматів, так як знання формату є необхідним для індексації корисного для користувача матеріалу.

Проблемою при розробці таких засобів є розробка нових методів оцінки релевантності. Такі методи як PageRank, що базуються на посиланнях, неприйнятні внаслідок відсутності самих посилань. Крім того, при пошуку на файловій системі зникає задача оцінки правильності і корисності документу, так як користувач свідомо розміщує у себе лише ті документи, які сам вважає важливими. В даному випадку виникає задача виключення з індексу службових файлів. Замість традиційного і все більш актуального для інтернет-пошукових систем завдання фільтрації спаму більшого значення набуває задача оцінки відповідності документу конкретному запиту. На сьогодні не існує ефективної системи оцінки релевантності для документів користувача і результати пошуку можуть бути дещо “дивними”.

Пошук інформації досить часто організовується ієрархічно, тобто ведеться пошук у результатах попереднього пошуку. У таких випадках пошук буде проводитися швидше, якщо зберігати проміжні результати пошуку. Таким чином можна не починати пошук спочатку. Однак такий підхід складно



застосувати у деяких пошукових системах, так як він вимагає додаткових ресурсів. Тому досить часто ієрархічний пошук реалізується простим додаванням нової умови до попередніх (Наприклад, у пошуковій системі Google).

Використання індексації, класифікації та тегування вимагає первинної обробки інформації. Така обробка включає аналіз інформації з метою її класифікації. Таким чином неструктуровану інформацію можна певним чином структурувати. Аналіз документів може проводитись за змістом або за діями користувачів. Аналіз за діями користувачів дозволяє з'ясувати, наскільки документ відповідає визначеним для нього ключовим словам чи класам. Прикладом може бути індексація веб-ресурсів роботами пошукових систем. Такі роботи оцінюють релевантність сторінок. Для фільтрування „небажаних” сторінок можна використовувати сервіси, які дозволяють користувачам відмічати їх, так що в подальшому вони не будуть включатися в пошук.

#### 2.4 Інтеграція та інтероперабельність інформаційних ресурсів

Розробка ізольованих інформаційних ресурсів в сучасних умовах часто не дає позитивного результату [6, 52, 53]. Це пов'язано з тим, що в сучасні організаційні системи працюють не ізольовано а у тісній співпраці з іншими Орг.С. Тому забезпечення взаємодії між інформаційними системами організацій забезпечує кращу цілісність даних та швидкість передачі інформації.

На рівні файлових систем інтероперабельність забезпечується засобами операційних систем. При цьому досягається певний рівень розподілу доступу, однак така взаємодія обмежується наперед заданими можливостями ОС.

Взаємодія між базами даних та сховищами організується, як правило, в межах конкретних ІКС. При цьому розробляються спеціальні механізми передачі даних між системами. Це пов'язано з тим, що характер інформації у таких типах ресурсів як правило передбачає її закритість. Крім того самі дані та інформація як правило не представляють великої цінності без наявності

відповідних методів та засобів обробки. Тому створення глобальних методів взаємодії між базами даних та інформаційними сховищами у загальному вигляді не є доцільним. Розробляються спеціальні алгоритми взаємодії окремих визначених БД та сховищ із забезпеченням необхідного рівня безпеки та максимальної швидкості взаємодії.

Інформаційні колектори розробляються для збереження знань. Знання дуже часто необхідно зробити відкритими та загальнодоступними. Тому

Для забезпечення розповсюдження наукових знань у світі розробляються засоби взаємодії наукових архівів. Ініціатива “Відкриті архіви” OAI (Open Archives Initiative) [38], започаткована розробниками системи EPrints, розробляє і просуває стандарти взаємодії та інтеоперабельності бібліотечних архівів з метою більш ефективного використання наукових знань. Ініціатива включає в себе протокол збору метаданих, оформлених за визначеними стандартами. Ті системи, які підтримують OAI, можуть бути поєднані у загальну мережу архівів, які взаємодіють через задані інтерфейси.

Для того, щоб користувач міг швидко і з одного місця знайти потрібну йому інформацію була розроблена ініціатива OCLC (Online Computer Library Center) [40]. Її ідея полягає у тому, що кожному документу ставиться у відповідність набір метаданих, за якими проводиться класифікація і пошук. Таким чином користувач за своїм запитом може отримати посилання на документи у різних ресурсах різних організацій по всьому світу.

У 2005 році ідеї OCLC були реалізовані в системі WorldCat, яка забезпечує автоматичну індексацію більше 10000 бібліотечних архівів і містить посилання на більш ніж 1.2 млрд. документів [39]. Документи у всіх розглянутих системах можуть бути індексовані у WorldCat. Система CONTENTdm забезпечує автоматичну індексацію документів у каталозі WorldCat, тобто без необхідності яких-небудь дій зі сторони користувача.

Варто зазначити, що практично всі із зазначених функціональних можливостей можуть бути забезпечені і для систем, які не мають базової

підтримки таких можливостей. Але така підтримка за допомогою сторонніх по відношенню до ресурсу засобів потребує додаткових розробок та витрат.

Враховуючи постійну глобалізацію та інтернаціоналізацію знань виникає потреба у створенні системи або мережі інформаційних ресурсів. Мережа Інтернет ефективно використовується для розповсюдження наукових знань. При цьому не можна не помітити, що сучасні засоби пошуку (такі як Google, Yahoo) не забезпечують користувачу можливості швидко знайти потрібні документи, особливо у випадку коли користувач не знає точно що він шукає. Однак такі пошукові системи можуть більш-менш ефективно працювати з ресурсами, які не приймають участь в глобальних проектах типу WorldCat. Саме тому необхідним є поєднання можливостей всіх пошукових систем. Модель глобального інформаційного ресурсу представлена на рис. 2.4.

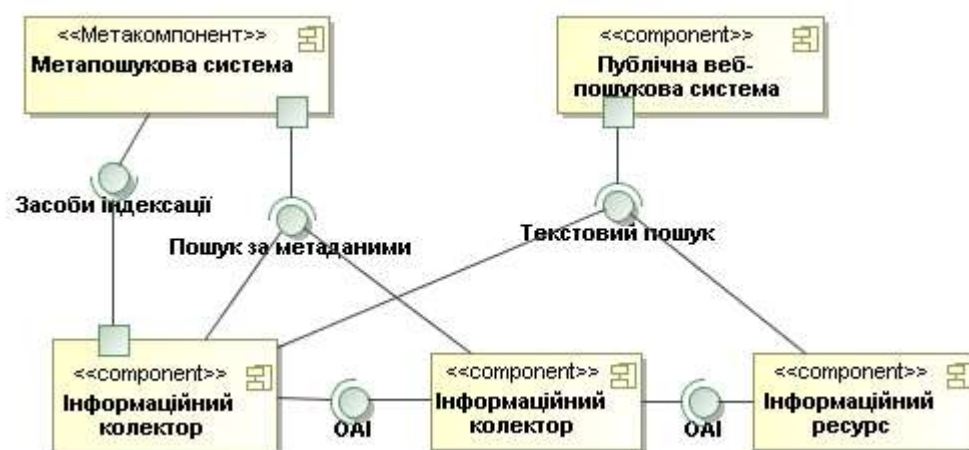


Рисунок 2.4 – Модель глобального інформаційного ресурсу

Таким чином мета пошукові та публічні пошукові системи індексують і таким чином упорядковують систему інформаційних ресурсів. Самі ресурси можуть взаємодіяти через стандарти типу OAI, але така взаємодія є обмеженою у порівнянні з системами класу WorldCat. З іншого боку упорядкована взаємодія колекторів дозволяє створювати системи масштабу країни, такі як Проект цифрових музеїв Китаю [54].

## ВИСНОВКИ ДО РОЗДІЛУ 2

Інформаційні ресурси є невід’ємною складовою ІКС Орг.С. Кожен з розглянутих типів інформаційних ресурсів використовується для задоволення певного кола інформаційних потреб.

Класифікація інформаційних ресурсів є основою для визначення типу ресурсу та його характеристик. Формулювання вимог до ресурсу ІКС на основі вказаної класифікації дозволяє обґрунтовано обрати один або декілька типів ресурсів, які мають бути розроблені та впроваджені.

При розробці ресурсу слід передбачити засоби ефективного пошуку документів. Для цього можуть використовуватись як спеціально розроблені засоби так і пошукові та метапошукові системи. Ефективне поєднання сучасних засобів пошуку дозволяє забезпечити максимальну ефективність роботи користувача з ресурсом.

Функціональні вимоги, що стосуються взаємодії між компонентами ресурсу та між ресурсом та іншими компонентами ІКС, пропонується описувати за допомогою інтерфейсів. Використання мови моделювання UML дозволяє забезпечити підтримку необхідної функціональності при проектуванні окремих компонентів.

## РОЗДІЛ 3. ПРОДУКУВАННЯ ІНФОРМАЦІЙНИХ РЕСУРСІВ ІНФОРМАЦІЙНО - КОМУНІКАЦІЙНИХ СИСТЕМ

Продуктування інформаційних ресурсів є одним з основних завдань інформаційного суспільства [21]. Під продуктуванням інформаційних ресурсів ми розуміємо створення нових видів інформаційних ресурсів на основі існуючих даних, інформації, знань [2, 3].

Для продуктування інформаційних ресурсів необхідно [2, 3, 21]:

- вивчати структуру та загальні властивості документальної інформації;
- провести аналіз закономірностей інформаційних комунікацій.

З теоретичної точки зору, для ефективного продуктування інформаційних ресурсів необхідно провести розробку теоретичних засад кумуляції, аналітико-синтетичної переробки, збереження й використання даних і знань [16, 21, 55].

Це є основною проблемою при створенні автоматизованих систем продуктування інформаційних ресурсів [9, 16, 55].

Розробка інформаційного ресурсу повинна базуватись на проведеному моделюванні ІКС. Це дозволяє розробити всю систему так, що вона з одного боку зможе надійно функціонувати в цілому та з іншого може бути розділена на окремі компоненти, що будуть розроблені окремо [11].

Засоби продуктування інформаційного ресурсу можна розділити на ручні та автоматизовані. Автоматизовані засоби відрізняються певним ступенем автоматизації окремих ключових етапів розробки.

### 3.1 Ручні засоби

Історично першою технологією продуктування інформаційних ресурсів є ручна обробка даних, інформації, знань. Цей процес проводиться розробниками за певними інструкціями. При цьому можуть використовуватись певні автоматизовані засоби. Термін «ручні» тут означає, що забезпечення

взаємодії між різними компонентами ресурсу та між ресурсом та іншими компонентами ІКС організовується вручну розробником системи.

Для налаштування файлових систем в такому випадку користувач за інструкцією, розробленою відповідними спеціалістами, організовує файловий простір. При цьому використовуються механізми ОС. На сьогодні це практично єдиний засіб створення ФС. Можлива певна автоматизація за допомогою певних скриптів або спеціально розроблених програмних засобів. При цьому мова йде саме про налаштування ФС, що включає організацію ієрархії файлів і каталогів. Розробка засобів фізичного збереження даних на носії для створення ІКС окремої Орг.С. як правило не розглядається, а обирається один із існуючих варіантів реалізації ФС.

Для продукування БД ручними засобами необхідно розробити структуру БД, при цьому розробник має забезпечити відповідність структури сучасним правилам побудови БД, наприклад відповідність 3-й нормальній формі (3 НФ).

Наступним кроком стає створення самої БД засобами системи управління базами даних (СУБД). Розробник сам відповідає за забезпечення взаємодії між базою даних та іншими компонентами ІКС, зокрема з іншими ресурсами. Такий підхід на сьогодні використовується для малих проектів, у яких використання автоматизованих засобів не виправдано з економічної точки зору і не дає значного приросту продуктивності. Створення альтернативної архітектури БД та СУБД в сучасних умовах не актуальне, так як витрати на розробку тільки окремих функцій перевищують вартість придбання повного рішення від світових лідерів ринку. Крім того, існують досить потужні безкоштовні системи.

Так як інформаційні сховища базуються на БД, тому для створення інформаційних сховищ вручну необхідно створити спеціальну схему БД та засоби її наповнення. Тому при створенні сховищ при ручному проектуванні можна використовувати автоматизовані засоби розробки БД. В окремих випадках ручне створення засобів наповнення сховища є виправданим, враховуючи значну вартість готових систем.

Ручна розробка інформаційних колекторів не є актуальною задачею на сьогодні, так як безкоштовні автоматизовані системи надають добру функціональність. При цьому поставка з відритим кодом дозволяє за необхідності доповнити систему бажаною функціональністю.

## 3.2 Автоматизовані засоби

Зараз все більшого поширення набувають автоматизовані системи продукування інформаційних ресурсів. До них можна віднести засоби автоматизації розробки та наповнення баз даних, інформаційних колекторів, тощо. Зазначимо, що бази даних, сховища та колектори стають інформаційними ресурсами лише тоді, коли вони наповнені даними, інформацією та знаннями. Крім того, ці засоби забезпечують створення лише певних видів інформаційних ресурсів і не визначають загального підходу до створення інформаційного ресурсу. А у деяких випадках, окремі бази даних (інформаційні сховища, інформаційні колектори) можуть бути лише фрагментами великого інформаційного ресурсу.

### 3.2.1 Побудова баз даних

Для реалізації баз даних створені та постійно розробляються системи управління базами даних (СУБД), такі як MySQL, MS SQL, Oracle, Informix, тощо. Для автоматизованого проектування баз даних існують декілька інструментів, таких як PowerDesigner, MagicDraw, Rational Suite, та інші. За допомогою таких засобів можна спроектувати базу даних засобами графічної мови моделювання UML. За такою моделлю автоматично створюється схема бази даних, а також може бути частково згенероване програмне забезпечення по робот з нею. Принципи розробки баз даних за допомогою графічного моделювання описані в [4, 12].

### 3.2.2 Побудова інформаційних сховищ

Сховища базуються на реляційних базах даних, спеціальних платформах чи OLAP-серверах. Дані, які містяться у них, обробляються спеціальними модулями, які виконують функції вивантаження, очищення та фільтрації оперативних даних, візуалізацію даних та генерацію звітів. Для побудови таких систем розробляються автоматизовані засоби проектування, які дуже часто використовують мову UML [56]. Рішення для проектування, продукування та впровадження сховищ розробляються компаніями IBM, Oracle, Hewlett-Packard, NCR, Informix Software, SAS Institute, Sybase, Software AG та іншими [56, 57, 58]. Аналіз основних розробок наведено у табл. 3.1.

Таблиця 3.1 – Аналіз засобів продукування інформаційних сховищ

№ п.п.	Назва і розробник	Функціональні характеристики	Технічні характеристики	Економічні характ-ки
1	Microsoft	Налаштування продуктивності; Пакетна поставка; Функції чисельного аналізу; Акумуляування запитами користувача	Базується на MS SQL Server; ODBC, JDBC, ADO, MDX, DSO, Pivot Table, XML аналізу;	20 тис. дол. на процесор
2	Oracle	Широкий набір типів індексації; Розширене поєднання; Різний розмір блокувань; Чисельний аналіз з прогнозуванням	Базується на Oracle 9i; OLAP DML: Java API;	40 тис. дол. на процесор
3	Data Warehouse Plus, IBM	Гібридний аналіз; Модульність; Гнучкість збору даних	Базується на DB2; Розширюваність через Java, C, Visual Basic	25 - 138 тис. дол. на процесор
4	Essbase, Hyperion	Служби інтеграції сховищ Гібридний аналіз	Довільна БД; Розширюваність через Java, C, Visual Basic; Мультивалютність	28 тис. дол. на процесор
5	Sybase	Індекси: бітові, швидкі, низькорівневі, словесні, аналітичні, поєднані, порівняльні; Зберігання та індексація XML; Імпорт з DB2 та Oracle	Вбудований HTTP(S) сервер; Інтеграція через .Net; Інтероперабельність через EnterpriseConnect	40 тис. дол. на процесор
6	CFO Vision, SAS	Інтегровані фінансова звітність та аналітичні звіти; Багаторівнева структура з вкладеними класифікаціями; Визначення метаданих	Базується на SAS OLAP-Server або сторонніх серверах; Мультивалютні звіти; Інтегроване ПЗ	17-50 тис. дол. на процесор



Наведені економічні показники є приблизними так як запропоновані рішення мають декілька варіантів поставки. Так, наприклад, IBM пропонує окремо базу даних DB2 з базовими функціями сховища за 25 тис. дол. OLAP Server за 28 тис., Intelligent Miner за 75 тис. Кожна з систем надає певні розширені функції, але лише комбінація їх всіх надає повні функціональні можливості інформаційного сховища.

З таблиці видно, що всі системи на сьогодні пропонують певні рішення для забезпечення інтегруєбельності. Швидко розвивається функціональність, пов'язана з математичною обробкою інформації. Сучасні системи з одного боку підтримують БД різних виробників, а з іншого наголошують на перевагах використання єдиної однорідної платформи.

### 3.2.3 Побудова інформаційних колекторів

На сьогодні на ринку систем продукування інформаційних ресурсів представлені як безкоштовні так і платні рішення. Для аналізу були обрані 6 найбільш популярних безплатних систем:

- Fedora, яка була першим проектом по створенню універсальної технології продукування інформаційного колектору [59];
- DSpace, розроблена Масачусетським Технологічним Інститутом (MIT) у співробітництві із Hewlett-Packard для забезпечення інформаційних потреб інституту [60];
- CDS Invenio, розроблена Європейською організацією ядерних досліджень для зберігання ресурсів у галузі ядерних досліджень [61];
- Greenstone, проект Новозеландського Проекту Цифрових Бібліотек і Університету Вайкато, призначений для створення простого і компактного ресурсу [62, 63];
- EPrints, який був розроблений університетом Саутгемптону на основі дослідження вимог освітніх організацій до інф. колектору [64];

- Archimede, розроблений бібліотекою університета Лавалю на базі DSpace [65].

Також були розглянуті 2 платні системи:

- Verpress, сервер інформаційних ресурсів, розроблений Berkeley Electronic Press для створення та зберігання інформаційних ресурсів користувачів [66];
- CONTENTdm, розроблений DiMeMa, Inc. як інструмент для створення та управління інформаційними архівами з широкими функціональними можливостями [67].

Порівняльний огляд основних показників існуючих рішень наведено у табл. 3.2 [37].

Таблиця 3.2 – Порівняльний аналіз основних показників інформаційних колекторів

Характеристика		Fedora	DSpace	CDS Invenio	Greenstone	Eprints	Archimede	bepress	CONTENTdm
Загальні	Розробник	Fedora Commons	MIT + HP	CERN	NZ Dig.Lib.Proj + Univ Waikato	Univ. Southampton	Laval University Library	Berkeley Electronic Press	DiMeMa, Inc.
	Перша версія	1997	2002						
	Поточна версія	2.2 (2007)		~2006					
	Продовження розробки	+	+	+	+	+	+	+	+
	Розміщення	Корист.	Корист.	Корист.	Корист.	Корист.	Корист.	bepress	Корист.
	Кількість застосувань		307						
Економічні	Вартість повна	0-GPL	0-GPL	0-GPL	0-GPL	0-GPL	0-GPL	немає	7-40 тис. \$
	Вартість річна	0	0	0	0	0	0	8-50 тис. \$ *	1-6 тис. \$
	Відкритість	OS	OS	OS	OS	OS	OS	закритий	закритий
	Додаткова вартість	0	0	підтримка	0	0	0	4-5 тис \$ ** за серію (без *)	2500+ тис. \$ налаштування
Функціональні	Різні формати файлів (+ AV)	+	+	?	-	+	+	?	+
	Засоби веб-доступу	+	+	+	-	+	+	+	?
	Інтеграція з сайтом	+	+	+	-	+	+	-	+
	Персоналізація веб-інтерфейсу	-	+	+	-	-	+	-	+
	Інтеграція з навчальними системами	-	+	-	-	+	+	-	-
Повнотекстовий пошук	+	Lucene Google	Google	+	+	Lucene	+	по колекціям	

Продовження таблиці 3.3 – Порівняльний аналіз основних показників інформаційних колекторів.

Функціональні	Процес по прийняттю документів		+ Децентралізований		-	+			
	Багатомовність	-	-	-	4(rus)+25	-	3, no rus	-	-
	Індексація у WorldCat	-	-	-	-	-	-	-	+
Технічні характеристики	Метадані	Dublin Core metadata	Dublin Core metadata	MARC 21	Dublin Core Metadata	Довільна схема	Dublin Core Metadata	Dublin Core Metadata	Dublin Core Metadata
	XML	+	+	+	+	+	+	+	+
	OAI	+	+	+			+	+	+
	Z39.50	-	-	-	-	-	-	-	+
	RSS					+			
	Розширюваність	API for services	Java API	API		API-Perl			
	Версійність контенту	+							
	Міграція	+ утіліта							
	БД		Postge SQL or SQL-transact DB (Oracle MySQL...)	MySQL		MySQL			
	Export to CDROM	-	-	-	+	-	-	-	-
Платформа				Any	Unix				

Серед основних характеристик слід виділити розміщення системи та кількість застосувань. Розміщення визначає, де фізично розташовується бібліотечний архів. При розміщенні у користувача (Корист.) замовник певним чином отримує дистрибутив системи (самостійно завантажує з інтернету або купує) і встановлює її на своєму обладнанні (можливо, за допомогою розробника). При цьому CONTENTdm та CDS Invenio пропонують послуги по встановленню системи та навчанню, але за додаткові кошти. Натомість система Verpress пропонує розміщення основного програмного забезпечення та інформації на сервері розробника, таким чином користувач не витрачає кошти на придбання та налаштування обладнання. Але при цьому вартість такого рішення вища. Кількість застосувань дозволяє оцінити популярність системи [37].

Важливим блоком досліджуваних характеристик були економічні характеристики. Перші 6 систем є безкоштовними відкритими системами, вони розповсюджуються за умовами GPL (General Public License) з відкритим кодом. Останні дві системи (Verpress та CONTENTdm) є платними продуктами. Серед економічних характеристик виділені повна вартість системи, річна вартість, вартість використання та вартість підтримки. Verpress не передбачає повної покупки системи, так як користувач постійно використовує обладнання розробника. Натомість пропонуються альтернативні варіанти оплати річного користування або оплати за додавання порції документів. Обидві платні системи а також CDS Invenio пропонують платну підтримку, встановлення системи та навчання.

Функціональні характеристики дозволяють оцінити відповідність технології вимогам, які виставляються перед ресурсом. Останні версії практично всіх систем, за виключенням GreenStone, дозволяють працювати з документами різних форматів, включаючи текст, зображення, аудіо- та відеозаписи. При цьому передбачаються відповідні засоби обробки інформації, такі як перегляд, масштабування, тощо.

Як вже було зазначено, для забезпечення доступу до ресурсу є засоби веб-доступу. Саме тому більшість розробників інформаційних колекторів інтегрують

у свої розробки підтримку таких засобів. Важливим аспектом такої підтримки є можливість інтеграції засобів веб-доступу до ресурсу із порталом замовника, що передбачає оформлення ресурсу у заданому стилі, підтримку зручних для користувача функцій, тощо.

Однією із важливих функцій інформаційного ресурсу сьогодні є забезпечення навчального процесу. Оригінальною функцією, яку забезпечують лише DSpace та побудований на його основі Archimede є можливість інтеграції із навчальними системами.

Для забезпечення ефективної роботи користувачів із ресурсом необхідно передбачити можливості пошуку по ресурсу [6]. Більшість колекторів забезпечують можливості як повнотекстового пошуку (пошуку за фрагментами текстів) так і пошук за метаданими. Метадані являють собою певну додаткову інформацію, яка додається до документу. Пошук за метаданими дозволяє забезпечити швидший та більш релевантний пошук, але лише за умови “правильних” метаданих та запитів “користувача”. Деякі з систем використовують існуючі розробки у сфері пошуку, такі як Lucene та Google, що робить пошук звичним для користувачів таких систем.

Процес по прийняттю документів передбачає наявність засобів, які контролюють додавання документів для ресурсу. Це актуально для відкритих ресурсів, додавати документи у які може будь-який користувач системи. Надання таких можливостей користувачам вимагає забезпечення засобів контролю за якістю документів, що додаються. Як правило, документи приймаються адміністратором системи, але до цього процесу можуть бути залучені інші користувачі.

Багатомовність передбачає підтримку декількох мов як інтерфейсу користувача так і документів. Створення різних мов інтерфейсу є тривіальною задачею і підтримується всіма системами. Значно важчим є забезпечення підтримки різних мов документів, так щоб користувач міг обирати мову, на якій він хоче отримувати інформацію. При цьому виникають задачі по забезпеченню пошуку та організації метаданих.

Технічні характеристики визначають конкретні характеристики ресурсу що впливають на забезпечення функціональності ресурсу, порядок його створення і використання.

Для забезпечення взаємодії інформаційних ресурсів необхідно встановити стандарт метаданих. У обраних для аналізу технологіях використовуються такі стандарти метаданих:

- MARC – Machine Readable Cataloging – запропонований у 1980 р. стандарт метаданих.
- Dublin Core Metadata – стандарт, запропонований у 1995 році.

Обидва вказані стандарти передбачають створення набору визначених тегів для кожного документу з можливістю їх використання для індексації, класифікації, пошуку, тощо. Теги, визначені цими стандартами, можуть бути поставлені у взаємну відповідність.

Формат XML сьогодні став одним з найбільш популярних форматів представлення даних. Тому більшість інформаційних колекторів підтримують експорт та імпорт даних у XML.

Z39.50 являє собою затверджений ISO клієнт-серверний протокол, який забезпечує пошук та збір інформації у базах даних віддалених комп'ютерів. Стандарт був розроблений на початку 70-х і зараз є популярним у бібліотечних системах [68].

RSS (Rich Site Summary, Really Simple Syndication) – набір XML-форматів, призначених для опису новин, змін на сайтах, блогах, інших інформаційних ресурсах. Інформація, зібрана за допомогою RSS, може бути агрегована і представлена у зручному для користувача вигляді спеціальними програмами-агрегаторами. Таким чином користувач може у короткому вигляді отримувати інформацію про зміни на ресурсі, який підтримує RSS [69].

Розширюваність інформаційного ресурсу досягається через забезпечення можливості підключення до нього додаткових модулів. Більшість ресурсів підтримують API (Application Programming Interfaces) – програмні інтерфейси для забезпечення взаємодії. Ці API можуть бути основані на мовах Perl, Java, та ін.

Версійність контенту полягає в призначенні документу номерів версій, за якими можна буде від слідкувати зміни у документі. Ця функція особливо важлива для інформаційних ресурсів, документи у яких змінюються, розвиваються.

Враховуючи різноманіття технологій та стандартів інформаційних ресурсів, а також їх стрімкий розвиток, інколи доречним є перехід з одного продукту на інший (або з однієї версії на сильно змінену нову). У таких випадках необхідні інструменти міграції. Наявність таких інструментів може бути додатковою перевагою продукту при прийнятті рішення про його впровадження.

Основою інформаційного колектора є баз даних у тому чи іншому виді. Як правило, для організації бази даних для безкоштовних систем обирається MySQL, так як цей продукт також є безкоштовним.

Порівняльний аналіз основних показників існуючих рішень зведено у табл. 3.4.

Із таблиці видно, що платні системи пропонують кращу функціональність, ніж безкоштовні. Тому такі системи найбільш прийнятні для компаній, які хочуть швидко та без ризику отримати інформаційний колектор. Державні установи, наукові та освітні організації, які є основними виробниками сучасних знань, обирають безкоштовні рішення, які теоретично мають аналогічні можливості, але потребують значних зусиль для налаштування і розгортання.

Практично всі системи на сьогодні підтримують сучасний стандарт метаданих Dublin Core, текстовий пошук у певному вигляді та засоби веб-доступу. Важливою характеристикою для впровадження в Україні є багатомовність системи. При цьому лише одна із розглянутих систем (Greenstone) підтримує російську мову в базовій версії.

Сучасні системи підтримують зберігання документів довільного формату. Такі системи, як Fedora мають засоби поєднання мультимедійного контенту із засобами веб-доступу, що дозволяє забезпечити реалізацію гнучкого та зручного інтерфейсу користувача.



Таблиця 3.4 – Порівняльний аналіз основних показників систем продукування інформаційних колекторів.

№ п/п	Назва системи	Призначення	Функціональні характеристики	Технічні характеристики	Економічні характеристики
1	Fedora	Розробка технологій розробки, створення, розповсюдження та повторного використання інформаційних ресурсів які формують інтелектуальну спадщину людства	Збереження інформації різноманітної природи з довільними зв'язками Збереження метаданих Засоби доступу через веб	Метадані Dublin Core Імпорт/експорт XML Маштабованість Розширюваність (API для управління і доступу) ОАІ-підтримка Версійність контенту Утіліта для "міграції"	Безкоштовна Open source Mozilla open source License
2	DSpace	Репозиторій для збереження, накопичення, .. Інформації інституту (intitutional repository) (розроблявся для MIT)	Збереження інформації у довільному форматі Взаємодія з навчальними системами Персоналізація веб-інтерфейсу Повнотекстний пошук (lucene, google) Воркфлоу по прийняттю документів Децентр. процес наповн.	Метадані Dublin Core Імпорт/експорт XML ОАІ-підтримка Розширюваність через Java API БД: PostgreSQL, або транзакційна SQL	Безкоштовна Open source
3	CDS Invenio	Створення електронних бібліотечних каталогів, системи документів, серверу препринтів для великих систем, зокрема ядерної енергетики	Повнотекстний пошук Персоналізація користувача, кошики документів, e-mail повідомлення	ОАІ-підтримка Стандарт метаданих MARC 21 БД: MySQL Розширюваність через API Пошук (Google-like) Обмеж. підтримка e-mail	Безкоштовна Open source Платна підтримка
4	Eprints	Організація репозитарію препринтів результатів наукових досліджень	Збереження інформації у довільному форматі Повнотекстний пошук Довільна схема метаданих Веб-інтерфейс Воркфлоу по прийняттю документів	БД: MySQL Розширюваність через API (Perl) генерація RSS	Безкоштовна Open source
6	Greenstone	Створення бібліотечних архівів	Багатомовність (з російською).	Демонстрація перед побудовою колекції Запис на диск	Безкоштовна Open source GNU GPL
7	Achimede	Створення інститутських репозитаріїв з багатомовною підтримкою	Продовження Dspace Багатомовність (English, French and Spanish)	Пошук (Lucene) ОАІ-підтримка Метадані Dublin Core	Безкоштовна Open source
8	bepress	Створення, зберігання, забезпечення доступу, до інформації користувачів у сховищі розробника (Berkeley Electronic Press)	Інструментарій користувача Налаштування через API Повнотекстний пошук	ОАІ-підтримка Експорт даних у XML	Повна: 8-50 тис. дол./ рік за контент: 4-5 тис. дол. За серію
9	CONTENTdm	Засоби для створення і управління інформаційними сховищами із забезпеченням максимально можливої функціональності	Всі типи контенту Пошук по колекціям Автоматична індексація у WorldCat Поставляється як програмний продукт на комп'ютери користувача	ОАІ-підтримка Метадані Dublin Core Експорт/імпорт XML Сумісність Z39.50	Повна: 7-40 тис. дол./ рік Підтримка: 1-6 тис. дол. на рік Інсталяція: 2500 дол.

### 3.2.4 Побудова веб-ресурсів

Веб ресурс є специфічним типом інформаційного ресурсу. Він базується на одному або декількох з інших типах ресурсів та забезпечує особливі засоби доступу до інформації у них.

З точки зору компонентного підходу розробку веб-ресурсу можна поділити на розробку ресурсу іншого типу (бази даних, інформаційного сховища, інформаційного колектору) та розробку засобів доступу. Необхідно створити модель веб-ресурсу, яка опише взаємодію між цими двома компонентами на рівні інтерфейсів.

Розробка ресурсу за визначеними вимогами описана у попередніх розділах. Для забезпечення засобів доступу ефективно використати корпоративні портали. Принципи розробки і побудови порталів описані у [13].

## ВИСНОВКИ ДО РОЗДІЛУ 3

Компонентний підхід передбачає розділення ІКС на окремі компоненти. У випадку створення комбінованого ресурсу (наприклад, такого як наведено на рис. 2.1) автоматизовані засоби продукування інформаційних ресурсів дозволяють значно покращити процес розробки компонентів. Сучасні засоби розробки інформаційних ресурсів підтримують стандарти взаємодії ресурсів. Завдяки таким стандартам забезпечується автоматизоване створення компонентів з підтримкою поставлених до них вимог (інтерфейсів).

Для проектування всього ІКС та зокрема інформаційного ресурсу як його компоненту доречно використати мову графічного моделювання UML. Сучасні засоби автоматизації створення ресурсів дозволяють за отриманими моделями в певній мірі створити необхідний ресурс. В тому числі є можливість генерувати основу програмних кодів.

При цьому обирати засоби реалізації необхідно з урахуванням вимог до компонентів, визначених на етапі проектування. Такі вимоги описуються за допомогою інтерфейсів а різні засоби продукування ресурсів забезпечують підтримку інтерфейсів за допомогою загальноприйнятих стандартів.

Є можливість описати весь ресурс на рівні окремих компонент з інтерфейсами між ними. Подальшим проектуванням та розробкою окремих компонент можуть займатися різні спеціалісти, але за умови використання стандартизованих засобів забезпечується сумісність компонентів та бажана функціональність всього ресурсу.

Було проведено аналіз найбільш популярних засобів створення інформаційних сховищ і інформаційних колекторів. Результати аналізу, представлені в табличному вигляді, дозволяють виходячи з вимог до конкретного ресурсу обрати найкращі засоби його створення з функціональної, технічної та економічної точки зору.

## РОЗДІЛ 4. ЕТАПИ ТА ПРАКТИЧНЕ ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ ПРОДУКУВАННЯ ІНФОРМАЦІЙНИХ РЕСУРСІВ

### 4.1 Етапи продукування інформаційного ресурсу

Процес продукування інформаційного ресурсу потребує реалізації декількох стадій [70]:

1. Проведення бізнес-моделювання Орг.С.
2. Розробка моделі ІКС.
3. Формалізація вимог до інформаційного ресурсу.
4. Проектування ресурсу як системи компонентів (менших ресурсів).
5. Проектування кожного окремого компоненту.
6. Оцінка та оптимізація проекту.
7. Реалізація компонентів ресурсу.
8. Інтеграція компонентів.
9. Впровадження системи.

Перші дві стадії відносяться до проектування всієї ІКС. Природно, що ресурс повинен виконувати певні задачі в межах ІКС, тому створення його без урахування вимог до ІКС не доречно. Сам по собі ресурс забезпечує лише обмежені засоби обробки інформації, а тому так чи інакше потребує інтеграції у ІКС. Тому у випадках, коли ресурси розробляються окремо, вони рідко можуть бути ефективно використані.

Формалізація вимог до інформаційного ресурсу передбачає опис на певній формальній мові вимог до ресурсу. Такі вимоги можна поділити на функціональні, технічні та економічні. Вони включають опис задач, характеристики продуктивності, надійності, масштабованості, розширюваності, вартість реалізації та підтримки [30].

Проектування ресурсу як системи компонентів передбачає визначення набору та типів ресурсів, які будуть виконувати окремі функції в межах ресурсу

ІКС. Необхідно описати їх взаємодію як з іншими компонентами ІКС так і з зовнішніми інформаційними ресурсами. Для такого моделювання доречно використати графічну мову моделювання UML та відповідні програмні засоби [4].

Проектування кожного окремого компоненту може проводитись окремими спеціалістами. Основою для такого проектування є специфікації, розроблені на попередніх етапах. Для його автоматизації можна використати засоби автоматизованого проектування відповідних типів ресурсів.

Побудова моделей як ресурсу в цілому так і його компонентів дозволяє розрахувати значення показників ресурсу до початку його реалізації. У випадку, якщо характеристики спроектованого ресурсу не відповідають вимогам, поставленим на етапі формалізації вимог, необхідно провести оптимізацію проекту, яка передбачає певні зміни на різних рівнях, від рівня найменших компонент до рівня моделі ІКС в цілому. Використання графічних мов моделювання дозволяє за допомогою певних надстроек до систем моделювання одразу після внесення змін у моделі оцінювати нові значення характеристик. Відзначимо, що процес оптимізації може проводитись не лише з метою досягнення необхідних показників а і з метою зниження витрат на проект [70]. Оцінку та оптимізацію також доречно проводити і після моделювання ІКС в цілому, і після моделювання ресурсу. Однак без моделювання окремих компонентів не можливо точно передбачити значення параметрів компонентів системи і тому така оцінка є неточною а оптимізація не завжди ефективна. Виконання оптимізації після створення, а тим більше інтеграції та впровадження, неминуче пов'язане з певними матеріальними витратами а тому сильно обмежене. Хоча інколи навіть на такій пізній стадії оптимізація є необхідною і проводиться.

Реалізація компонентів ресурсу являє собою придбання апаратних і програмних засобів та створення на їх основі у відповідності до розроблених моделей окремих компонентів ресурсу. Сучасні системи графічного моделювання дозволяють за моделлю провести генерацію певної частини коду програмних засобів, що створюються.

Інтеграція компонентів у випадку добре проведеного проектування не викликає проблем і зводиться до налаштування, тестування та часткових виправлень у розроблених компонентах.

При процесі впровадження можуть бути знайдені недоліки та помилки, допущені на попередніх етапах. Тому процес створення ресурсу є ітеративним.

Процес продукування інформаційного ресурсу як компоненти потребує реалізації декількох етапів [70, 71 72]:

1. Формалізація задачі продукування інформаційного ресурсу;
2. Визначення параметрів ресурсу;
3. Інтеграція інформаційного ресурсу;
4. Захист інформаційного ресурсу;
5. Визначення засобів продукування інформаційного ресурсу;
6. Системна інтеграція технології продукування інформаційного ресурсу.

На першому етапі вирішується задача формалізації створення того чи іншого ресурсу. Слід пам'ятати, що створення інформаційного ресурсу заради самого ресурсу не має сенсу. Інформаційний ресурс повинен дозволяти вирішувати певні задачі (існуючі або потенціальні). Знаючи задачі, що ставляться перед ресурсом, можна визначити ряд вимог до ресурсу. На даному етапі проводиться формальний опис таких вимог. Формальний опис представляє собою опис на певній мові вимог до даного ресурсу – від мети його створення до засобів реалізації. Для формалізації може бути використана структурована природна мова, умови, характеристики, моделі, правила тощо.

На другому етапі визначаються конкретні параметри необхідного ресурсу. Серед них можуть бути спосіб та порядок доступу, характеристики інформації (даних, знань), що зберігаються, об'єм даних, структура ресурсу, тощо.

Важливим питанням є інтеграція інформаційного ресурсу з іншими ресурсами та з інформаційною системою користувача. Зрозуміло, що в переважній більшості випадків неможливо створити ізольований інформаційний ресурс. Так чи інакше для вирішення поставлених завдань інформаційний ресурс

повинен взаємодіяти з іншими ресурсами. Інтеграція інформаційних ресурсів може проходити на трьох рівнях:

- рівень файлів
- рівень операційного середовища
- рівень інформаційно-телекомунікаційних систем.

Інтеграція інформаційних ресурсів може бути досягнута за рахунок визначення єдиних стандартів інтеграції ресурсів. Прикладом такого стандарту може бути клас стандартів OSI (Open Systems Interconnection Reference Model).

Дуже важливим на сьогодні є задача забезпечення захисту інформаційного ресурсу, тому що у інформаційному суспільстві знання, інформація, дані стають найбільшою цінністю. Все це призводить до необхідності урахування питань безпеки при розробці ресурсу на абонентському каналному і мережених рівнях [73].

Для цього використовуються організаційні і технічні методи та засоби захисту. На абонентському рівні застосовуються системи розмежування та розподілу прав доступу до засобів доступу інформаційно-комунікаційної мережі. На каналному рівні застосовуються методи сегментації та логічної структуризації інформаційних потоків та відповідні технічні засоби, як то мережеві маршрутизатори та мережеві екрани для локальних мереж. На мережевому рівні застосовуються міжмережеві екрани та криптографічні методи та засоби захисту від несанкціонованого доступу. Для забезпечення адекватного цінності інформаційного ресурсу захисту інформації пропонується застосування системи управління безпекою організаційної структури, яка об'єднує методи і засоби захисту в єдину систему (Security Manager) [14].

За визначеними на попередніх етапах відомостями обираються засоби створення інформаційного ресурсу. Варто зазначити, що створення комп'ютеризованої інформаційно-комунікаційної системи не завжди є виправданим. Якщо приймається рішення про доцільність створення саме комп'ютеризованого ресурсу, то для його створення (або для створення окремих

його частин) можливе використання існуючих технологій автоматизації розробки баз даних, інформаційних сховищ та інформаційних колекторів.

Продуктування інформаційного ресурсу являє собою циклічний процес, як показано на рис. 4.1. Інформаційний ресурс повинен постійно розвиватися. Після реалізації ресурсу виникають нові задачі, які призводять до необхідності нового циклу вдосконалення. Крім зазначеного порядку роботи можливий (і природний) зворотній шлях – повернення на попередні етапи продуктування інформаційного ресурсу.

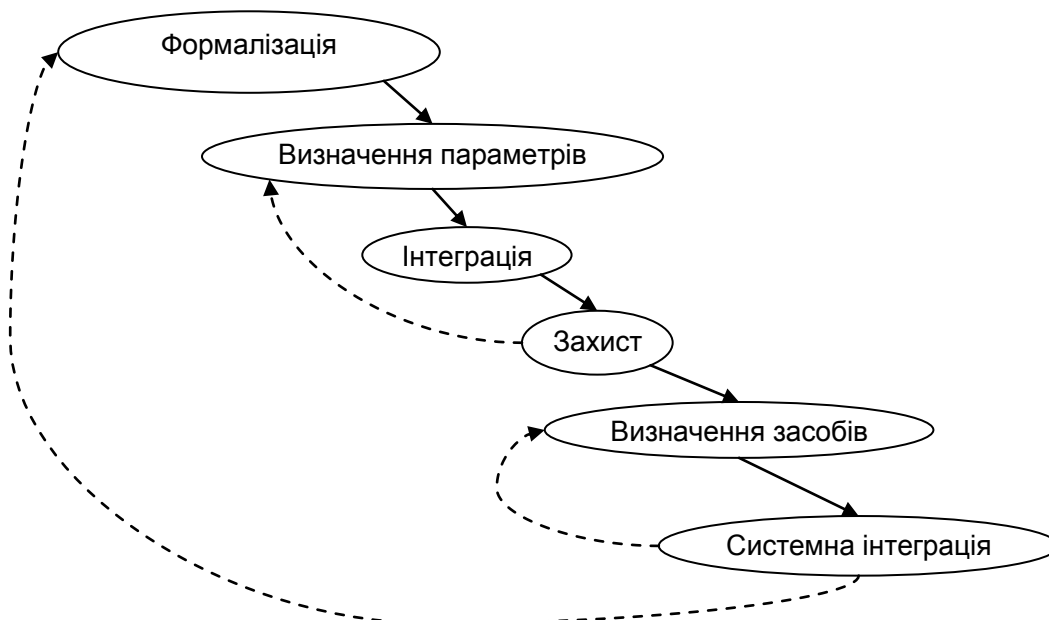


Рисунок 4.1 – Етапи продуктування інформаційного ресурсу

З поширенням інформаційних технологій у всіх сферах життєдіяльності людини все більшого значення набуває створення ефективних засобів доступу до інформації, а також інтелектуальних систем генерації нових знань, інформації, даних. Рівень розвитку інформаційних ресурсів визначає рівень конкурентоспроможності організаційної структури на ринку. Тому проблема продуктування інформаційних ресурсів є актуальною на сьогодні.

Нашим завданням є розробка єдиних принципів, методів і рішень для розробки технології продуктування інформаційного ресурсу. На їх основі можна розробити єдиний універсальний підхід до створення ресурсів, який дозволить



приймати обґрунтовані рішення та обирати оптимальні засоби та методики для продукування інформаційних ресурсів.

Співвідношення стадій проектування ресурсу, етапів проектування окремого ресурсу та стадій моделювання [70, 74] наведено у табл. 4.1.

Таблиця 4.1 – Співвідношення стадій продукування ресурсу, етапів продукування та стадій моделювання

Стадії продукування ресурсу	Етапи проектування ресурсів як компонентів	Стадії моделювання
Проведення бізнес-моделювання Орг.С		Аналіз, проектування
Розробка моделі ІКС		Аналіз, проектування
Формалізація вимог до інформаційного ресурсу		Аналіз, проектування
Проектування ресурсу як системи компонентів (менших ресурсів)	Формалізація Визначення параметрів Інтеграція	Проектування
Проектування кожного окремого компоненту	Захист Визначення засобів продукування	Аналіз, проектування
Оцінка та оптимізація проекту		Проектування
Реалізація компонентів ресурсу	Системна інтеграція	Проектування, реалізація
Інтеграція компонентів	Системна інтеграція	Реалізація
Впровадження системи		Розгортання

Таким чином, процес створення окремого ресурсу проводиться після і базується на проектуванні всієї ІКС. При добре проведеному проектуванні розробка окремих інформаційних ресурсів закінчується до впровадження системи, і на стадії впровадження подальша доробка добре спроектованого і розробленого ресурсу не потрібна.

## 4.2 Застосування технології продукування інформаційного ресурсу для Державної податкової адміністрації України

Застосування описаної технології продемонструємо на прикладі побудови ресурсу для Державної податкової адміністрації України (ДПА) [37]. Інформаційно-комунікаційна система ДПА повинна забезпечити облік платників податків, їх доходів та виплат, забезпечення процесів обробки податкових документів, забезпечення прийому та обліку коштів, забезпечити інформаційні сервіси, такі як базу документів нормативно-правового характеру, надати інструменти аналітичної обробки інформації. Доступ до системи має забезпечуватись із локальної мережі ДПА а також ззовні (через Інтернет, як для співробітників ДПА так і для громадян). При цьому необхідно забезпечити відповідний рівень безпеки.

Модель ІКС, отримана на етапі проектування, показана на рис. 4.2.

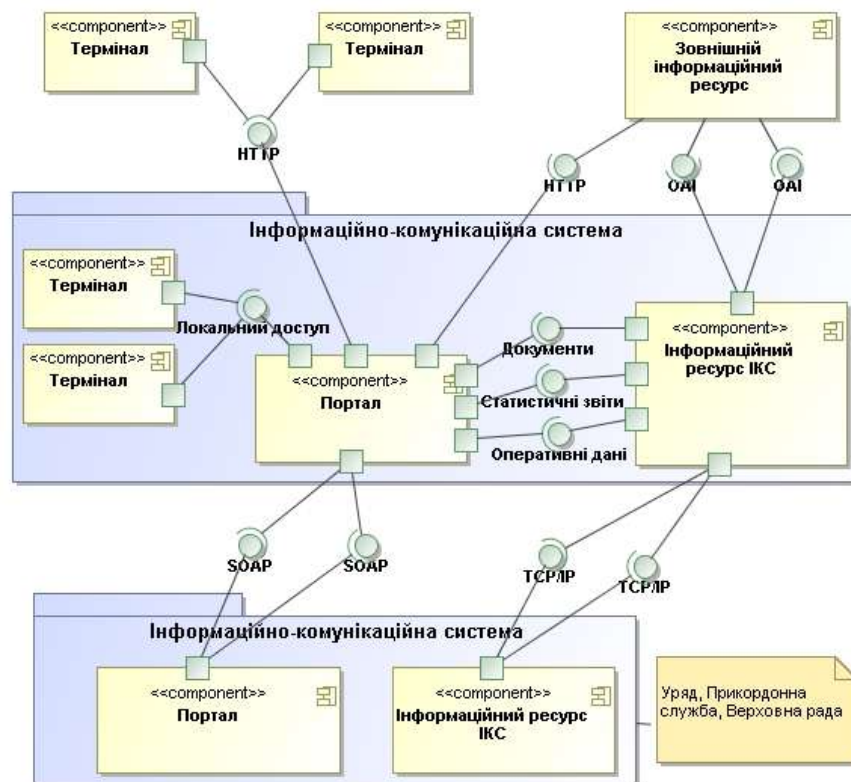


Рисунок 4.2 – Модель ІКС Державної податкової адміністрації України. Діаграма компонентів в нотації UML.

Портал забезпечує авторизацію та забезпечення роботи користувачів, доступ до ресурсів. Для платників податків забезпечується неавторизована робота через зовнішній портал, основною задачею якого є інформування громадян, відповідь на їх запитання, представлення шаблонів документів, тощо.

Інформаційний ресурс має забезпечити доступ до оперативних даних, статистичних звітів та документів. Необхідно забезпечити інтеграцію ресурсу з двома групами інформаційних ресурсів: ресурсами інших органів державної влади та публічних ресурсів.

Для опису вимог до системи використовується метрика. Метрика – це необхідна і достатня множина характеристик атрибутів і операцій, призначених для проектування та експлуатації ІКС. Для кожного конкретного класу ІКС та її реалізації метрика набуває конкретних значень характеристик.

Метрика використовується як розробниками ресурсу для контролю та покращення якості так і замовником при оцінці представленого рішення.

Загальноприйнятої метрики для інформаційних ресурсів немає. Це пов'язано з рядом проблем по визначенню показників інформаційних ресурсів. Однак розробка такої системи показників є необхідною для створення системи управління та обліку ресурсів [75].

Тому ми пропонуємо включити в метрику найбільш розповсюджені та необхідні для проектування характеристики, які описують атрибути та операції окремих компонентів та/або всього ресурсу.

До таких характеристик насамперед слід віднести:

- Характеристики продуктивності функціонування (пропускну спроможність, максимально допустимий час відповіді на запит до оперативних даних, аналітичних звітів та документів, час затримки).
- Характеристики розширюваності та масштабованості
- Об'єм даних, що зберігаються
- Характеристики безпеки інформаційного ресурсу

Максимально допустимий час відповіді на запит визначає зручність роботи у системі. Оперативні дані включають дані про користувачів системи, реєстр

платників податків, тощо. Для отримання такої інформації використовуються прості запити до бази даних. Оперативна інформація необхідна практична для кожної дії користувача. Допустимий час її отримання не має перевищувати 1с. Документи та аналітичні звіти необхідні не так часто. Крім того така інформація представляє більшу “значимість” для користувача і вимагає певної обробки користувачем, тому тут допустимий більший час відповіді.

Необхідно зазначити, що час відповіді залежить від кількості одночасно підключених користувачів (терміналів), від якої залежить кількість одночасних запитів. Тому при оцінці часових параметрів необхідно враховувати навантаження на систему. При цьому варто врахувати сезонні навантаження на систему, пов’язані з визначеними законодавством періодами подання податкових документів та загальну кількість платників податків. Так у Києві кількість платників податків складає 308 тисяч фізичних і юридичних осіб.

Стійкість та надійність визначається імовірністю відмови, наявністю резервування та часом відновлення системи після збою. Для ДПА допускається тимчасова недоступність сервісів і не допускається втрата інформації.

Модель інформаційного ресурсу представлено на рис. 4.3.

Для того, щоб забезпечити одночасну обробку запитів від великої кількості операторів інформаційний ресурс реалізується у вигляді розподіленої системи. Для кожного регіонального відділення податкової адміністрації встановлюється окрема система апаратних і програмних засобів. З точки зору ресурсів до такої регіональної системи відноситься регіональна база даних. Регіональна БД зберігає основну інформацію, необхідну для роботи системи в регіонах. Для більшості запитів звернення до центрального серверу не є необхідним.

Центральний сервер зберігає дані та інформацію, необхідні для роботи системи в цілому. До них відноситься центральна база даних, інформаційний колектор та інформаційне сховище. Інформаційне сховище призначено для збору аналітичної інформації. Наповнення сховища відбувається при заповненні даних у регіональних та центральній БД.

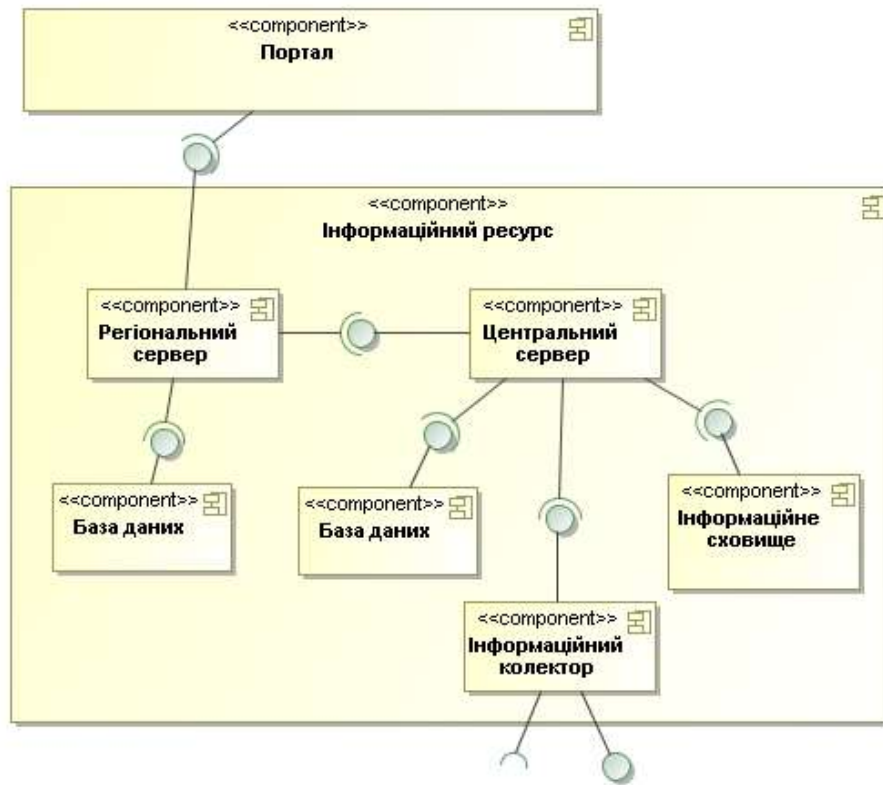


Рисунок 4.3 – Модель інформаційного ресурсу ДПА. Діаграма компонентів у нотації UML.

Інформаційний колектор призначено для зберігання нормативних актів, законів, правил, інструкцій та інших документів, на яких базується робота ДПА та які є корисними для користувачів. Колектор забезпечує зберігання інформації ДПА та доступ до інформації в колекторах інших засобів державної влади.

Кожен з компонентів деталізується при подальшому моделюванні. На апаратному рівні як регіональний, так і центральний сервер включає сервер застосувань, сервер баз даних та проксі-сервер. Така система забезпечує необхідний рівень надійності та безпеки.

## ВИСНОВКИ ДО РОЗДІЛУ 4

Застосування прикладного системно аналізу та компонентного підходу до проектування інформаційних ресурсів ІКС дозволяє упорядкувати і суттєво спростити процес проектування інформаційного ресурсу, врахувати конкретні вимоги, провести оптимізацію структурного та динамічного представлення інформаційного ресурсу, закласти необхідні рішення у відповідності із іменованою специфікацією на інформаційний ресурс [70].

Застосування для проектування графічної мови моделювання UML та засобів автоматизації процесу проектування забезпечує автоматизовану генерацію програмного забезпечення інформаційного ресурсу та тестування на кожній ітерації та фазі розробки.

Таким чином, компонентний підхід дозволяє створити автоматизовану систему проектування ІКС. Складовою такої системи є підсистема автоматизації проектування інформаційного ресурсу, яка дозволяє спроектувати та продукувати ресурс як окремий компонент ІКС.

Специфікація вимог до ресурсу за допомогою визначення інтерфейсів дозволяє розділити процес розробки ресурсу на незалежні компоненти, що можуть бути спроектовані і розроблені окремо. Визначені вимоги дозволять обґрунтовано обрати засоби створення окремих компонентів ресурсу. Використання сучасних стандартів взаємодії забезпечує надійне поєднання розроблених окремо компонентів на стадіях реалізації та розгортання.

## РОЗДІЛ 5. ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ПРОЕКТУ. ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

### 5.1 Постановка задачі техніко-економічного дослідження

Розробка програмного забезпечення допускає багато різних варіантів реалізації, які включають різну функціональність продукту. Тому постає задача вибору такого варіанту, який забезпечить необхідну замовнику функціональність при мінімальних затратах на реалізацію. Такий вибір проводиться на основі техніко-економічного дослідження.

Ефективним методом техніко-економічного дослідження є Функціонально-вартісний аналіз (ФВА) [76, 77]. В основу ФВА покладено функціональний підхід, при якому об'єктом аналізу є не сам виріб (у даному випадку програмний продукт), а основні та допоміжні функції, які він виконує. Метод ФСА визначає алгоритм, за яким проводиться пошук можливостей зниження витрат на розробку, виготовлення та експлуатацію програмного продукту із забезпеченням необхідної якості. Таким чином ФСА дозволяє обгрунтовано обрати найкраще за визначеними критеріями інженерне рішення для реалізації програмного продукту.

У роботі описано проблеми продукування інформаційних ресурсів, зокрема інформаційних колекторів. У даному розділі буде проведено оцінку різних варіантів реалізації функцій інформаційного колектора та вибір найкращого варіанта для реалізації архіву КПІ [78, 79]. Метод ФВА дозволить обрати найбільш раціональний з функціональної точки зору варіант реалізації.

## 5.2 Аналіз виявлених варіантів ПП

### 5.2.1 Формування функцій ПП

Основна функція (призначення інформаційного колектору):

Інформаційний колектор (бібліотечний архів) призначено для зберігання та первинної обробки документів в цифровому виді з забезпеченням можливості доступу до них користувачів.

Допоміжні функції:

F1. Зберігання документів різного типу та формату.

F2. Підтримка метаданих.

F3. Забезпечення веб-доступу.

F4. Розміщення системи

### 5.2.2 Варіанти реалізації функцій

F1. а) Зберігання текстових документів.

б) Зберігання текстових, аудіо та відео документів.

с) Зберігання інформації в довільному форматі

F2. а) Не підтримуються.

б) Метадані в форматі Dublin Core.

с) Метадані в форматі MARC.

д) Довільна схема.

F3. а) Визначені інтерфейси доступу до інформації.

б) Вбудована інтеграція у портал користувача.

F4. а) На обладнанні користувача

б) На обладнанні розробника



Позитивно-негативна матриця варіантів реалізації функцій (табл. Таблиця 5.1) відображає переваги і недоліки варіантів реалізації функцій і використовується при відборі варіантів, які максимально задовольняють задачі системи [80].

Таблиця 5.1 – Позитивно-негативна матриця

Функції	Варіанти	Переваги	Недоліки
F1	a	Простота реалізації. Простота обробки.	Обмеженість контенту.
	b	Забезпечення необхідних користувачу форматів документів.	Складність розробки. Великий об'єм документів.
	c	Забезпечування розвитку в майбутньому	Складність реалізації. Складність та повільність обробки.
F2	a	Простота реалізації	Неефективність індексації. Неможливість інтеграції з іншими системами.
	b	Ефективна індексація.	Інтеграція тільки з системами із стандартом метаданих Dublin Core.
	c	Ефективна індексація.	Інтеграція тільки з системами із стандартом метаданих MARC.
	d	Можливість інтеграції з усіма системами.	Складність реалізації.

Продовження таблиці Таблиця 5.1 – Позитивно-негативна матриця

Функції	Варіанти	Переваги	Недоліки
F3	a	Простота реалізації самого колектору.	Складність реалізації засобів доступу.
	b	Простота реалізації системи.	Висока вартість інструментів інтеграції.
F4	a	Повний контроль над інформацією	Необхідність найму спеціалістів для розробки та підтримки
	b	Повна кваліфікована підтримка та сервіс	Необхідність щорічної оплати.

### 5.2.3 Морфологічна карта

Проаналізуємо значення в матриці для виключення частини варіантів реалізації:

#### *Функція F1:*

Оскільки необхідно зберігати інформацію у текстовому, графічному, аудіо та відео форматі, то будемо використовувати варіанти b та c. Так як цих форматів на сьогодні та передбачуване майбутнє достатньо для представлення інформації, будемо враховувати лише варіант b.

#### *Функція F2:*

Оскільки формат Dublin Core став фактично єдиним загальноприйнятим стандартом метаданих, інші стандарти розглядати недоречно. Будемо використовувати варіанти a та b.

#### *Функція F3:*

Для забезпечення меншої вартості продукту та за наявності спеціалістів будемо використовувати варіант a.

*Функція F4:*

Так як необхідно забезпечити конфіденційність та керованість інформації, а технічна база КПІ дозволяє розгорнути систему, оберемо для реалізації варіант а.

Таким чином, будемо розглядати 2 варіанти реалізації ПП:

- 1) F1b - F2a - F3a - F4a;
- 2) F1b - F2b - F3a - F4a.

Морфологічна карта відбиває всі можливі комбінації варіантів реалізації функцій, які становлять повну множину варіантів програмного продукту [77]. Морфологічну карту даного ПП наведено на рис. Рисунок 5.1.

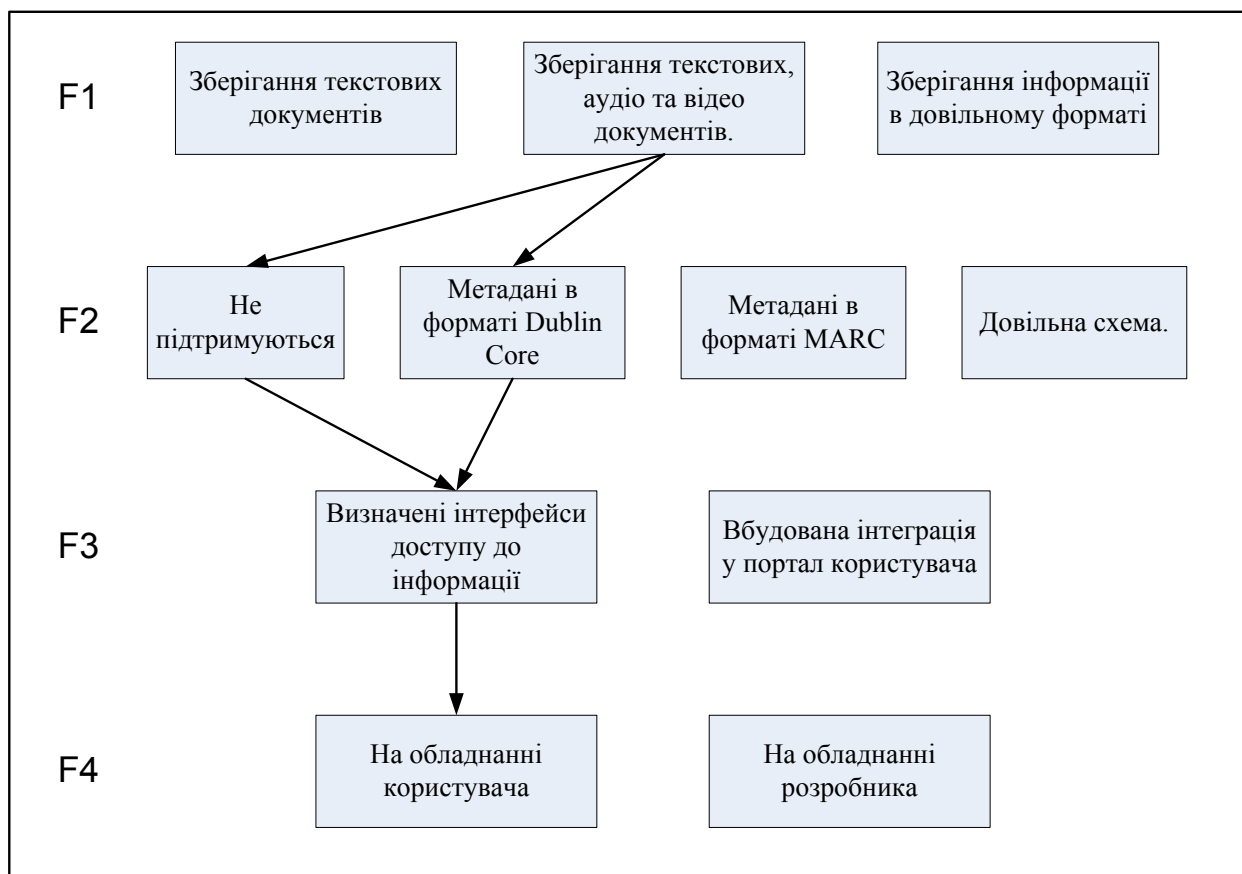


Рисунок 5.1 – Морфологічна карта

### 5.3 Обґрунтування системи параметрів

#### 5.3.1 Опис параметрів

Для оцінки варіантів реалізації програмного продукту будемо застосувати наступні параметри:

- X1 – можливості інтеграції з іншими ресурсами;
- X2 – релевантність інформації за пошуковим запитом;
- X3 – складність ПП;
- X4 – швидкість доступу до інформації;
- X5 – зручність використання.

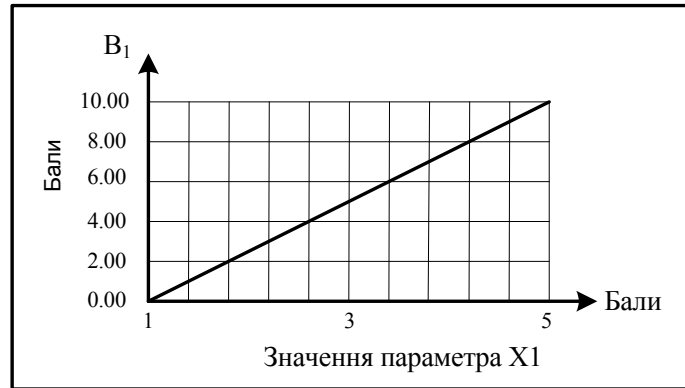
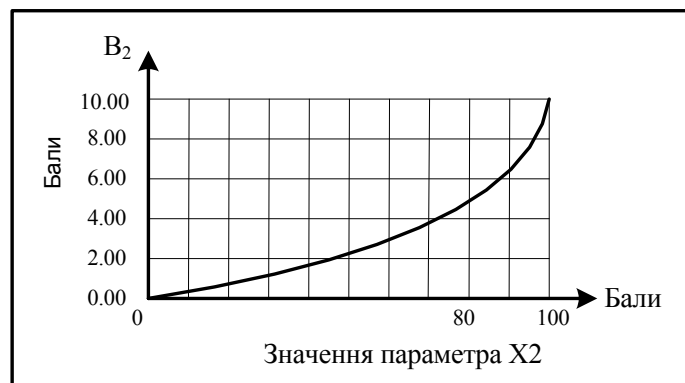
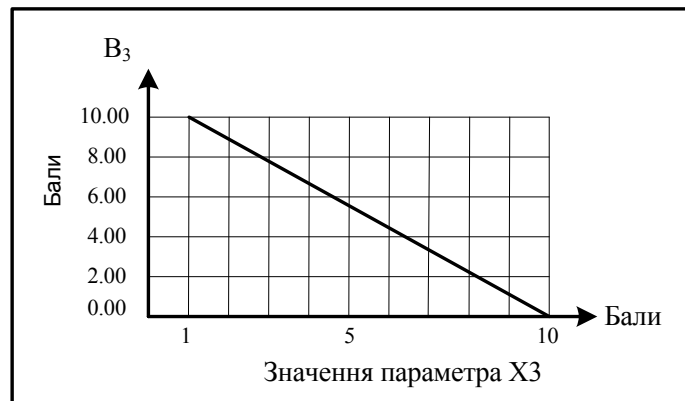
#### 5.3.2 Кількісна оцінка параметрів

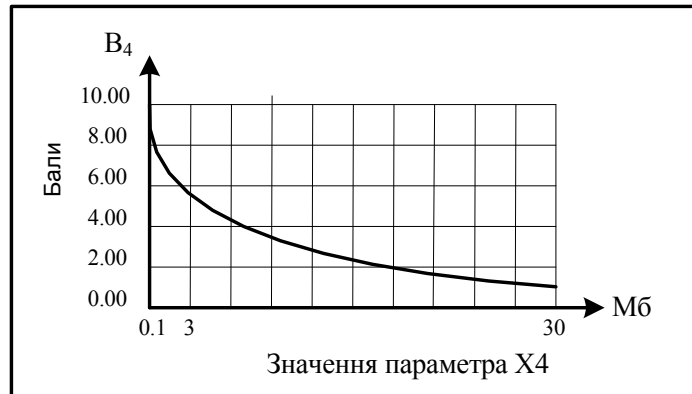
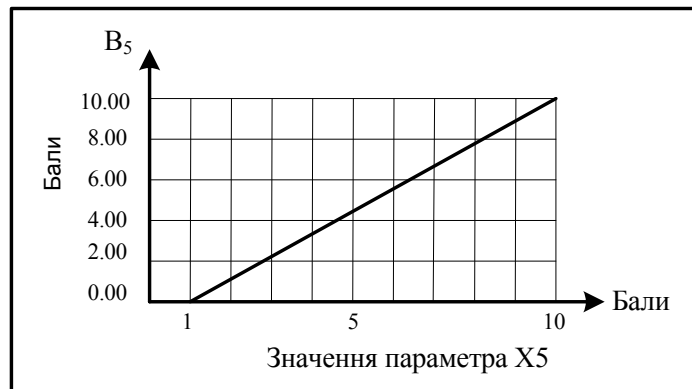
За даними технічної літератури та досвідом попередніх розробок визначаємо значення параметрів. Результати наведені в таблиці Таблиця 5.2.

Таблиця 5.2 – Основні параметри програми.

Назва параметра	Умовне позначення	Значення параметра		
		Гірше	Середнє	Краще
Показник можливості інтеграції з іншими ресурсами	X1	1	3	5
Релевантність інформації за пошуковим запитом, бали	X2	0	80	100
Складність ПП, бали	X3	10	5	1
Швидкість доступу до інформації, с	X4	30	3	0.1
Показник рівня зручності використання ПП, бали	X5	1	5	10

Виходячи з даних табл. Таблиця 5.2 побудовані графічні характеристики параметрів (рис. 5.2 - 5.3).

Рисунок 5.4– Оцінка 1-го параметра,  $V_1$ Рисунок 5.5– Оцінка 2-го параметра,  $V_2$ Рисунок 5.6 – Оцінка 3-го параметра,  $V_3$

Рисунок 5.7– Оцінка 4-го параметра,  $V_4$ Рисунок 5.8– Оцінка 5-го параметра,  $V_5$ 

### 5.3.3 Визначення коефіцієнтів вагомості параметрів

Для визначення коефіцієнтів вагомості кожного з показників ПП, наведених у табл. 5.6., проведемо експертне порівняння пар параметрів. У порівнянні будуть брати участь 7 експертів. Результати порівняння зводяться до табл. 5.3. Спершу визначимо ступінь важливості параметрів шляхом присвоєння їм рангів і перевірки придатності експертних оцінок для подальшого використання.

Кожний експерт оцінює ступінь важливості параметрів, вказуючи їх ранги. Результати експертного ранжування представлені в табл. 5.3.

Таблиця 5.3– Результат ранжування параметрів.

Параметр	Ранг параметра по оцінці експерта							Сума рангів $R_i$	Відхилення, $\Delta_i$	$\Delta_i^2$
	1	2	3	4	5	6	7			
X1	5	4	4	5	5	5	4	32	11	121
X2	1	2	1	2	1	1	1	9	-12	144
X3	4	5	5	4	3	4	3	27	7	49
X4	2	1	3	3	4	3	5	21	0	0
X5	3	3	2	1	2	2	2	16	-6	36
	15	15	15	15	15	15	15	105	0	350

#### 5.3.4 Перевірка ступеня придатності експертних оцінок

Визначення можливості використання результатів ранжування показників для наступних розрахунків проводимо на підставі розрахунку коефіцієнта конкордації експертних оцінок.

1) Визначаємо суму рангів кожного показника:  $R_i = \sum_{j=1}^N r_{ij}$ ;  $i = \overline{1, n}$ ;  $j = \overline{1, N}$ ,

де  $r_{ij}$  - ранг  $i$ -го показника у  $j$ -го експерта;  $N$  – число експертів.

Загальна сума рангів має дорівнювати  $R_{ij} = \frac{Nn(n+1)}{2} = 105$ .

2) Визначаємо середню суму рангів по формулі:  $T = \frac{1}{n} \cdot R_{ij} = 21$ .

3) Визначаємо відхилення суми рангів кожного показника ( $R_i$ ) від середньої суми ( $T$ ):  $\Delta_i = R_i - T$ .

Сума відхилень за всіма показниками повинна дорівнювати 0.

4) Розраховуємо квадрат відхилень за кожним показником:

$$S = \sum_{i=1}^n \Delta_i^2 = 314.$$

5) Коефіцієнт конкордації визначаємо за формулою:

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 350}{49 \cdot (125 - 5)} = 0.714$$





### 5.3.6 Розрахунок пріоритету параметрів

У табл. 5.5 наведено розрахунки коефіцієнтів важливості кожного показника.

Таблиця 5.5– Розрахунок вагомості параметрів ПП.

	Параметри					1-а ітерація		2-а ітерація	
	X1	X2	X3	X4	X5	$b_i$	$\varphi_i$	$b_i$	$\varphi_i$
<b>X1</b>	1.0	1.5	1.5	1.5	1.5	7	0.28	34	0.296
<b>X2</b>	0.5	1.0	0.5	0.5	0.5	3	0.12	14	0.121
<b>X3</b>	0.5	1.5	1.0	1.5	1.5	6	0.24	27.5	0.239
<b>X4</b>	0.5	1.5	0.5	1.0	1.5	5	0.2	22	0.191
<b>X5</b>	0.5	1.5	0.5	0.5	1.0	4	0.16	17.5	0.154
<b>Усього:</b>						25	1.00	115	1.001

Розрахунки виконані за формулами:

$$b_i = \sum_{j=1}^n a_{ij}, \quad \varphi_i = \frac{b_i}{\sum_{i=1}^n b_i}, \quad b'_i = \sum_{j=1}^n a_{ij} b_j, \quad \varphi'_i = \frac{b'_i}{\sum_{i=1}^n b'_i}$$

Розбіжність між першою та другою ітерацією не перевищує 6%, тому двох ітерацій достатньо.

Таким чином, коефіцієнти важливості параметрів:

$$\varphi_1 = 0.296;$$

$$\varphi_2 = 0.121;$$

$$\varphi_3 = 0.239;$$

$$\varphi_4 = 0.191;$$

$$\varphi_5 = 0.154.$$

### 5.3.7 Визначення рівня якості

Коефіцієнт технічного рівня для кожного варіанта рішення будемо розраховувати за формулою [77]:

$$K_{TP} = \sum_i \varphi_i \cdot B_i, i = 1..n,$$

де n - кількість параметрів;

$\varphi_i$  – коефіцієнт важливості і-го параметра;

$B_i$  – оцінка і-го параметра в балах.

Як було визначено на етапі аналізу варіантів реалізації функцій продукту, будемо розглядати 2 варіанти реалізації:

1) F1b - F2a - F3a - F4a;

2) F1b - F2b - F3a - F4a.

Результати розрахунку наведені у табл. 5.6.

Таблиця 5.6 – Розрахунок показників якості варіантів реалізації функцій

ПП.

Показник	$\varphi_i$	Варіант рішення			
		1		2	
		Абс	Бал	Абс	Бал
1	0.296	2	2.5	4	7.5
2	0.121	50	2.25	90	6.25
3	0.239	4	6.5	8	2
4	0.191	10	2.4	1	8
5	0.154	5	5	5	5
$K_{TP}$		3.794		5.572	

Таким чином, найкращий показник технічного рівня має варіант 2 реалізації програмного продукту.

## 5.4 Економічний аналіз варіантів ПП

### 5.4.1 Визначення трудомісткості розробки програмного продукту

Вихідні дані:

- кількість макетів – 3;
- кількість різновидів форм вихідної інформації – 5;
- ступінь новизни групи завдань – В;
- складність алгоритму – 2;
- складність організації контролю вхідної і вихідної інформації 12 і 22 відповідно;
- вид використовуваної інформації: змінної – 1, нормативно-довідкової – 1, БД – 3;
- мова програмування високого рівня – С#;
- використовуються стандартні модулі.

На основі вихідних даних при кількості макетів вхідної інформації (3 макети), різноманітності форм вихідної інформації (5 форм) і на основі даних типових норм часу на програмування визначаємо норму часу  $T_p$ , що дорівнює 175 людино-днів:

$T_p = 175$  людино-днів.

Розрахуємо поправочний коефіцієнт  $K_p$  по формулі

$$K_p = \frac{K_1 * m + K_2 * n + K_3 * p}{m + n + p}$$

де  $K_1$ ,  $K_2$ ,  $K_3$  – поправочні коефіцієнти;

$m$ ,  $n$ ,  $p$  – відповідно коефіцієнти змінної, нормативно-довідкової інформації, та інформації від БД.

Коефіцієнти  $K_1$ ,  $K_2$  і  $K_3$  визначені за таблицею згідно обсягу видів використаної інформації (змінної – 2; нормативно-довідкової – 1; від БД – 3) і групи складності алгоритму (друга група);

вони дорівнюють відповідно:

$$K1 = 1.2,$$

$$K2 = 0.72,$$

$$K3 = 0.6.$$

Поправочний коефіцієнт:

$$K_{\Pi} = \frac{K1 \cdot m + K2 \cdot n + K3 \cdot p}{m + n + p} = \frac{1.2 \cdot 2 + 0.71 \cdot 1 + 0.6 \cdot 3}{2 + 1 + 3} = \frac{4.92}{6} = 0.82$$

Розраховуємо поправочний коефіцієнт  $K_{СК}$ , що характеризує складність контролю вхідної та вихідної інформації. Складність контролю вхідної інформації – 12, вихідної – 22.

$$K_{СК} = 1.00$$

ПП розробляється мовою програмування високого рівня, тому коефіцієнт  $K_{М}$  не враховується.

При розробці ПП використовується невелика кількість стандартних модулів. Тому:

$$K_{СТ} = 0.8$$

Отже, загальна трудомісткість ПП становить

$$T_0 = T_P \cdot K_{\Pi} \cdot K_{СК} \cdot K_{СТ} = 175 \cdot 0.81 \cdot 1 \cdot 0.8 = 115 \text{ людино-днів.}$$

Для варіанту реалізації 1 кількість макетів вхідної інформації становить 2, тому

$$T_P = 135 \text{ людино-днів.}$$

$$T_0 = T_P \cdot K_{\Pi} \cdot K_{СК} \cdot K_{СТ} = 135 \cdot 0.81 \cdot 1 \cdot 0.8 = 89 \text{ людино-днів.}$$

#### 5.4.2 Визначення витрат на розробку і розрахунок вартості ПП

Для розробки ПП задіяно 2 програмісти, яких зайнято на 65 людино-днів першого і на 50 другого. Місячний посадовий оклад програміста дорівнює 4520 грн., тривалість умовного місяця 21,1 днів ( 5-ти денний робочий тиждень) [76].

Результати розрахунків основної заробітної плати наведено в табл. 5.7.

Таблиця 5.7 – Основна заробітна плата.

	Місячний оклад, грн.	Денна заробітна плата, грн.	Трудомісткість, людино-дні	Основна заробітна плата, грн.
1-й програміст	4520	214,22	65	13924.3
2-й програміст	4250	214.22	50	10711.0
Усього				24635.3

Для розробки варіанту 1 необхідно 90 людино-годин, тобто 50 для першого і 40 для другого, тому основна заробітна плата складатиме 19199.8грн.

Додаткова з/п видається всім учасникам проекту у вигляді премії в розмірі 25% від основної з/п.

Для створення ПП будемо використовувати 2 ЕОМ, які будуть працювати 5 днів у тиждень по 8 годин, час на ремонт становить 15% від загального часу роботи. Тоді ефективний годинний фонд роботи ПК за рік:

$$T_{эф} = (365 - 11 - 52 \cdot 2) \cdot 1 \cdot 8 \cdot \left(1 - \frac{15}{100}\right) = 1500 \text{ годин.}$$

Машину буде обслуговувати фахівець із з/п 600 грн/міс. Додаткова з/п за рік становить 1000 грн. Вартість одного ПК 4000 грн., споживана потужність одного ПК 350 Вт. Річні експлуатаційні витрати на обслуговування ПК  $C_{ЭК}$  включають:

- основну і додаткову заробітну плату фахівця, що обслуговує машину, з врахуванням його зайнятості на обслуговування ПК, грн.;
- відрахування на соціальні потреби (37,5% від фонду оплати праці), грн.;

- амортизаційні відрахування (25% відповідно до норми), грн.;
- витрати на електроенергію, які визначаються як добуток тарифу за 1 кВт/год на потужність, споживану ПК і на ефективний годинний фонд роботи ПК за рік, грн.;
- накладні витрати (75% від витрат на оплату праці), грн.

$$C_{\text{ЭКС}} = (600 \cdot 12 + 1000) + (600 \cdot 12 + 1000) \cdot 0.375 + (4000 \cdot 0.25) + (0.2 \cdot 0.35 \cdot 1500) + (600 \cdot 12 + 1000) \cdot 0.75 = 18530 \text{ грн.}$$

Собівартість однієї машинної години роботи ПК:

$$C_{\text{Мч}} = \frac{18530}{1500} = 12.35 \text{ грн./година}$$

Вартість машинного часу визначається по формулі:  $C_M = C_{\text{Мч}} \cdot t_M$ .

Результати розрахунків витрат на оплату машинного часу наведені в табл. 5.8.

Таблиця 5.8 – Витрати на оплату машинного часу

Роботи, які виконуються на ЕОМ	Тривалість виконання робіт по варіантах, годин	
	1	2
1. Реалізація основних функцій	175	185
2. Реалізація допоміжних функцій	152	195
3. Реалізація інтерфейсу	95	120
5. Компонування програми	50	65
6. Налаштування програми	120	145
7. Тестування програми	120	210
Усього годин	712	920
Вартість однієї машино-години, грн.	12.35	12.35
Усього, грн.	8793.2	11362.0

Так як ФВА виконується на етапі розробки програмного продукту, будемо розраховувати собівартість розробки як поточні витрати  $C_{III}$ , які розраховуються по формулі  $C_{III} = C_{зп} + C_{отч} + C_M + C_H$ , де

$C_{зп}$  – заробітна плата розробників ПП, грн.;

$C_{отч}$  – відрахування на соціальні потреби (37,5% від фонду оплати праці), грн.;

$C_M$  – вартість машинного часу, необхідного для розробки й налаштування ПП, грн.;

$C_H$  – накладні витрати, містять у собі витрати на оплату роботи управлінського персоналу, оплату службових відряджень, консультаційно-інформаційні витрати, ремонт і технічне обслуговування інших основних фондів, крім ПК, оренду приміщення та ін.

Накладні витрати становлять 72% від витрат на оплату праці. Кошторис витрат на розробку ПП наведений у табл. 5.9.

Таблиця 5.9 – Кошторис витрат на розробку ПП

Статті витрат	Сума витрат по варіантах реалізації функцій, грн.	
	1	2
1. Основна заробітна плата	19199.8	24635.3
2. Додаткова зарплата	4799.95	6158.825
Фонд оплати праці	23999.75	30794.13
3. Відрахування на соц. заходи, у т.ч.	8999.906	11547.8
на обов'язкове пенсійне страхування,	7679.92	9854.12
на обов'язкове соц. страхування	1267.187	1625.93
4. Експериментально - виробничі витрати	8793.2	11362
5. Накладні витрати	17279.82	22171.77
Сума витрат	50279.48	64513.69

### 5.4.3 Розрахунок показників економічної ефективності кожного з варіантів програмного продукту

Для порівняння ефективності розробки варіантів ПП будемо визначати витрати на одиницю якості по кожному з варіантів:  $K_{TEP} = \frac{K_{TP}}{C_{ПП}}$ . Результати аналізу наведені в табл. 5.10.

Таблиця 5.10 – Порівняння ефективності розробки варіантів ПП

Варіант реалізації	$C_{ПП}$ , грн.	$K_{TP}$	$K_{TEP}$
1	50279.48	3.794	$7.55 \cdot 10^{-5}$
2	64523.69	5.572	$8.64 \cdot 10^{-5}$

Таким чином, найбільш ефективним є другий варіант реалізації функцій ПП, що має максимальну величину коефіцієнта техніко-економічного рівня.



## 5.5 ВИСНОВКИ ДО РОЗДІЛУ 5

У результаті виконання даної роботи було проведено функціонально-вартісний аналіз інформаційного колектору для НТУУ «КПІ».

На етапі аналізу варіантів реалізації ПП було обрано два варіанти, які найкраще відповідали вимогам до програмного продукту. В результаті аналізу вищий показник техніко-економічного рівня отримав варіант 2. Він є важчим для реалізації та дорожчим, однак дозволяє забезпечити максимальну функціональність та зручність для користувачів, тому такі витрати є виправданими.

Обраний варіант буде розроблений за 2.5 місяці. Сума витрат на його розробку складе 64523.69грн. Узагальнений показник технічного рівня програмного продукту дорівнює  $K_{ТР} = 5.572$ , показник техніко-економічного рівня ПП  $K_{ТЕР} = 8.64 \cdot 10^{-5}$ .

## ВИСНОВКИ

Методи прикладного системно аналізу та компонентний підхід до проектування інформаційних ресурсів ІКС дозволяють упорядкувати і суттєво спростити процес проектування інформаційного ресурсу, врахувати конкретні вимоги, провести оптимізацію структурного та динамічного представлення інформаційного ресурсу, закласти необхідні рішення у відповідності із іменованою специфікацією на інформаційний ресурс. Розробка сучасних засобів та удосконалення технологій продукування інформаційних ресурсів дозволить підвищити ефективність їх використання.

В результаті виконання роботи отримані наступні результати.

1. Проведено порівняльний аналіз підходів до вимірювання інформації та визначено межі їх застосування до різних типів інформаційних ресурсів.
2. Розглянуті основні типи інформаційних ресурсів, визначені їх класифікація, характеристики та напрямки застосування.
3. Розглянуті принципи побудови і роботи пошукових систем.
4. Проведено аналіз сучасних засобів та інструментів продукування інформаційних ресурсів.
5. Запропоновано технологію проектування інформаційного ресурсу, що базується на компонентному підході.
6. Реалізація запропонованої технології виконана в проекті інформатизації Державної податкової адміністрації України.
7. Проведено функціонально-вартісний аналіз варіантів реалізації ресурсу.

Результати роботи можуть бути впроваджені у процес розробки інформаційного ресурсу ІКС Орг.С. і спрямовані на спрощення проектування ресурсу, підвищення економічного ефекту від впровадження ІКС.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Згуровський М.З., Панкратова Н.Д. Системний аналіз: проблеми, методологія, застосування. Київ, Наукова думка 2005, 743 с.
2. Маслянюк П.П. Технология информатизации корпоративных структур, ч.1 // Корпоративные системы. - 2003. - №1. - С. 17-19.
3. Маслянюк П.П. Технология информатизации корпоративных структур, ч.2 // Корпоративные системы. - 2003. - №4. - С.17-19.
4. Кватрани Т. Визуальное моделирование с помощью Rational Rose 2002 и UML.: Пер. С англ. – М.: Вильямс, 2003.
5. Згуровський М.З., Родіонов М.К., Жилияев І.Б. Розвиток інформаційного суспільства в Україні. Київ, НТУУ «КПІ» 2006, 543 с.
6. Маслянюк П.П., Ліссов П.М. Інформаційні ресурси та засоби їх створення // Вісник Східноукр. нац. унів-ту ім. Володимира Даля. – 2007р. – №5 (111) – С. 141-145.
7. Панкратова Н.Д. Загальні тенденції та системні проблеми розвитку інформаційних технологій //Проблеми управління та інформатики, Київ 1999, №1- С.59 – 68.
8. OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2 November 2007 <http://www.omg.org/docs/formal/07-11-02.pdf>.
9. Маслянюк П.П. Концепція інформатизації корпоративних структур // Наукові вісті НТУУ "КПІ" - 2003р. - №3 - с.510-525.
10. Маслянюк П.П. Основні положення методологій системного проектування інформаційно-комунікаційних систем // TCSET 2008.
11. Маслянюк П.П., Майстренко О.С. Моделювання бізнес-процесів організаційної структури // Вісник східноукр. національного університету ім. В. Даля. - 2007р. - №5. - С. 136-141.
12. Маслянюк П.П., Майстренко О.С. Система моделювання бізнес-процесів організаційної структури. // Праці XIII Всеукраїнської наукової

конференції “Сучасні проблеми прикладної математики та інформатики”, Львів 2006. – С.103.

13. Маслянюк П.П., Стокоз К.В. Проблеми проектування та застосування порталів // Вісник східноукраїнського національного університету імені Володимира Даля № 5, 2007 – с. 149-157.

14. Маслянюк П.П. Концепція управління безпекою інформації в корпоративних структурах // Ювілейна науково-технічна конференція „Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні”, Україна, Київ, 9-11 червня 1998.

15. Ракитина Е.А., Пархоменко В.Л. Информатика и информационные системы в экономике: Учеб. пособие. Ч.1. Тамбов: Изд. тамб. гос. техн. ун-та - 2005. - 148с.

16. Закон України “Про телекомунікації” (від 18.112003р. № 1280-IV).

17. [www.answers.com](http://www.answers.com).

18. Черненко М., Слепцов С. Принципы классификации управленческих информационных систем // Корпоративные системы. – 2004г. – №1.

19. Фридланд А.Я. Интеграция технического и гуманитарного подходов к информации и информационным процессам.

20. Закон України “Про Національну програму інформатизації”.

21. Костенко Л.Й., Сорока М.Б. Бібліотека інформаційного суспільства // Бібл. вісн. — 2002. — №3. — С. 33-38.

22. Маслянюк П.П., Ліссов П.М. Інформаційні ресурси та підходи до вимірювання інформації // Вісник КУЕІТУ “Нові технології” – №2(20) – 2008 – с.300-307.

23. Информатика. Учебное пособие для студентов Калининградского государственного технического университета, 2003.

24. Хартли Р.В.Л. Передача информации. // Теория информации и ее приложения. М., 1959.

25. Вяткин В.Б. Синергетическая теория информации: общая характеристика и примеры использования // Наука и оборонный комплекс -

основной ресурс российской модернизации: Материалы межрегиональной научно-практической конференции. Екатеринбург: УрО РАН, 2002.

26. Шеннон К. Работы по теории информации и кибернетике. М., 1963.

27. Бриллюэн Л. Научная неопределенность и информация. М., 1966. С. 34.

28. Колмогоров А.Н. Три подхода к определению понятия "Количество информации". М., 1965.

29. Маслянюк П.П., Лісов П.М. Проблеми і технології продукування інформаційних ресурсів // Матеріали Всеукр. наук.-практ. конф. «Сучасні тенденції розвитку інформаційних технологій в ...» 11-13 грудня 2006р., м. Луганськ, - С.184-189.

30. Маслянюк П.П., Лісов П.М. Інформаційно-комунікаційні системи та технології обробки інформаційних ресурсів // Вісник КУЕІТУ «Нові технології» - 2007р. - №1-2 (15-16) - С. 20.

31. <http://ru.wikipedia.org/>.

32. [http://ru.wikipedia.org/wiki/Список файловых систем](http://ru.wikipedia.org/wiki/Список_файловых_систем).

33. Codd, E. Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks // Communications of the ACM - 1970.

34. Сиротюк О. Особенности проектирования современных баз данных / [www.computerworld.com.ua](http://www.computerworld.com.ua).

35. Inmon, W.H. Tech Topic: What is a Data Warehouse? // Prism Solutions. - 1995. - Vol. 1.

36. Parsaye K.A. A Characterization of Data Mining Technologies and Processes // The Journal of Data Warehousing. - 1998. - vol.1. - P.43-55.

37. Маслянюк П.П., Лісов П.М. Системне проектування інформаційних ресурсів // Матеріали X Міжн. наук.-техн. конф. "Системний аналіз та інформаційні технології" - 20-24 травня 2008р., м.Київ - С.105.

38. Lagoze C., Sompel H. The Open Archives Initiative: Building a low-barrier interoperability framework // JCDL '01 June 17-23, 2001, Roanoke, VA.

39. WorldCat. <http://www.worldcat.org/>.

40. Online Computer Library Center, <http://www.oclc.org>.

41. Shirky C. Ontology is Overrated: Categories, Links, and Tags.
42. Общая информация о поисковых системах. WebKomora. [В Интернете] <http://webkomora.com.ua/ru/articles/web/raskrutka/textbook/910.html>. Общая информация о поисковых системах.
43. Поисковые системы. История и принципы работы поисковых систем. <http://www.prsiterun.com/poi.html#ex2>.
44. Архитектура поисковой системы Rambler. <http://www.rambler.ru/doc/architecture.shtml>.
45. High Solutions. Принципы работы поисковых систем. [http://www.info-highsol.ru/seo\\_poisk.php](http://www.info-highsol.ru/seo_poisk.php).
46. Глосарий Ingate. <http://www.ingate.ru/info/glossary/.rR/relevance/>.
47. Google Inc. Технології пошукової системи Google. <http://www.google.com/technology>.
48. Пошукова система "Апорт". <http://www.aport.ru>.
49. S. Brin, L. Page. The anatomy of a large-scale hypertextual Web search engine. Proceedings of the seventh international conference on World Wide Web 7: 107-117 (Section 2.1.1 Description of PageRank Calculation). 1998 r.
50. Wall A. Local Link Building and Google LocalRank // Search Engine Journal, 12 May, 2006.
51. Gyongyi Z., Garcia-Molina H., Pedersen J. Combating Web Spam with TrustRank.
52. Codata report. <http://www.codata.org>.
53. Антопольский А.Б. Вопросы интеграции информационных ресурсов и структура информационного пространства // Техн. информ. общ. – Интернет и совр. общ.: VI Всеросс. объедин. конф. СПб, 3 - 6 ноября 2003 г – СПб.: Изд-во Филолог. ф-та СПбГУ, 2003. С. 42-43.
54. Tansley R. Building a Distributed, Standards-based Repository Federation, The China Digital Museum Project // D-Lib Magazine. - 2006. - vol.12, num. 7/8.
55. Л.В.Бойчук. Управление знаниями в теории и на практике // Корпоративные системы. - 2006г. - №1. - С. 48-50.

56. Некрасов В. Инструменты для создания Хранилищ данных // "RM-Magazin" - 2002. - №1.
57. Обзор возможностей применения ведущих СУБД для построения хранилищ данных (DataWarehouse). Кузнецов С.А., Артемьев В.В.
58. 2004 CAMagazine survey of Business Intelligence systems.
59. Сообщество Fedora Commons. <http://www.fedora-commons.org/>.
60. <http://www.dspace.org/>.
61. CDSWare site. <http://cdsware.cern.ch>.
62. Інформаційний колектор Greenstone. <http://www.greenstone.org/cgi-bin/library>.
63. Loots M., Camarzan D., Witten I. Greenstone Digital Library. Developer's guide.
64. E-Prints software. <http://software.eprints.org>.
65. Бібліотека Лавалля. Система Archimede. <http://www1.bibl.ulaval.ca/archimede/index.en.html>.
66. Репозитарий BEPress. <http://www.bepress.com/repositories.html>.
67. CONTENTdm. <http://contentdm.com/>.
68. Стандарт Z39.50. <http://en.wikipedia.org/wiki/Z39.50>.
69. Стандарт RSS. <http://ru.wikipedia.org/wiki/RSS>.
70. Маслянюк П.П., Лісов П.М. Дослідження та розробка підсистеми автоматизації проектування інформаційних ресурсів організаційних систем // Матеріали II Всеукр. наук.-практ. конф. "Сучасні тенденції розвитку інформаційних технологій ...", 8-10 квітня 2008 р., м. Луганськ - С.57-59.
71. Маслянюк П.П., Лісов П.М. Проблеми і технології продукування інформаційних ресурсів // VII міжнародна конференція "Интеллектуальный Анализ Информации", ИАИ-2007, Киев, 15-18 мая 2007, Сборник трудов, с.221-229.
72. Маслянюк П.П., Лісов П.М. Проблеми і технології продукування інформаційних ресурсів // Вісник Східноукраїнського Національного ун-ту імені Володимира Даля - №4(110) (Частина 2) – 2007р – с. 136-141.

73. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» (від 31.052005р. М2594-ІУ).
74. Маслянюк П.П. Системне проектування процесів інформатизації // Науові Вісті НТУУ «КПІ» - 2008р. - №1. - С.201-208.
75. Антопольский А.Б. Проблемы управления публичными информационными ресурсами России.
76. Методичні вказівки до виконання організаційно-економічної частини дипломних проектів. Під редакцією А.Т.Чернявського – К.:НТУУ "КПІ", 1999. – 66с.
77. Методичні вказівки до виконання курсових робіт та організаційного розділу дипломних робіт і проектів (для студентів спеціальностей електронного приладобудування). / Укл. М.М. Дугенко, В.С. Сичов, Л.В. Луночна. – К.: КПІ, 1994 – 32с.
78. Маслянюк П.П., Майстренко О.С., Ліссов П.М. Розробка бізнес-моделі НТУУ «КПІ» // VIII міжнародна науково-практична конференція «Системний аналіз та інформаційні технології, 13-16 вересня 2006, Київ, Україна, (КПІ), тези, с.36.
79. Маслянюк П.П., Майстренко О.С. Розробка бізнес-моделі НТУУ «КПІ» // XIII Всеукраїнська наукова конференція “Сучасні проблеми прикладної математики та інформатики” (присвячена 150-річчю з дня народження Івана Франка) 3-5 жовтня 2006, Львів, тези, с.161.
80. Методичні вказівки до використання ФВА при розробці програмних продуктів для студентів факультету інформатики та обчислювальної техніки всіх форм навчання. // Укл. А.Т.Чернявський, К.В. Березовський, Л.В. Швець, В.Ф.Шудра та ін. – К.:КПІ, 1994. – 48с.



## ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

## ДОДАТОК Б. ФРАГМЕНТИ ПРОГРАМНОГО КОДУ

### Фрагмент автоматично генерованого скрипту створення БД

```
/*=====*/
/* Table: "Model" */
/*=====*/
create table "Model" (
  "ModelID"          INTEGER                not null,
  "ProjID"           INTEGER                not null,
  "ModelName"        VARCHAR2(50)          not null,
  "ModelType"        VARCHAR2(50)
  constraint CKC_MODELTYPE_MODEL check ("ModelType" is null or ( "ModelType"
in ('Organization','BusinessProcess','Mixed') )),
  "ModelDescription" VARCHAR2(250)         default 'no description',
  constraint PK_MODEL primary key ("ModelID"),
  constraint FK_MODEL_INCLUDES_PROJECT foreign key ("ProjID")
  references "Project" ("ProjID")
  on delete cascade
);

/*=====*/
/* Table: "OrgUnit" */
/*=====*/
create table "OrgUnit" (
  "OrgUnitID"        INTEGER                not null,
  "ProjID"           INTEGER                not null,
  "Org_OrgUnitID"    INTEGER,
  "OrgUnitName"      VARCHAR2(50)          not null,
  "OrgUnitPplNumber" INTEGER                not null,
  "OrgUnitDescription" VARCHAR2(250)       default 'no description',
  constraint PK_ORGUNIT primary key ("OrgUnitID"),
  constraint FK_ORGUNIT_INCLUDES_PROJECT foreign key ("ProjID")
  references "Project" ("ProjID")
  on delete cascade,
  constraint FK_ORGUNIT_IS_PART_O_ORGUNIT foreign key ("Org_OrgUnitID")
  references "OrgUnit" ("OrgUnitID")
);

/*=====*/
/* Table: "Document" */
/*=====*/
create table "Document" (
  "DocumentID"       INTEGER                not null,
  "ProjID"           INTEGER                not null,
  "DocumentName"     VARCHAR2(50)          not null,
  "DocumentDate"     DATE,
  "DocumentDescription" VARCHAR2(250)       default 'no description',
  constraint PK_DOCUMENT primary key ("DocumentID"),
  constraint FK_DOCUMENT_INCLUDES_PROJECT foreign key ("ProjID")
  references "Project" ("ProjID")
  on delete cascade
);

/*=====*/
/* Table: "Param" */
/*=====*/
create table "Param" (
  "ParamID"          INTEGER                not null,
  "InvProjectID"     INTEGER                not null,
  "ParamType"        VARCHAR2(10)          not null
  constraint CKC_PARAMTYPE_PARAM check ("ParamType" in ('CHPD','ID','PO')),

```

```

"ParamIZ"          FLOAT          not null,
"ParamDescription" VARCHAR2(250)   default 'no description',
constraint PK_PARAM primary key ("ParamID"),
constraint FK_PARAM_IS_DESCRI_INVPROJE foreign key ("InvProjectID")
references "InvProject" ("InvProjectID")
on delete cascade
);

/*=====*/
/* Table: "ParamList" */
/*=====*/
create table "ParamList" (
"ParamListID"      INTEGER          not null,
"ParamID"          INTEGER          not null,
"ParamName"        VARCHAR2(10)     not null,
"ParamPeriod"      INTEGER          not null,
constraint PK_PARAMLIST primary key ("ParamListID"),
constraint FK_PARAMLIS_PARAM_LIS_PARAM foreign key ("ParamID")
references "Param" ("ParamID")
on delete cascade
);

/*=====*/
/* Table: "ParamListValue" */
/*=====*/
create table "ParamListValue" (
"ParamListValueID" INTEGER          not null,
"ParamListID"      INTEGER          not null,
"ParamValue"       FLOAT            not null,
"ParamLName"       VARCHAR2(5)      not null,
constraint PK_PARAMLISTVALUE primary key ("ParamListValueID"),
constraint FK_PARAMLIS_PARAMETERS_PARAMLIS foreign key ("ParamListID")
references "ParamList" ("ParamListID")
on delete cascade
);

/*=====*/
/* Table: "ParamResult" */
/*=====*/
create table "ParamResult" (
"ParamRID"         INTEGER          not null,
"ParamID"          INTEGER          not null,
"ParamDateTime"   TIMESTAMP         not null,
"ParamX"           FLOAT            not null,
"ParamY"           FLOAT            not null,
constraint PK_PARAMRESULT primary key ("ParamRID"),
constraint FK_PARAMRES_RESULTS_PARAM foreign key ("ParamID")
references "Param" ("ParamID")
on delete cascade
);

/*=====*/
/* Table: "Process" */
/*=====*/
create table "Process" (
"ProcessID"        INTEGER          not null,
"ProjID"           INTEGER          not null,
"ProcessName"      VARCHAR2(50)     not null,
"ProcessDescription" VARCHAR2(250)   default 'no description',
constraint PK_PROCESS primary key ("ProcessID"),
constraint FK_PROCESS_INCLUDES_PROJECT foreign key ("ProjID")
references "Project" ("ProjID")
on delete cascade
);

```

```

/*=====*/
/* Table: "Human" */
/*=====*/
create table "Human" (
  "HumanID"          INTEGER                not null,
  "ProjID"           INTEGER                not null,
  "OrgUnitID"        INTEGER,
  "HumanName"        VARCHAR2(25)          not null,
  "HumanJob"         VARCHAR2(50)          not null,
  "HumanBirthday"    DATE                  not null,
  "HumanJobDate"     DATE                  not null,
  "HumanPassData"    VARCHAR2(50),
  constraint PK_HUMAN primary key ("HumanID"),
  constraint FK_HUMAN_INCLUDES__PROJECT foreign key ("ProjID")
    references "Project" ("ProjID")
    on delete cascade,
  constraint FK_HUMAN_WORKS_IN_ORGUNIT foreign key ("OrgUnitID")
    references "OrgUnit" ("OrgUnitID")
);

/*=====*/
/* Table: "ITSolution" */
/*=====*/
create table "ITSolution" (
  "ITSolutionID"     INTEGER                not null,
  "ITStandartID"     INTEGER                not null,
  "ITSolutionName"   VARCHAR2(25)          not null,
  "ITSolutionPrice"  NUMBER(8,2)          not null,
  "ITSolutionDesc"   VARCHAR2(250)
);

```

### Фрагмент автоматично генерованого скрипта створення тригерів

```

CREATE OR REPLACE PROCEDURE "report" (org number, proj in number ) as
  tt number;
  fileID1 sys.UTL_FILE.FILE_TYPE;

  CURSOR stat_cur
  IS
    SELECT "Model"."ModelType", "Model"."ModelName"
    FROM "Model", "Project", "Organization" where
      "Model"."ProjID" = "Project"."ProjID" AND
      "Project"."OrgID" = "Organization"."OrgID" AND
      "Project"."ProjID" = proj AND
      "Organization"."OrgID" = org;

begin
  fileID1 := sys.UTL_FILE.FOPEN ('MYDIR', 'report.txt', 'W');

  FOR stat_rec IN stat_cur
  LOOP

    SYS.UTL_FILE.PUTF(fileID1, '%s %s\n',
      lpad(TO_CHAR (stat_rec."ModelName"), 4),
      lpad(TO_CHAR (stat_rec."ModelType"),14));

  END LOOP;
  sys.utl_file.fclose(fileID1);
end;
/

```

```

CREATE OR REPLACE PROCEDURE "admin_create" as
  ex NUMBER;
begin
  SELECT COUNT(*) INTO ex FROM "Worker";
  if ex=0 then
    INSERT INTO "Worker"
VALUES(0,'Alex','Maystrenko','admin','master','master',99.99,'30-May-1985','01-
May-2002','');
  end if;
  COMMIT;
end;
/

CREATE OR REPLACE TRIGGER "TTT"
AFTER UPDATE
ON DBUSER."Model"
REFERENCING NEW AS New OLD AS Old
FOR EACH ROW
DECLARE
tmpVar NUMBER;
ex NUMBER;
BEGIN
  tmpVar := 0;
/*
  SELECT MySeq.NEXTVAL INTO tmpVar FROM dual;
  :NEW.SequenceColumn := tmpVar;
  :NEW.CreatedDate := SYSDATE;
  :NEW.CreatedUser := USER;
*/
  SELECT COUNT(*) INTO ex FROM "Model", "uses_proc" WHERE
"Model"."ModelID"="uses_proc"."ModelID" AND
"Model"."ModelType"!='BusinessProcess';
  if ex!=0 then
    UPDATE "Model" SET "ModelType"='BusinessProcess' WHERE "ModelID" IN
(SELECT "Model"."ModelID" FROM "Model", "uses_proc" WHERE
"Model"."ModelID"="uses_proc"."ModelID" AND
"Model"."ModelType"!='BusinessProcess');
  end if;

  SELECT COUNT(*) INTO ex FROM "Model", "uses_orgunit" WHERE
"Model"."ModelID"="uses_orgunit"."ModelID" AND
"Model"."ModelType"!='Organization';
  if ex!=0 then
    UPDATE "Model" SET "ModelType"='Organization' WHERE "ModelID" IN (SELECT
"Model"."ModelID" FROM "Model", "uses_orgunit" WHERE
"Model"."ModelID"="uses_orgunit"."ModelID" AND
"Model"."ModelType"!='Organization');
  end if;

  SELECT COUNT(*) INTO ex FROM "Model", "uses_bpgoal" WHERE
"Model"."ModelID"="uses_bpgoal"."ModelID" AND "Model"."ModelType"!='Mixed';
  if ex!=0 then
    UPDATE "Model" SET "ModelType"='Mixed' WHERE "ModelID" IN (SELECT
"Model"."ModelID" FROM "Model", "uses_bpgoal" WHERE
"Model"."ModelID"="uses_bpgoal"."ModelID" AND "Model"."ModelType"!='Mixed');
  end if;

  SELECT COUNT(*) INTO ex FROM "Model", "uses_human" WHERE
"Model"."ModelID"="uses_human"."ModelID" AND "Model"."ModelType"!='Mixed';
  if ex!=0 then
    UPDATE "Model" SET "ModelType"='Mixed' WHERE "ModelID" IN (SELECT
"Model"."ModelID" FROM "Model", "uses_human" WHERE
"Model"."ModelID"="uses_human"."ModelID" AND "Model"."ModelType"!='Mixed');
  end if;

```

```

        SELECT COUNT(*) INTO ex FROM "Model", "uses_thing" WHERE
"Model"."ModelID"="uses_thing"."ModelID" AND "Model"."ModelType"!='Mixed';
        if ex!=0 then
            UPDATE "Model" SET "ModelType"='Mixed' WHERE "ModelID" IN (SELECT
"Model"."ModelID" FROM "Model", "uses_thing" WHERE
"Model"."ModelID"="uses_thing"."ModelID" AND "Model"."ModelType"!='Mixed');
        end if;

        SELECT COUNT(*) INTO ex FROM "Model", "uses_doc" WHERE
"Model"."ModelID"="uses_doc"."ModelID" AND "Model"."ModelType"!='Mixed';
        if ex!=0 then
            UPDATE "Model" SET "ModelType"='Mixed' WHERE "ModelID" IN (SELECT
"Model"."ModelID" FROM "Model", "uses_doc" WHERE
"Model"."ModelID"="uses_doc"."ModelID" AND "Model"."ModelType"!='Mixed');
        end if;

    EXCEPTION
        WHEN OTHERS THEN
            -- Consider logging the error and then re-raise
            RAISE;
END ;

/
ALTER TRIGGER "TTT" ENABLE
/
CREATE OR REPLACE TRIGGER "ThingAdd" after insert
on "uses_thing"
REFERENCING NEW AS newRow
for each row
declare
    mytype VARCHAR(50);
begin
    SELECT "Model"."ModelType" INTO mytype FROM "Model" WHERE
"Model"."ModelID"=:newRow."ModelID";
    if mytype != 'Mixed' then
        DELETE FROM "uses_thing" WHERE "ModelID"=:newRow."ModelID";
    end if;
end;
/
ALTER TRIGGER "ThingAdd" ENABLE
/
CREATE OR REPLACE TRIGGER "ProcessAdd" after insert
on "uses_proc"
REFERENCING NEW AS newRow
for each row
declare
    mytype VARCHAR(50);
begin
    SELECT "Model"."ModelType" INTO mytype FROM "Model" WHERE
"Model"."ModelID"=:newRow."ModelID";
    if mytype != 'BusinessProcess' then
        DELETE FROM "uses_proc" WHERE "ModelID"=:newRow."ModelID";
    end if;
end;
/
ALTER TRIGGER "ProcessAdd" ENABLE
/
CREATE OR REPLACE TRIGGER "OrgUnitAdd" after insert
on "uses_orgunit"
REFERENCING NEW AS newRow
for each row
declare
    mytype VARCHAR(50);
begin

```

```

        SELECT "Model"."ModelType" INTO mytype FROM "Model" WHERE
"Model"."ModelID"=:newRow."ModelID";
        if mytype != 'Organization' then
            DELETE FROM "uses_orgunit" WHERE "ModelID"=:newRow."ModelID";
        end if;
end;
/
ALTER TRIGGER "OrgUnitAdd" ENABLE
/
CREATE OR REPLACE TRIGGER "HumanAdd" after insert
on "uses_human"
REFERENCING NEW AS newRow
for each row
declare
    mytype VARCHAR(50);
begin
    SELECT "Model"."ModelType" INTO mytype FROM "Model" WHERE
"Model"."ModelID"=:newRow."ModelID";
    if mytype != 'Mixed' then
        DELETE FROM "uses_human" WHERE "ModelID"=:newRow."ModelID";
    end if;
end;
/
ALTER TRIGGER "HumanAdd" ENABLE
/
CREATE OR REPLACE TRIGGER "DocAdd" after insert
on "uses_doc"
REFERENCING NEW AS newRow
for each row
declare
    mytype VARCHAR(50);
begin
    SELECT "Model"."ModelType" INTO mytype FROM "Model" WHERE
"Model"."ModelID"=:newRow."ModelID";
    if mytype != 'Mixed' then
        DELETE FROM "uses_doc" WHERE "ModelID"=:newRow."ModelID";
    end if;
end;
/
ALTER TRIGGER "DocAdd" ENABLE
/
CREATE OR REPLACE TRIGGER "ChangeType_Thing" after update
of "ModelID",
    "ProjID"
on "Model" for each row
declare
    ex                integer;
    integrity_error   exception;
    errno             integer;
    errmsg            char(200);
    dummy             integer;
    found             boolean;
begin
    IntegrityPackage.NextNestLevel;
    IntegrityPackage.PreviousNestLevel;

    SELECT COUNT(*) INTO ex FROM "Model", "uses_thing" WHERE
"Model"."ModelID"="uses_thing"."ModelID" AND "Model"."ModelType"!='Mixed';
    if ex!=0 then
        UPDATE "Model" SET "ModelType"='Mixed' WHERE "ModelID" IN (SELECT
"Model"."ModelID" FROM "Model", "uses_thing" WHERE
"Model"."ModelID"="uses_thing"."ModelID" AND "Model"."ModelType"!='Mixed');
    end if;

-- Errors handling

```

```

exception
  when integrity_error then
    begin
      IntegrityPackage.InitNestLevel;
      raise_application_error(errno, errmsg);
    end;
end;
/
ALTER TRIGGER "ChangeType_Thing" ENABLE
/
CREATE OR REPLACE TRIGGER "ChangeType_Proc" after update
of "ModelID",
  "ProjID"
on "Model" for each row
declare
  ex          integer;
  integrity_error exception;
  errno      integer;
  errmsg     char(200);
  dummy      integer;
  found      boolean;
begin
  IntegrityPackage.NextNestLevel;
  IntegrityPackage.PreviousNestLevel;

  SELECT COUNT(*) INTO ex FROM "Model", "uses_proc" WHERE
"Model"."ModelID"="uses_proc"."ModelID" AND
"Model"."ModelType"!='BusinessProcess';
  if ex!=0 then
    UPDATE "Model" SET "ModelType"='BusinessProcess' WHERE "ModelID" IN
(SELECT "Model"."ModelID" FROM "Model", "uses_proc" WHERE
"Model"."ModelID"="uses_proc"."ModelID" AND
"Model"."ModelType"!='BusinessProcess');
  end if;

-- Errors handling
exception
  when integrity_error then
    begin
      IntegrityPackage.InitNestLevel;
      raise_application_error(errno, errmsg);
    end;
end;
/
ALTER TRIGGER "ChangeType_Proc" ENABLE
/

```